

Informacja o programie studiów

Jednostka realizująca studia:	Kolegium Studiów Administracji i Informatyki	
Nazwa kierunku:	Informatyka	
Poziom:	Pierwszego stopnia inżynierski	
Profil kształcenia:	Praktyczny	
Forma studiów:	Niestacjonarna	
Dziedzina nauki:	Dziedzina nauk inżynieryjno-technicznych	
Dyscyplina wiodąca:	Informatyka techniczna i telekomunikacja	
Liczba semestrów:	7	
Liczba punktów ECTS:	210	
Uzyskiwany tytuł zawodowy:	inżynier	
Wymiar odbywania praktyk zawodowych:	6 miesięcy (960 godzin)	
Łączna liczba godzin zajęć w programie studiów:	1320 / 2280 z praktyką	
	Wartość	Udział procentowy
Liczba punktów ECTS przypisana do dyscypliny wiodącej:	210	100%
Liczba godzin zajęć prowadzonych przez pracowników zatrudnionych w Uczelni jako podstawowym miejscu pracy:	784	59% bez praktyki
Liczba punktów ECTS, jaką student musi uzyskać w ramach praktyk zawodowych:	36	17%
Łączna liczba punktów ECTS, jaką student musi uzyskać na zajęciach w ramach zajęć prowadzonych z bezpośrednim udziałem nauczycieli akademickich lub innych osób prowadzących zajęcia:	89,4	43%
Liczba punktów ECTS, jaką student musi uzyskać w ramach zajęć z dziedziny nauk humanistycznych lub nauk społecznych: - Praca w zespole - Praca w środowisku międzynarodowym - Problemy społeczne i zawodowe informatyki	6	2,86%
Liczba punktów ECTS, jaką student otrzyma realizując przedmioty do wyboru:	85	40%
Liczba punktów ECTS, jaką student otrzyma realizując zajęcia kształtujące umiejętności praktyczne:	152,5	73%

Opis programu studiów

Koncepcja i cele kształcenia

Koncepcja kształcenia na kierunku Informatyka opiera się na ścisłym powiązaniu teorii z praktyką, wiedzy z umiejętnością i kompetencjami społecznymi. Celem kierunku jest zapewnienie wysokiej jakości kształcenia w oparciu o wartości akademickie, a jednocześnie otwartej na nowoczesne metody dydaktyczne i techniki nauczania oraz wdrażanie studentów do ustawicznego doskonalenia zawodowego – przede wszystkim przez samokształcenie. Celem kształcenia jest przygotowanie kompetentnego i świadomego pracownika posiadającego wiedzę ogólną i szczegółową z zakresu informatyki oraz nauk społecznych. Absolwent dzięki posiadaniu wiedzy i umiejętności praktycznych jest właściwie przygotowany do podejmowania wyzwań dynamicznego i konkurencyjnego rynku pracy. Jest to możliwe dzięki zdobyciu wiedzy zawodowej; praktycznego opanowania metod, środków i form działalności zawodowej; rozwinięcia uzdolnień niezbędnych do wykonywania zawodu; rozwinięcia zainteresowań związanych z zawodem; kształtowania społecznie oczekiwanych i akceptowanych postaw.

Związek z misją i strategią Uczelni

Program studiów jest skonstruowany w ten sposób, aby zgodnie z Misją Uczelni umożliwić wszechstronny rozwój studentów. Podstawą jest proponowanie kandydatom zróżnicowanej oferty dydaktycznej umożliwiającej zaspokojenie z jednej strony potrzeb lokalnego rynku pracy w sektorze IT, a z drugiej strony zaspokojenie społecznej potrzeby edukacji na wysokim poziomie, umożliwiającym twórcze funkcjonowanie absolwenta w skali globalnej.

Realizacja celów strategicznych Uczelni

- w obszarze jakości kształcenia polega na dostosowywaniu programu studiów do oczekiwań studentów oraz interesariuszy z otoczenia społeczno-gospodarczego, efektywne realizowanie procesu dydaktycznego poprzez wykorzystywanie nowoczesnych metod dydaktycznych i realizacji ponad 65% zajęć w formie praktycznej, wspieranie studentów w procesie uczenia się poprzez wprowadzenie przedmiotów ułatwiających studiowanie oraz dostarczanie materiałów dydaktycznych przez platformę e-learningową, reagowanie na informacje pozyskiwane z systemu zapewniania jakości kształcenia oraz usprawnianie jego funkcjonowania;
- w obszarze pracowników polega na wspieraniu umiejętności dydaktycznych oraz rozwoju nauczycieli akademickich poprzez szkolenia i udział w seminariach oraz konferencjach, dostosowanie kompetencji pracowników administracyjnych poprzez szkolenia i funkcjonowanie systemu informatycznego, działań zwiększających zaangażowanie pracowników oraz utożsamianie się z Uczelnią poprzez dostęp do pracowni i zasobów bibliotecznych, możliwość publikacji w wydawnictwie czy organizowanie spotkań towarzyskich;
- w obszarze otoczenia polega na wzmacnianiu pozytywnego wizerunku Uczelni poprzez realizowanie projektów inżynierskich pod indywidualne zamówienia interesariuszy, zwiększenie zaangażowania otoczenia w proces kształcenia poprzez zapraszanie fachowców do prowadzenia zajęć czy opiekę nad studentami realizującymi praktyki studenckie, zwiększanie atrakcyjności Uczelni w środowisku lokalnym poprzez udzielanie się w akcjach społecznych czy branie udziału w lokalnych konkursach;
- w obszarze Uczelni polega na udzielaniu pomocy w procesach informatyzacji procesów zarządzania. prowadzenie zrównoważonej polityki finansowej, dbanie o infrastrukturę oraz planowa rozbudowa zasobów o nowe laboratoria oraz ich wyposażenie.

Wszystkie te działania mają na celu podniesienie rozpoznawalności Kierunku Informatyka w skali regionu.

Warunki przyjęć na studia

Warunkiem przyjęć na studia jest zarejestrowanie się w systemie rekrutacyjnym, złożenie dokumentów (kopia świadectwa dojrzałości, ankieta osobowa ze zdjęciem, zaświadczenie stwierdzające brak przeciwwskazań do podjęcia studiów wydane przez lekarza z uprawnieniami do przeprowadzania badań profilaktycznych) oraz pozytywny wynik postępowania kwalifikacyjnego. W postępowaniu kwalifikacyjnym brane są pod uwagę rezultaty egzaminu maturalnego, przy czym wśród przedmiotów dodatkowych preferowane są wyniki z informatyki, matematyki lub fizyki. Warunkiem wpisania na listę studentów jest uzyskanie wyniku kwalifikacyjnego równego lub wyższego od granicznego. Warunki rekrutacji, w tym sposób wyliczania wyniku kwalifikacyjnego oraz jego wartości granicznej, ustalane są corocznie przez Senat Uczelni. Kandydaci wpisani na listę stają się studentami dopiero po złożeniu ślubowania oraz podpisaniu z Uczelnią umowy o warunkach odpłatności za studia.

Realizacja praktyki

Studenckie praktyki zawodowe są integralną częścią studiów i mają za zadanie stworzenie warunków do praktycznego wykorzystania zdobytej wiedzy i nabycie doświadczenia przez studenta w wybranych obszarach zastosowań informatyki. Studenckie praktyki zawodowe trwają sześć miesięcy (120 dni, max. 960 godz.) i mogą być realizowane jako ciągłe lub nieciągłe. Zaliczenie poszczególnych etapów praktyk odbywa się w każdym semestrze od drugiego roku. Łączna liczba punktów ECTS przypisanych studenckim praktykom zawodowym wynosi 36. Praktyki organizowane są w podmiotach instytucjonalnych i biznesowych w obszarze IT w zakresie:

- organizacji i zakresu działania służb informatycznych firmy,
- planowania i rozwoju systemów informatycznych,
- zadań utrzymania systemów informatycznych,
- planowania i wdrażania polityki bezpieczeństwa,
- wykorzystania różnych technologii informatycznych w przedsiębiorstwach,
- doboru systemów i narzędzi informatycznych i zasad ich pozyskania,
- dokumentowania prac rozwojowych i eksploatacyjnych,
- serwisu sprzętu informatycznego,
- organizacji wsparcia użytkowników końcowych.

Proces dyplomowania

Warunkiem koniecznym ukończenia studiów jest zaliczenie wszystkich zajęć z programu studiów i uzyskanie 210 punktów ECTS oraz zdanie egzaminu dyplomowego. W ciągu ostatnich trzech semestrów studenci wykonują indywidualny lub zespołowy projekt inżynierski pod kierunkiem nauczyciela akademickiego. Musi to być praktyczne rozwiązanie zadania inżynierskiego z zakresu informatyki. Na egzaminie dyplomowym studenci prezentują przygotowany projekt oraz odpowiadają na co najmniej dwa pytania z zestawu zagadnień przygotowanego na podstawie programu kształcenia. Pytania egzaminacyjne są udostępnione studentom.

Możliwość dalszego kształcenia się absolwenta

Absolwenci kierunku Informatyka SWPW w Płocku mogą kształcić się na studiach drugiego stopnia na kierunku informatyka lub innych kierunkach umożliwiających uzyskanie tytułu magistra oraz na studiach podyplomowych.

Kompetencje absolwenta

Absolwenci inżynierskich studiów na kierunku Informatyka posiadają wiedzę z zakresu budowy i funkcjonowania współczesnych systemów komputerowych, systemów operacyjnych, sieci komputerowych, baz danych, algorytmów, języków programowania. Posiadają umiejętności programowania komputerów, projektowania i implementacji aplikacji internetowych, systemów automatyki, programowania komputerów jednoukładowych i robotów. Absolwenci uzyskują wiedzę z aktualnej tematyki bezpieczeństwa systemów informatycznych i kryptografii oraz sztucznej inteligencji. Dzięki wiedzy i umiejętności z zakresu inżynierii oprogramowania, cyfrowej dokumentacji technicznej, komunikacji człowiek-komputer, budowy interfejsów użytkownika, pracy zespołowej, wykonywania projektów informatycznych oraz pracy w środowisku międzynarodowym są praktycznie przygotowani do aktywnego udziału w realizacji projektów informatycznych. W zależności od wybranej ścieżki kształcenia – studenci mogą ukończyć studia w jednej z wybranych specjalności: Aplikacje internetowe i mobilne lub Sieci komputerowe i cyberbezpieczeństwo.

Posiadane kwalifikacje oraz uprawnienia zawodowe (o ile możliwe)

Możliwość zatrudnienia absolwenta

Absolwent może pracować w organach administracji publicznej lub samorządowej, instytucjach publicznych, instytucjach finansowych, podmiotach gospodarczych wykorzystujących w swojej działalności systemy informatyczne; firmach programistycznych projektujących, wytwarzających i wdrażających systemy informatyczne; podmiotach zapewniających utrzymanie infrastruktury sieciowej; firmach zapewniających bezpieczeństwo systemów teleinformatycznych.

Specjalność: Aplikacje internetowe i mobilne

Absolwent specjalności **Aplikacje internetowe i mobilne** dodatkowo nabywa wiedzę i umiejętności z zakresu internetowych technologii informatycznych, programowania urządzeń mobilnych, programowania interfejsów graficznych, programowania systemów back-end oraz front-end.

Dodatkowe możliwości zatrudnienia absolwenta

Absolwent jest przygotowany do pracy jako administrator firmowych portali informacyjnych, projektant informatycznych systemów internetowych i mobilnych, programista aplikacji internetowych i mobilnych.

Specjalność: Sieci komputerowe i cyberbezpieczeństwo

Absolwent specjalności **Sieci komputerowe i cyberbezpieczeństwo** dodatkowo nabywa wiedzę i umiejętności z zakresu bezpieczeństwa oprogramowania i testów penetracyjnych, projektowania i eksploatacji bezpiecznych sieci teletransmisyjnych, administracji sieciami komputerowymi, wirtualizacji i rozwiązań chmurowych, ochrony i zarządzania bezpieczeństwem cyberprzestrzeni przedsiębiorstw.

Dodatkowe możliwości zatrudnienia absolwenta

Absolwent jest przygotowany do pracy jako administrator sieci komputerowych, specjalista w zakresie zapewnienia bezpieczeństwa systemów teleinformatycznych, doradca lub konsultant ds. bezpieczeństwa oraz ochrony informacji.

Wykaz Zajęć do wyboru realizowanych w ramach Zajęć kierunkowych

Lp.	Zajęcia	Liczba godzin i forma zajęć	Rodzaj zaliczenia	ECTS
1	Podstawy telekomunikacji	8W+16C	Z	3
2	Wirtualizacja systemów komputerowych	8W+16C	Z	3
3	Język Python	8W+16C	Z	3
5	Zarządzanie procesami i systemami IT	8W+16C	Z	3
4	Podstawy programowania robotów	8W+16C	Z	3
6	Grywalizacja, czyli zastosowanie mechanizmów gier w informatyce	8W+16C	Z	3
7	Komputerowa grafika przestrzenna	8W+16C	Z	3
8	Internet rzeczy	8W+16C	Z	3
9	Administracja serwerami internetowymi	8W+16C	Z	3

Wykaz Zajęć do wyboru realizowanych w ramach Zajęć wynikających ze specjalności**Specjalność: Sieci komputerowe i cyberbezpieczeństwo**

L.p.	Zajęcia	Liczba godzin i forma zajęć	Rodzaj zaliczenia	ECTS
1	Zarządzanie bezpieczeństwem informacji	16W + 16C + 8Pr	Z	6
2	Monitoring zagrożeń cyberbezpieczeństwa	16W + 16C + 8Pr	Z	6
3	Informatyka śledcza	16W + 16C + 8Pr	Z	6
4	Systemy zarządzania bezpieczeństwem sieci komputerowych	8W + 24C	Z	4
5	Bezpieczeństwo oprogramowania i testy penetracyjne	8W + 24C	Z	4

Specjalność: Aplikacje internetowe i mobilne

L.p.	Zajęcia	Liczba godzin i forma zajęć	Rodzaj zaliczenia	ECTS
1	Front-end developer	16W + 16C + 8Pr	Z	6
2	Internetowe technologie informatyczne	16W + 16C + 8Pr	Z	6
3	Programowanie aplikacji mobilnych	16W + 16C + 8Pr	Z	6
4	Projektowanie interfejsów graficznych	8W+24C	Z	4
5	Bezpieczeństwo oprogramowania i testy penetracyjne	8W + 24C	Z	4

Język angielski I

Nazwa zajęć po angielsku: English in practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: II

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 2

Materiały: kurs e-learningowy

Język zajęć: angielski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia ogólnouczelniane

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Ćwiczenia.....30 / 30 godz..... 10 / 14 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Małgorzata Piątek

Opis zajęć:

Zajęcia przeznaczone są dla studentów pragnących doskonalić swoje umiejętności z języka angielskiego na poziomie A2+. Językiem prowadzenia zajęć jest język angielski z zastosowaniem języka polskiego w wyjątkowych sytuacjach podczas zajęć. Celem zajęć jest doskonalenie zastosowania języka angielskiego w środowisku profesjonalnym każdego studenta jak i w sytuacjach codziennych (zakupy, podróże, restauracje, hobby...). Zajęcia wzbogacone są słownictwem z zakresu informatyki - budowy wewnętrznej i urządzeń zewnętrznych.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
opisuje w języku angielskim wybrane praktyczne zastosowania informatyki, w szczególności dotyczące sprzętu komputerowego	K_W03
Umiejętności (student potrafi)	
porozumiewa się w prostych lub złożonych sytuacjach zawodowych informatyka, wykorzystując terminologię specjalistyczną oraz techniki komunikacji ustnej i pisemnej	K_U14
uczestniczy w dyskusji, przedstawia stanowisko i reaguje na opinie innych osób w języku angielskim	K_U15
formułuje wypowiedzi ustne i pisemne w języku angielskim na poziomie właściwym dla danego semestru, z wykorzystaniem słownictwa ogólnego i informatycznego	K_U16
współpracuje w parach i zespołach podczas realizacji zadań językowych oraz sytuacyjnych związanych z pracą w środowisku IT	K_U18
Kompetencje społeczne (student jest gotów do)	
jest gotów oceniać poprawność własnych wypowiedzi i krytycznie analizować otrzymywaną informację zwrotną	K_K01
jest gotów rozwijać kompetencje językowe z wykorzystaniem kursu e-learningowego, źródeł elektronicznych i materiałów anglojęzycznych	U_K07

Metody i formy kształcenia:

- Metoda ćwiczeń
- Metoda sytuacyjna
- Praca ze źródłem elektronicznym

Metody oceniania:

- Zaliczenie ustne
- Zaliczenie testowe
- Częstkowe sprawdziany pisemne - kolokwia

Ćwiczenia (Ć)

- Zastosowanie czasu Present Simple do opisanie czynności stałych i czynności powtarzalnych, nawyków.
- Zastosowanie czasu Present Continuous dla czynności tymczasowych.

3. Zastosowanie czasów przeszłych (Past Simple i Past Continuous) w narracji.
4. Wygląd zewnętrzny; opisywanie cech charakteru, ubioru; opis wakacji.
5. Budowa komputera: computer hardware and peripherals, basic computer components, describing a workstation, simple instructions and user needs, describing everyday use of digital devices.

Literatura podstawowa

1. ENGLISH FILE 4th edition (C. Latham-Koening, C. Oxenden, J. Lambert, P. Seligson) Oxford University Press, 2019
2. B. Błaszczak, English 4 IT. Praktyczny kurs języka angielskiego dla specjalistów IT i nie tylko. Helion 2017
3. GCFGlobal — Computer Basics (https://iab.live/modules/en-gcf_learn_2021/edu.gcfglobal.org/en/computerbasics/index.html) - wybrane lekcje: Basic Parts of a Computer, Buttons and Ports on a Computer, Inside a Computer, Mobile Devices, Setting Up a Computer, Getting Started with Your First Computer.
4. British Council LearnEnglish Teens — Technology (<https://learnenglishteens.britishcouncil.org/vocabulary/a1-a2-vocabulary/technology>)

Literatura uzupełniająca

1. ENGLISH FILE 4th edition (C. Latham-Koening, C. Oxenden, J. Lambert, P. Seligson) Oxford University Press, 2019 Workbook

Źródła dodatkowe

1. Platforma edukacyjna wlodek.edu.pl
2. BBC portal (www.bbc.co.uk)
3. TechTerms (<https://techterms.com/category/hardware>)
4. DigitalLearn (<https://training.digitallearn.org/courses?category=pc-hardware>)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	20 godz.....	16 godz.
• Samodzielne wykonanie zadań.....	10 godz.....	10 godz.
• Przygotowanie się do zajęć.....	10 godz.....	10 godz.
• Realizacja kursu e-learningowego.....	10 godz.....	14 godz.
Suma obciążeń:	50 godz.....	50 godz.

Zaliczenie zajęć:

Oceniając każdą pracę studenta (zarówno ustną jak i pisemną) zastosowanie mają poniższe progi procentowe:

- 91%–100%: ocena bardzo dobra (5.0)
- 81%–90%: ocena dobra plus (4.5)
- 71%–80%: ocena dobra (4.0)
- 61%–70%: ocena dostateczna plus (3.5)
- 51%–60%: ocena dostateczna (3.0)
- 50% i poniżej: niedostateczna (2.0) – zajęcia niezaliczone.

Warunkiem koniecznym do zaliczenia semestru jest uzyskanie pozytywnych wyników z wszystkich elementów.

Język angielski II

Nazwa zajęć po angielsku: English in practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: III

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 2

Materiały: kurs e-learningowy

Język zajęć: angielski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia ogólnouczelniane

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Ćwiczenia.....30 / 30 godz..... 10 / 14 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Małgorzata Piątek

Opis zajęć:

Zajęcia z języka angielskiego przeznaczone są dla studentów pragnących doskonalić swoje umiejętności z języka angielskiego na poziomie B1. Językiem prowadzenia zajęć jest język angielski z zastosowaniem języka polskiego w wyjątkowych sytuacjach podczas zajęć. Celem zajęć jest doskonalenie zastosowania języka angielskiego w środowisku profesjonalnym każdego studenta jak i w sytuacjach codziennych (zakupy, podróże, restauracje, hobby...). Zajęcia wzbogacone są słownictwem z zakresu informatyki - systemów zarządzania bazami danych.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
opisuje w języku angielskim wybrane praktyczne zastosowania informatyki, w szczególności dotyczące baz danych	K_W03
Umiejętności (student potrafi)	
porozumiewa się w prostych lub złożonych sytuacjach zawodowych informatyka, wykorzystując terminologię specjalistyczną oraz techniki komunikacji ustnej i pisemnej	K_U14
uczestniczy w dyskusji, przedstawia stanowisko i reaguje na opinie innych osób w języku angielskim	K_U15
formułuje wypowiedzi ustne i pisemne w języku angielskim na poziomie właściwym dla danego semestru, z wykorzystaniem słownictwa ogólnego i informatycznego	K_U16
współpracuje w parach i zespołach podczas realizacji zadań językowych oraz sytuacyjnych związanych z pracą w środowisku IT	K_U18
Kompetencje społeczne (student jest gotów do)	
jest gotów oceniać poprawność własnych wypowiedzi i krytycznie analizować otrzymywaną informację zwrotną	K_K01
jest gotów rozwijać kompetencje językowe z wykorzystaniem kursu e-learningowego, źródeł elektronicznych i materiałów anglojęzycznych	U_K07

Metody i formy kształcenia:

- Metoda ćwiczeń
- Metoda sytuacyjna
- Praca ze źródłem elektronicznym

Metody oceniania:

- Zaliczenie ustne
- Zaliczenie testowe
- Częstkowe sprawdziany pisemne - kolokwia

Ćwiczenia (Ć)

- Słownictwo związane z podróżą samolotem, zakupami; opisywanie miast i miejsc zamieszkania
- Wyrażanie przyszłości (to be going to, Present Continuous)

3. Zastosowanie czasu Present Perfect Simple oraz porównanie z Past Simple
4. Zastosowanie konstrukcji porównawczych ze stopniem wyższym i najwyższym przymiotnika
5. Bazy danych i DBMS: database, table, record, field, query, describing database functions, explaining a simple data model, reporting a data problem.

Literatura podstawowa

1. ENGLISH FILE 4th edition (C. Latham-Koenig, C. Oxenden, J. Lambert, P. Seligson) Oxford University Press, 2019
2. B. Błaszczak, English 4 IT. Praktyczny kurs języka angielskiego dla specjalistów IT i nie tylko. Helion 2017
3. Microsoft Support, Database basics (<https://support.microsoft.com/en-US/Access/database-basics>)
4. Khan Academy, Intro to SQL: Querying and managing data (<https://www.khanacademy.org/computing/computer-programming/sql/sql-basics>)

Literatura uzupełniająca

1. ENGLISH FILE 4th edition (C. Latham-Koenig, C. Oxenden, J. Lambert, P. Seligson) Oxford University Press, 2019 Workbook

Źródła dodatkowe

1. Platforma edukacyjna wlodek.edu.pl
2. www.bbc.co.uk
3. W3Schools, SQL Tutorial (<https://www.w3schools.com/Sql/>)
4. Runestone Academy, A Practical Introduction to Databases (https://runestone.academy/ns/books/published/practical_db/PART2_DATA_MODELING/02-ERD/ERD.html)
5. IBM, What is a database? (<https://www.ibm.com/think/topics/database>)
6. AWS, What is a Database Management System? (<https://aws.amazon.com/what-is/dbms/>)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	20 godz.....	16 godz.
• Przygotowanie się do zajęć.....	10 godz.....	10 godz.
• Samodzielne wykonanie zadań.....	10 godz.....	10 godz.
• Realizacja kursu e-learningowego.....	10 godz.....	14 godz.
Suma obciążeń:.....	50 godz.....	50 godz.

Zaliczenie zajęć:

Oceniając każdą pracę studenta (zarówno ustną jak i pisemną) zastosowanie mają poniższe progi procentowe:

- 91%–100%: ocena bardzo dobra (5.0)
- 81%–90%: ocena dobra plus (4.5)
- 71%–80%: ocena dobra (4.0)
- 61%–70%: ocena dostateczna plus (3.5)
- 51%–60%: ocena dostateczna (3.0)
- 50% i poniżej: niedostateczna (2.0) – zajęcia niezaliczone.

Warunkiem koniecznym do zaliczenia semestru jest uzyskanie pozytywnych wyników z wszystkich elementów.

Język angielski III

Nazwa zajęć po angielsku: English in practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: IV

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 2

Materiały: własny skrypt lub podręcznik

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia ogólnouczelniane

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Ćwiczenia.....30 / 30 godz..... 10 / 14 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Małgorzata Piątek

Opis zajęć:

Zajęcia z języka angielskiego przeznaczone są dla studentów pragnących doskonalić swoje umiejętności z języka angielskiego na poziomie B1+. Językiem prowadzenia zajęć jest język angielski z zastosowaniem języka polskiego w wyjątkowych sytuacjach podczas zajęć. Celem zajęć jest doskonalenie zastosowania języka angielskiego w środowisku profesjonalnym każdego studenta jak i w sytuacjach codziennych (zakupy, podróże, restauracje, hobby...). Zajęcia wzbogacone są słownictwem z zakresu sieci komputerowych LAN.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
opisuje w języku angielskim wybrane praktyczne zastosowania informatyki, w szczególności dotyczące sieci LAN	K_W03
Umiejętności (student potrafi)	
porozumiewa się w prostych lub złożonych sytuacjach zawodowych informatyka, wykorzystując terminologię specjalistyczną oraz techniki komunikacji ustnej i pisemnej	K_U14
uczestniczy w dyskusji, przedstawia stanowisko i reaguje na opinie innych osób w języku angielskim	K_U15
formułuje wypowiedzi ustne i pisemne w języku angielskim na poziomie właściwym dla danego semestru, z wykorzystaniem słownictwa ogólnego i informatycznego	K_U16
współpracuje w parach i zespołach podczas realizacji zadań językowych oraz sytuacyjnych związanych z pracą w środowisku IT	K_U18
Kompetencje społeczne (student jest gotów do)	
jest gotów oceniać poprawność własnych wypowiedzi i krytycznie analizować otrzymywaną informację zwrotną	K_K01
jest gotów rozwijać kompetencje językowe z wykorzystaniem kursu e-learningowego, źródeł elektronicznych i materiałów anglojęzycznych	U_K07

Metody i formy kształcenia:

- Metoda ćwiczeń
- Metoda sytuacyjna
- Praca ze źródłem elektronicznym

Metody oceniania:

- Zaliczenie ustne
- Zaliczenie testowe
- Częstkowe sprawdziany pisemne - kolokwia

Ćwiczenia (Ć)

- Słownictwo dotyczące zdrowia, części ciała, lekarstwa; słownictwo wyrażające emocje
- Zastosowanie czasu przyszłego Future Simple

3. Poznanie form czasownika (bezokolicznik, gerund) oraz czasowników modalnych;
4. Zdania warunkowe – typ 1 (przyszłość)
5. Lokalne sieci komputerowe LAN: network devices and topology, describing LAN problems, giving troubleshooting instructions, reporting connectivity issues, explaining security rules in a local network.

Literatura podstawowa

1. ENGLISH FILE 4th edition (C. Latham-Koenig, C. Oxenden, J. Lambert, P. Seligson) Oxford University Press, 2019
2. B. Błaszczak, English 4 IT. Praktyczny kurs języka angielskiego dla specjalistów IT i nie tylko. Helion 2017
3. Cisco, What is a LAN? (<https://www.cisco.com/site/us/en/learn/topics/networking/what-is-a-lan-local-area-network.html>)
4. Cisco, What is Network Topology? (<https://www.cisco.com/site/us/en/learn/topics/networking/what-is-network-topology.html>)
5. Cisco, What is Network Troubleshooting? (<https://www.cisco.com/site/us/en/learn/topics/networking/what-is-network-troubleshooting.html>)

Literatura uzupełniająca

1. ENGLISH FILE 4th edition (C. Latham-Koenig, C. Oxenden, J. Lambert, P. Seligson) Oxford University Press, 2019 Workbook

Źródła dodatkowe

1. Platforma edukacyjna wlodek.edu.pl
2. BBC portal (www.bbc.co.uk)
3. Microsoft Support, Fix Wi-Fi connection issues in Windows (<https://support.microsoft.com/en-US/Windows/Experience/Connectivity-Networking/fix-wi-fi-connection-issues-in-windows>)
4. Cloudflare, What is a LAN? (<https://www.cloudflare.com/en-gb/learning/network-layer/what-is-a-lan/>)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	20 godz.....	16 godz.
• Samodzielne wykonanie zadań.....	10 godz.....	10 godz.
• Przygotowanie się do zajęć.....	10 godz.....	10 godz.
• Realizacja kursu e-learningowego.....	10 godz.....	14 godz.
Suma obciążeń:.....	50 godz.....	50 godz.

Zaliczenie zajęć:

Oceniając każdą pracę studenta (zarówno ustną jak i pisemną) zastosowanie mają poniższe progi procentowe:

- 91%–100%: ocena bardzo dobra (5.0)
- 81%–90%: ocena dobra plus (4.5)
- 71%–80%: ocena dobra (4.0)
- 61%–70%: ocena dostateczna plus (3.5)
- 51%–60%: ocena dostateczna (3.0)
- 50% i poniżej: niedostateczna (2.0) – zajęcia niezaliczone.

Warunkiem koniecznym do zaliczenia semestru jest uzyskanie pozytywnych wyników z wszystkich elementów.

Język angielski IV

Nazwa zajęć po angielsku: English in practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 2

Materiały: kurs e-learningowy

Język zajęć: angielski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia ogólnouczelniane

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Ćwiczenia.....30 / 30 godz..... 10 / 14 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Małgorzata Piątek

Opis zajęć:

Zajęcia z języka angielskiego przeznaczone są dla studentów pragnących doskonalić swoje umiejętności z języka angielskiego na poziomie B2. Językiem prowadzenia zajęć jest język angielski z zastosowaniem języka polskiego w wyjątkowych sytuacjach podczas zajęć. Celem zajęć jest doskonalenie zastosowania języka angielskiego w środowisku profesjonalnym każdego studenta jak i w sytuacjach codziennych (zakupy, podróże, restauracje, hobby...). Zajęcia wzbogacone są słownictwem z zakresu informatyki - internetu, stron www oraz podstaw cyberbezpieczeństwa.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
opisuje w języku angielskim wybrane praktyczne zastosowania informatyki, w szczególności dotyczące Internetu, stron WWW oraz podstaw cyberbezpieczeństwa	K_W03
Umiejętności (student potrafi)	
porozumiewa się w prostych lub złożonych sytuacjach zawodowych informatyka, wykorzystując terminologię specjalistyczną oraz techniki komunikacji ustnej i pisemnej	K_U14
uczestniczy w dyskusji, przedstawia stanowisko i reaguje na opinie innych osób w języku angielskim	K_U15
formułuje wypowiedzi ustne i pisemne w języku angielskim na poziomie właściwym dla danego semestru, z wykorzystaniem słownictwa ogólnego i informatycznego	K_U16
współpracuje w parach i zespołach podczas realizacji zadań językowych oraz sytuacyjnych związanych z pracą w środowisku IT	K_U18
Kompetencje społeczne (student jest gotów do)	
jest gotów oceniać poprawność własnych wypowiedzi i krytycznie analizować otrzymywaną informację zwrotną	K_K01
jest gotów rozwijać kompetencje językowe z wykorzystaniem kursu e-learningowego, źródeł elektronicznych i materiałów anglojęzycznych	U_K07

Metody i formy kształcenia:

- Metoda ćwiczeń
- Metoda sytuacyjna
- Praca ze źródłem elektronicznym

Metody oceniania:

- Zaliczenie ustne
- Zaliczenie testowe
- Częstkowe sprawdziany pisemne - kolokwia

Ćwiczenia (Ć)

- Słownictwo opisujące zwierzęta, sporty, środowisko szkolne i zawodowe, narodowości

2. Zdanie warunkowe – typ 2 (teraźniejszość, udzielanie rady)
3. Zastosowanie strony bierniej czasownika
4. Wyrażenia *used to*, *might*
5. Konstrukcje z *'so'* i *'neither'*
6. Internet i strony www: websites and web applications, user interface and user experience, describing website functionality, presenting a web project, giving feedback on a digital product.
7. Podstawy cyberbezpieczeństwa: recognising common cyber threats (phishing, fake websites, malware, data leaks, identity theft), discussing website security (HTTPS, certificates, secure login forms, privacy settings, cookie notices), writing short security warnings and user instructions.

Literatura podstawowa

1. ENGLISH FILE 4th edition (C. Latham-Koenig, C. Oxenden, J. Lambert, P. Seligson) Oxford University Press, 2019
2. B. Błaszczuk, English 4 IT. Praktyczny kurs języka angielskiego dla specjalistów IT i nie tylko. Helion 2017
3. MDN Web Docs, How the web works (https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Web_standards/How_the_web_works)
4. MDN Web Docs, Website security (https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/First_steps/Website_security)
5. Cybersecurity & infrastructure Security Agency, Secure Our World (<https://www.cisa.gov/secure-our-world>)

Literatura uzupełniająca

1. ENGLISH FILE 4th edition (C. Latham-Koenig, C. Oxenden, J. Lambert, P. Seligson) Oxford University Press, 2019 Workbook
2. BBC portal (www.bbc.co.uk)

Źródła dodatkowe

1. Platforma edukacyjna wlodek.edu.pl
2. Nielsen Norman Group, The Definition of User Experience (<https://www.nngroup.com/articles/definition-user-experience/>)
3. FTC Consumer Advice, Online Privacy and Security (<https://consumer.ftc.gov/identity-theft-and-online-security/online-privacy-and-security>)
4. Open Worldwide Application Security Project, Top 10 (<https://owasp.org/Top10/>)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	20 godz.....	16 godz.
• Samodzielne wykonanie zadań.....	10 godz.....	10 godz.
• Przygotowanie się do zajęć.....	10 godz.....	10 godz.
• Realizacja kursu e-learningowego.....	10 godz.....	14 godz.
Suma obciążeń:.....	50 godz.....	50 godz.

Zaliczenie zajęć:

Oceniając każdą pracę studenta (zarówno ustną jak i pisemną) zastosowanie mają poniższe progi procentowe:

- 91%–100%: ocena bardzo dobra (5.0)
- 81%–90%: ocena dobra plus (4.5)
- 71%–80%: ocena dobra (4.0)
- 61%–70%: ocena dostateczna plus (3.5)
- 51%–60%: ocena dostateczna (3.0)
- 50% i poniżej: niedostateczna (2.0) – zajęcia niezaliczone.

Warunkiem koniecznym do zaliczenia semestru jest uzyskanie pozytywnych wyników z wszystkich elementów.

Technologia informacyjna I

Nazwa zajęć po angielsku: Information Technology I

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: I

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: kurs e-learningowy

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia ogólnouczelniane

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....11 / 6 godz.
- Ćwiczenia.....30 / 16 godz.....20 / 10 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot rozwija praktyczne kompetencje studentów w zakresie świadomego, bezpiecznego i legalnego wykorzystywania technologii informacyjnych w pracy akademickiej i zawodowej. Szczególnie skierowany jest do osób, które wykonują prace polegające na zbieraniu, obróbce i prezentacji danych. Materiał do zajęć został opublikowany jako kurs e-learningowy. W ramach kursu oraz prowadzonych zajęć student zdobywa informacje teoretyczne oraz umiejętności praktyczne z dziedziny wykorzystania oprogramowania pracującego „w chmurze” w procesie gromadzenia i przetwarzania informacji. Studenci zdobędą wiedzę na temat pracy z informacjami, legalności wykorzystania danych z Internetu, zespołowej pracy w aplikacjach chmurowych, opracowania danych zastanych oraz wywołanych, z zakresu wizualizacji danych, opracowania kwestionariusza ankiety, opracowania i analizy danych, przygotowania raportu i prezentacji. Na zajęciach praktycznych kształtowane są umiejętności w zakresie wyszukiwania i przetwarzania danych, tworzenia dokumentów tekstowych, analizowania danych w arkuszach kalkulacyjnych, właściwego zastosowania barw, typografii oraz zasad wizualizacji danych, realizacji badań CAWI, przygotowywania rozbudowanych raportów, projektowania i wykonania prezentacji zgodnie z przyjętym scenariuszem, wystąpienia publicznego. Aktywne uczestnictwo w zajęciach wymaga posiadania sprzętu komputerowego (nie wystarczy smartfon) z dostępem do Internetu oraz konto Google. Zajęcia nie powielają materiału ze szkoły średniej, ale wnoszą umiejętności studentów na wyższy poziom, przydatny w trakcie studiów oraz pracy zawodowej.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
zna i opisuje pojęcia dotyczące danych, informacji oraz wiedzy; organizuje proces zbierania informacji oraz korzystania z danych zastanych z zakresu informatyki oraz dyscyplin pokrewnych	K_W02
wyjaśnia praktyczne zastosowania technologii informacyjnej w pracy akademickiej i zawodowej informatyka	K_W03
dobiera narzędzia chmurowe, edytorskie, arkuszowe, ankietowe i prezentacyjne do realizacji zadań związanych z przetwarzaniem danych	K_W05
opisuje zasady prawa autorskiego, prawa cytatu i korzystania z utworów rozpowszechnianych na licencjach Creative Commons	K_W09
Umiejętności (student potrafi)	
pozyskuje dane z różnych źródeł i wykorzystuje je do rozwiązania problemu z zakresu technologii informacyjnej	K_U01
dobiera źródła, selekcjonuje informacje oraz ocenia ich wiarygodność i przydatność	K_U03
posługuje się technikami informacyjno-komunikacyjnymi do realizacji zadań związanych z przetwarzaniem i prezentacją danych do realizacji zadań typowych dla działalności	K_U04

inżynierskiej w dziedzinie informatyki	
tworzy, edytuje i formatuje dokumenty, przetwarza dane w arkuszu, opracowuje raport i prezentuje wyniki	K_U05
komunikuje wyniki pracy w formie raportu, prezentacji i wystąpienia	K_U14
przedstawia wnioski, uzasadnia je i odpowiada na pytania odbiorców	K_U15
planuje i realizuje własne uczenie się z wykorzystaniem kursu e-learningowego	K_U19
Kompetencje społeczne (student jest gotów do)	
akceptuje konieczność krytycznego podejścia do zasobów informacyjnych, szczególnie zdobytych z niesprawdzonych źródeł internetowych	K_K01
gotów jest do rozwiązywania problemów poznawczych i praktycznych z zakresu przetwarzania danych	K_K02
jest gotów do przestrzegania zasad etyki zawodowej, prawa autorskiego i legalnego korzystania z zasobów cyfrowych.	K_K05
rozumie potrzebę ciągłego samorozwoju oraz wykorzystania w tym celu ogólnodostępnych masowych kursów e-learningowych MOOC	U_K07
rozumie potrzebę ciągłego samorozwoju oraz wykorzystania w tym celu ogólnodostępnych masowych kursów e-learningowych MOOC	U_K07

Metody i formy kształcenia:

- Praca ze źródłem elektronicznym
- Metoda sytuacyjna
- Dyskusja
- Metoda ćwiczeń
- Film dydaktyczny
- Metoda pokazu

Metody oceniania:

- Zaliczenie testowe
- Częstkowe sprawdziany pisemne - kolokwia
- Ocena aktywności w kursie

Wykład konwersatoryjny (W)

1. Dane - informacja - wiedza. Skale prezentacji danych.
2. Źródła informacji. Luka informacyjna. Praca z danymi zastanymi. Wyszukiwanie danych.
3. Korzystanie z informacji. Prawo autorskie. Licencje.
4. Praca w aplikacjach chmurowych. Wspólna praca z dokumentami.
5. Opisywanie danych. Znaki interpunkcyjne. Kroje pisma. Parametry tekstu.
6. Opracowanie danych. Praca w arkuszu kalkulacyjnym.
7. Praca z danymi. Tabele przestawne.
8. Projektowanie wizualne - zasady przetwarzania informacji wizualnej, wykorzystanie koloru, graficzna prezentacja danych.
9. Zbieranie opinii - budowa kwestionariusza, typy pytań, dostęp warunkowy do grup pytań, dystrybucja formularza (CAWI)
10. Raporty - cechy efektywnych raportów, układ dokumentu, skład dokumentu.
11. Prezentacja danych - rodzaje prezentacji, scenariusze, układ slajdu, wizualne rozmieszczenie elementów, zapewnienie czytelności informacji.
12. Wystąpienie - przygotowanie narracji, storytelling, składowe wystąpienia, walka z tremą.

Ćwiczenia (Ć)

1. Wyszukiwanie proste i zaawansowane danych w internecie.
2. Praca w chmurze - zakładanie konta, konfigurowanie dysku, dodawanie nowych aplikacji.
3. Przygotowanie dokumentu tekstowego.
4. Praca w arkuszu kalkulacyjnym - adresowanie, formuły, formatowanie.
5. Zaawansowane praca z danymi i agregowanie danych przy pomocy tabel przestawnych.
6. Dobór wykresów, barw oraz elementów dodatkowych w celu właściwej wizualizacji danych.

7. Przygotowanie i udostępnienie formularza ankiety.
8. Obróbka wyników badań z zastosowaniem tabeli przestawnej.
9. Budowa raportu - szpalty, formatowanie, spis treści, atrakcyjność wizualna.
10. Opracowanie prezentacji - wybór scenariusza, dobór struktury slajdów.
11. Tekst i grafika na slajdach. Wprowadzenie animacji.
12. Wystąpienie - przygotowanie wystąpienia obudowanego prezentacją.

Literatura podstawowa

1. Kurs "Opracowanie i prezentacja danych" Portal Navoica/Włodek

Literatura uzupełniająca

1. M. Makowska, Analiza danych zastanych. Przewodnik dla studentów, WN Scholar Warszawa 2018
2. A. J. Blikle, J. Dominet, Komputerowa edycja dokumentów dla średnio zaawansowanych, Helion 2021
3. 101 tricków, dzięki którym w pełni wykorzystasz potencjał arkuszy Google. fly on the Cloud, ebook
4. C. Wilke, Podstawy wizualizacji danych. Zasady tworzenia atrakcyjnych wykresów. Gliwice, Helion 2020
5. C. N. Knaflic, Storytelling danych. Poradnik wizualizacji danych dla profesjonalistów. Gliwice, Helion Onepress 2019
6. E. Babbie, Badania społeczne w praktyce, PWN Warszawa, 2005
7. M. Netzley, C. Snow, Pisanie raportów. Wolters Kluwer Polska Warszawa, 2009
8. R. Williams, DTP od podstaw. Projekty z klasą, Wydanie IV, Helion Gliwice, 2016
9. P. Bucki, Prezentacje po prostu! Warszawa PWN 2020
10. N. Duarte, Slajd:ologia. Nauka i sztuka tworzenia genialnych prezentacji. Gliwice, Helionm 2021
11. L. Buksak, Szkoła mówców. Helion Gliwice, 2019

Źródła dodatkowe

1. <https://creativecommons.org/licenses/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	14 godz.....	8 godz.
• Realizacja kursu e-learningowego.....	30 godz.....	30 godz.
• Samodzielne wykonanie zadań.....	20 godz.....	23 godz.
• Przygotowanie się do kolokwium.....	11 godz.....	14 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia zajęć jest zrealizowanie z powodzeniem kursu e-learningowego oraz wykonanie zadanych zadań praktycznych w trakcie trzech kolokwium. Wynik realizacji kursu wyznaczony jest na podstawie odpowiedzi na pytania kontrolne oraz z testu końcowego. Wyniki kolokwium oceniane są punktowo za każde zadanie praktyczne. Ocena końcowa wyliczana jest jako średnia ważona w 40% z wyniku kursu oraz w 60% z wykonania ćwiczeń praktycznych, przy czym obie składowe muszą mieć uzyskane ponad połowę możliwych punktów (czyli być zaliczone). Ocena końcowa wyliczana jest zgodnie z następującym algorytmem:

0% - 50% - niedostateczna
 51%-60% - dostateczna
 61%-70% - dostateczna plus
 71%-80% - dobra
 81%-90% - dobra plus
 91%-100% - bardzo dobra.

Technologia informacyjna II

Nazwa zajęć po angielsku: Information Technology II

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: II

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: kurs e-learningowy + materiały

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia ogólnouczelniane

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Projekt.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....20 / 10 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot „Technologia informacyjna II” skierowany jest do studentów drugiego semestru studiów inżynierskich na kierunku Informatyka o profilu praktycznym. Celem zajęć jest rozwijanie umiejętności praktycznego posługiwania się narzędziami grafiki komputerowej (bitmapowej i wektorowej), edycji materiałów cyfrowych (w tym z wykorzystaniem sztucznej inteligencji), przygotowywania grafiki do zastosowań w projektowaniu i druku 3D. Zajęcia ćwiczeniowe opierają się na realizacji zadań z wykorzystaniem darmowego oprogramowania np. Gimp, Inkscape, Canva. W ramach zajęć projektowych studenci poznają zagadnienia druku trójwymiarowego, samodzielnie projektują elementy przestrzenne (np. Tinkercad), przygotowują je do druku przy pomocy programów typu slicer (np. FlashPrint) i drukują na drukarce 3D. Do zajęć przygotowano kurs e-learningowy na temat grafiki komputerowej oraz udostępniono materiały dotyczące funkcjonowania drukarek.

Przedmiot łączy aspekty twórcze i techniczne, rozwijając nie tylko kompetencje zawodowe, ale też postawy współpracy, odpowiedzialności i komunikacji – szczególnie w części projektowej.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia praktyczne zastosowania grafiki rastrowej, wektorowej oraz modelowania i druku 3D w projektach informatycznych	K_W03
rozdziela standardy i formaty plików graficznych oraz dobiera je do publikacji internetowej, edycji i przygotowania wydruku	K_W06
<i>Umiejętności (student potrafi)</i>	
dobiera i stosuje narzędzia informatyczne do retuszu i kompozycji obrazów oraz przygotowywania grafiki zgodnie z jej przeznaczeniem	K_U04
projektuje model 3D, dobiera parametry jego przygotowania i druku, wykonuje wydruk próbny, analizuje uzyskany rezultat oraz wprowadza konieczne poprawki	K_U07
integruje narzędzia AI w procesie twórczym (prompt-engineering, edycja, upscaling), ocenia ich jakość oraz ryzyko prawno-etyczne.	K_U08
współpracuje z innymi osobami przy definiowaniu potrzeb interesariusza, projektowaniu rozwiązania i prezentacji efektu	K_U18
<i>Kompetencje społeczne (student jest gotów do)</i>	
jest gotów realizować projekt na rzecz społeczności akademickiej lub interesariusza wewnętrznego/zewnętrznego	K_K03
jest gotów przestrzegać zasad legalnego i etycznego korzystania z zasobów graficznych, modeli 3D i narzędzi AI	K_K05

Metody i formy kształcenia:

- Pogadanka
- Praca ze źródłem elektronicznym
- Dyskusja
- Metoda sytuacyjna
- Metoda ćwiczeń
- Metoda pokazu
- Metoda projektu

Metody oceniania:

- Ocena aktywności w kursie
- Zaliczenie praktyczne
- Rozliczenie projektu

Ćwiczenia (Ć)

1. Formaty grafiki 2D: PNG, JPG, SVG, WEBP i inne - zastosowania, zalety i ograniczenia
2. Grafika bitmapowa - operacje geometryczne (skalowanie, obrót, kadrowanie), korekcja tonalna, warstwy, selekcje, maski, retusz nieniszczący, fotomontaż, eksport pliku.
3. AI dla grafików - generowanie obrazów, edycja obrazów (upscaling, kolorowanie), generowanie krótkich materiałów wideo na podstawie obrazu oraz promptu.
4. Grafika wektorowa - obiekty, ścieżki, węzły, typografia, eksport do publikacji.

Projekt (P)

1. Budowa drukarek 3D, materiały, zasady bezpieczeństwa.
2. Definiowanie potrzeb interesariusza wewnętrznego lub zewnętrznego.
3. Modelowanie obiektu (konstrukcja, łączenia, tolerancje, ergonomia).
4. Przygotowanie obiektu do druku (slicer, podpory, wypełnienie, podgląd).
5. Testowy wydruk i iteracje, analiza usterek, korekty, wydruk końcowy.
6. Dokumentacja i prezentacja projektu.

Literatura podstawowa

1. Von Glitschka, Grafika wektorowa. Szkolenie podstawowe, Helion 2012
2. Powers S., Grafika w Internecie, Helion 2009

Literatura uzupełniająca

1. Gajda W., Gimp praktyczne projekty, Helion 2012
2. Cieśla K., Inkscape. Podstawowa obsługa programu, Helion 2021

Źródła dodatkowe

1. <https://www.gov.pl/web/koduj/tinkerkad>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	25 godz.....	14 godz.
• Realizacja kursu e-learningowego.....	15 godz.....	15 godz.
• Samodzielne wykonanie zadań.....	10 godz.....	14 godz.
• Przygotowanie projektu.....	15 godz.....	17 godz.
• Przygotowanie się do zaliczenia końcowego.....	10 godz.....	15 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest przyswojenie wiadomości, zdobycie umiejętności oraz nabycie kompetencji społecznych związanych z grafiką komputerową. Weryfikacja polega na udzieleniu odpowiedzi na pytania kontrolne w lekcjach kursu e-learningowego, wykonaniu zadań praktycznych, zaliczeniu praktycznego kolokwium, złożenie wydrukowanego modelu 3D oraz jego dokumentacji. Ocena końcowa zależy od liczby uzyskanych punktów z pytań kontrolnych i zadań (30%), wyniku zaliczenia praktycznego (40%) oraz z realizacji projektu, czyli prezentacji wydruku 3D (30%). Ocena końcowa wyliczana jest ze średniej ważonej i wynosi:

0% - 50% - niedostateczny

51%-60% - dostateczny

61%-70% - dostateczny plus

71%-80% - dobry

81%-90% - dobry plus

91%-100% - bardzo dobry.

ABC studiowania

Nazwa zajęć po angielsku: Study Guide

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: I

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 2

Materiały: kurs e-learningowy

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia ogólnouczelniane

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....11 / 6 godz.
- Ćwiczenia.....15 / 8 godz.....11 / 6 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot "ABC studiowania" przeznaczony jest dla osób, które właśnie rozpoczynają studia i często nie wiedzą jakie prawa i obowiązki wynikają z tego faktu. Celem przedmiotu jest dostarczenie informacji na temat regulaminów obowiązujących w Uczelni, kodeksu etyki studenta, możliwości uzyskania stypendiów, korzystania z Biblioteki, zasad bezpieczeństwa i higieny studiowania, jakości kształcenia, poszanowania praw autorskich oraz własności intelektualnej, kształcenia na odległość, wykorzystania udostępnionych przez SWPW narzędzi internetowych: platformy e-learningowej WLODEK oraz systemu Wirtualnej Uczelni. Informacje teoretyczne przeplatane są ćwiczeniami, które mają na celu wykształcenia umiejętności studiowania oraz radzenia sobie w trudnych sytuacjach występujących na uczelni. Po ukończeniu kursu studenci będą znać podstawowe zasady obowiązujące w trakcie studiów - nie tylko swoje obowiązki, ale również przysługujące im prawa.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
opisuje zasady i normy etyczne obowiązujące studentów oraz ich znaczenie dla odpowiedzialnego funkcjonowania w społeczności akademickiej	K_W08
wyjaśnia zasady ochrony własności intelektualnej, prawa cytatu oraz legalnego korzystania z prac innych autorów w trakcie studiowania i pracy zawodowej	K_W09
wyjaśnia zasady funkcjonowania szkoły wyższej oraz wskazuje podstawowe akty prawne obowiązujące studentów i pracowników uczelni w Polsce oraz w SWPW	U_W11
Umiejętności (student potrafi)	
dobiera źródła i informacje potrzebne do rozwiązywania problemów związanych z organizacją własnego procesu uczenia się	K_U03
współpracuje z innymi studentami w trakcie realizacji zadań grupowych, przyjmując różne role w kolejnych zadaniach	K_U18
planuje i organizuje własny proces uczenia się z wykorzystaniem materiałów w kursie e-learningowym, biblioteki i źródeł internetowych	K_U19
rozpoznaje potrzebę adaptacji do zmieniającego się rynku pracy i wykorzystuje kursy e-learningowe do rozwoju osobistego i zawodowego	U_U20
Kompetencje społeczne (student jest gotów do)	
jest gotów krytycznie oceniać własną wiedzę, informacje z Internetu oraz poprawność wykorzystania źródeł	K_K01
jest gotów podejmować wyzwania związane ze studiowaniem, realizować zadania	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Praca ze źródłem elektronicznym
- Metoda sytuacyjna
- Grywalizacja
- Metoda ćwiczeń
- Metoda pokazu

Metody oceniania:

- Częstkowe sprawdziany testowe - kolokwia
- Zaliczenie testowe
- Ocena aktywności w kursie

Wykład konwersatoryjny (W)

1. Edukacja na odległość i zasady pracy z kursem e-learningowym.
2. Organizacja procesu uczenia się na studiach.
3. Zasady korzystania z uczelnianej Biblioteki i źródeł elektronicznych.
4. Szkoła Wyższa im. Pawła Włodkowica w Płocku (SWPW) - historia i informacje podstawowe.
5. Normy prawne i etyczne studiowania w Polsce oraz w SWPW.
6. System informatyczny Wirtualna Uczelnia i komunikacja formalna z uczelnią.
7. System Zapewniania Jakości Kształcenia i udział studentów w ocenie jakości.
8. Pomoc materialna dla studentów.
9. Zasady bezpieczeństwa i higieny studiowania, ergonomia pracy przy komputerze.
10. Prawo autorskie oraz ochrona własności intelektualnej.

Ćwiczenia (Ć)

1. Korzystanie z kursów umieszczonych na uczelnianej platformie WLODEK. Realizacja testów, ćwiczeń i quizów internetowych. Przesyłanie prac zaliczeniowych na uczelniany serwer edukacyjny.
2. Organizacja nauki z użyciem kalendarzy, list zadań i notatek cyfrowych
3. Korzystanie z tradycyjnych i internetowych katalogów bibliotecznych. Dostęp do źródeł elektronicznych.
4. Wyszukiwanie informacji na stronie internetowej Uczelni.
5. Odnajdywanie informacji w Regulaminie studiów. Interpretacja przepisów prawnych obowiązujących w SWPW.
6. Sprawdzanie informacji w systemie Wirtualnej Uczelni. Sposoby komunikowania się oraz przekazywania decyzji studentom przez pracowników uczelni.
7. Wpływ studentów na jakość kształcenia w SWPW. Wypełnianie ankiet i zapoznanie się z wynikami.
8. Warunki uzyskania pomocy materialnej. Wypełnianie dokumentów umożliwiających uzyskanie stypendiów studenckich.
9. Ergonomia i bezpieczeństwo podczas nauki, pracy przy komputerze i realizacji zadań domowych. Pomoc osobom poszkodowanym.
10. Odnajdywanie i korzystanie z legalnych źródeł materiałów dostępnych w Internecie.

Literatura podstawowa

1. Kurs na platformie wlodek.wlodkowic.pl
2. Regulamin Studiów w SWPW
3. Kodeks etyki studenta SWPW
4. Regulamin Biblioteki SWPW

Literatura uzupełniająca

1. J. Półturzycki, Jak studiować zaocznie, WNN Novum 1995
2. Regulamin Antyplagiatowy SWPW

Źródła dodatkowe

1. Ustawa z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (aktualny tekst jednolity)
2. Ustawa z dnia 4 lutego 1994 r. O prawie autorskim i prawach pokrewnych (aktualny tekst jednolity)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego..... 8 godz..... 4 godz.
- Realizacja kursu e-learningowego..... 30 godz..... 30 godz.

- Zapoznanie się z literaturą przedmiotu.....3 godz..... 3 godz.
- Samodzielne wykonanie zadań..... 9 godz.....13 godz.
- Suma obciążeń:.....50 godz.....50 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest przyswojenie wiadomości, zdobycie umiejętności oraz nabycie kompetencji związanych z szeroko rozumianym procesem kształcenia się w szkole wyższej. Weryfikacja polega na udzieleniu odpowiedzi na pytania kontrolne w lekcjach kursu e-learningowego, wykonaniu zadań praktycznych, zaliczeniu trzech kolokwii testowych w ciągu semestru. Ocena końcowa zależy od liczby uzyskanych punktów z pytań kontrolnych i zadań (30%) oraz z kolokwii testowych (70%). Studenci mogą uzyskać dodatkowe punkty za aktywność podczas realizacji kursu - wykonywanie dodatkowych zadań udostępnianych okazjonalnie w trakcie całego semestru. Ocena końcowa wyliczana jest w stosunku do maksymalnej liczby możliwych do uzyskania punktów i wynosi:

0% - 50% - niedostateczny

51%-60% - dostateczny

61%-70% - dostateczny plus

71%-80% - dobry

81%-90% - dobry plus

91%-100% - bardzo dobry.

Osoby, które nie uzyskają zaliczenia w trakcie semestru, przystępują w trakcie sesji egzaminacyjnej do zaliczenia poprawkowego w formie testu.

Przygotowanie do praktyki zawodowej

Nazwa zajęć po angielsku: Preparation for professional practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: I

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 1

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia ogólnouczelniane

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny 15 / 8 godz 0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Zajęcia przygotowują studentów do realizacji obowiązkowej praktyki zawodowej, która na studiach o profilu praktycznym trwa sześć miesięcy. Studenci zapoznawani są z obowiązującymi przepisami prawa – Uchwałami Regulaminem Praktyk oraz Zasadami, warunkami i trybem zaliczenia w Szkole Wyższej im. Pawła Włodkowica w Płocku praktyki zawodowej poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego. Studenci poznają katalog rekomendowanych zagadnień do realizacji w trakcie praktyk zawodowych. Praktycznie uczą się wypełniać Oświadczenie o gotowości przystąpienia do praktyki, Dziennik praktyk, Wniosek o zaliczenie praktyki na podstawie doświadczenia zawodowego. Ponadto studenci zapoznają się z procesem adaptacji społeczno-zawodowej, przygotowują się do wejścia na rynek pracy oraz poznają zasady savoir-vivre'u oraz dress code'u w miejscu pracy (odbywania praktyki).

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wskazuje typowe miejsca odbywania praktyki na kierunku Informatyka oraz rozpoznaje praktyczne obszary zadań informatycznych możliwe do realizacji w podmiocie zewnętrznym	K_W03
wyjaśnia podstawowe przepisy i procedury regulujące realizację oraz zaliczenie praktyki zawodowej, w tym możliwość uznania efektów uczenia się na podstawie doświadczenia zawodowego	K_W08
opisuje zasady adaptacji społeczno-zawodowej, normy zachowania w miejscu pracy oraz podstawowe oczekiwania organizacyjne i komunikacyjne wobec praktykanta	U_W11
Umiejętności (student potrafi)	
dobiera właściwe źródła informacji i dokumenty dotyczące praktyki oraz wypełnia oświadczenie, dziennik praktyk i wniosek o zaliczenie praktyki na podstawie doświadczenia zawodowego	K_U03
planuje przebieg praktyki zawodowej, dobiera zagadnienia praktyczne zgodne z kierunkiem Informatyka i uzasadnia ich przydatność dla rozwoju zawodowego	K_U17
Kompetencje społeczne (student jest gotów do)	
jest gotów zasięgać opinii opiekuna praktyk lub przedstawiciela podmiotu zewnętrznego w przypadku trudności z wyborem miejsca praktyki, zakresem zadań lub dokumentacją	K_K02
jest gotów przestrzegać zasad etyki zawodowej informatyka, kultury organizacyjnej, savoir-vivre'u oraz zasad odpowiedzialnego zachowania w miejscu praktyki	K_K05

Metody i formy kształcenia:

- Wykład informacyjny

Metody oceniania:

- Praca pisemna

- Dyskusja
- Metoda sytuacyjna
- Praca ze źródłem elektronicznym

- Zaliczenie testowe

Wykład konwersatoryjny (W)

1. Uwarunkowania prawne i organizacyjne praktyki zawodowej na studiach o profilu praktycznym.
2. Dokumentacja przebiegu realizacji praktyki zawodowej i warunków jej zaliczenia.
3. Analiza podmiotów (firm, instytucji, placówek, ośrodków), w których można realizować praktyki zawodowe.
4. Katalog rekomendowanych zagadnień do realizacji w trakcie praktyk zawodowych.
5. Procedura zaliczenia praktyki poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego.
6. Etyka zawodowa informatyka, poufność danych i odpowiedzialność za informacje uzyskane podczas praktyki.
7. Adaptacja społeczno-zawodowa, przygotowanie do wejścia na rynek pracy, zasady savoir-vivre'u oraz dress code'u w miejscu pracy.

Literatura podstawowa

1. Uczelniany Regulamin Studenckich Praktyk Zawodowych obowiązujący studentów Szkoły Wyższej im. Pawła Włodkowica w Płocku
2. Zasady, warunki i tryb zaliczenia w SWPW w Płocku praktyki zawodowej poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego
3. Katalog rekomendowanych zagadnień do rozpoznania praktycznego w trakcie praktyk zawodowych na kierunku Informatyka

Literatura uzupełniająca

1. Zarządzenie nr 06/2023 Rektora SWPW w Płocku z dnia 13 kwietnia 2023 r. w sprawie realizacji zadań Uczelnianego Pełnomocnika ds. praktyk w Szkole Wyższej im. Pawła Włodkowica w Płocku
2. Zasad oceny realizacji praktyk przez studentów SWPW w Płocku

Źródła dodatkowe

1. Materiały na platformie WLODEK dotyczące praktyk zawodowych

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	15 godz.....	8 godz.
• Zapoznanie się z literaturą przedmiotu.....	3 godz.....	5 godz.
• Samodzielne wykonanie zadań.....	4 godz.....	8 godz.
• Przygotowanie się do zaliczenia końcowego.....	3 godz.....	4 godz.
Suma obciążeń:.....	25 godz.....	25 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest przyswojenie wiadomości, zdobycie umiejętności oraz nabycie kompetencji społecznych z zakresu realizacji studenckich praktyk zawodowych oraz adaptacji społeczno-zawodowej. Weryfikacja polega na rozwiązaniu testu sprawdzającego wiedzę oraz faktycznym wypełnieniu dokumentacji przebiegu praktyki i dokumentacji zaliczenia praktyki. Ocena końcowa zależy od liczby uzyskanych punktów z zaliczenia teoretycznego (50%) oraz z pracy praktycznej (50%). Ocena końcowa zależy od liczby uzyskanych punktów i wynosi:

0% - 50% - niedostateczny
 51%-60% - dostateczny
 61%-70% - dostateczny plus
 71%-80% - dobry
 81%-90% - dobry plus
 91%-100% - bardzo dobry.

Work in an international environment

Nazwa zajęć po angielsku: Work in an international environment

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 2

Materiały: kurs e-learningowy

Język zajęć: angielski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia ogólnouczelniane

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....15 / 8 godz.
- Ćwiczenia.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr Kazimierz Waluch

Opis zajęć:

Zajęcia mają przygotować studentów do funkcjonowania w zróżnicowanym kulturowo i zmiennym środowisku międzynarodowym, zarówno w kontekście zawodowym, jak i społecznym. Wykład realizowany jest w formie kursu e-learningowego, po zapoznaniu się z którym student będzie wiedział: jakie mechanizmy mają wpływ na rozwój gospodarki światowej; jak działają firmy na rynku międzynarodowym; jaka jest specyfika zarządzania organizacjami międzynarodowymi; na czym polegają różnice kulturowe; jak się komunikować i negocjować z przedstawicielami innych kultur. Ćwiczenia nastawione są na kształtowanie umiejętności uwzględniania różnic kulturowych w funkcjonowaniu w środowisku międzynarodowym.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Student charakteryzuje zróżnicowanie struktur społeczno-ekonomicznych, ze szczególnym uwzględnieniem czynnika kulturowego.	K_W07
Student wyjaśnia kulturowe uwarunkowania funkcjonowania człowieka w społeczeństwie i ich wpływ na jego działalność gospodarczą.	K_W08
Umiejętności (student potrafi)	
Student określa specyfikę kulturową środowiska międzynarodowego i rozwiązuje problemy z nią związane dokonując doboru, selekcji i krytycznej analizy źródeł.	K_U03
Student posługuje się językiem angielskim w typowych sytuacjach zawodowych w zespołach międzynarodowych, w szczególności podczas komunikacji projektowej, omawiania dokumentacji technicznej oraz kontaktu z klientem lub użytkownikiem systemu informatycznego.	K_U16
Student interpretuje zjawiska kulturowe i ich wpływ na życie gospodarcze i zarządzanie.	U_U20
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny własnych założeń, stereotypów i sposobu komunikacji w międzynarodowym środowisku pracy.	K_K01
Student rozumie potrzebę pogłębiania wiedzy na temat uwarunkowań kulturowych środowiska międzynarodowego, zwłaszcza w kontekście procesów globalizacji.	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Metoda ćwiczeń
- Metoda sytuacyjna

Metody oceniania:

- Aktywność w trakcie zajęć
- Praca domowa
- Zaliczenie testowe

Wykład konwersatoryjny (W)

1. International economics
2. Motives and strategies of internationalization
3. Management of the organization on the international market
4. Cultural conditions of activity on the international market
5. Intercultural communication

Ćwiczenia (Ć)

1. Cultural similarities and differences
2. Communication in an international environment (Practical communication skills in an international environment (e.g. professional email, meeting notes, case study).
3. Culture shock
4. Culture, cultural differences, stereotypes and prejudices
5. Values and customs
6. Working in an international environment in the IT sector

Literatura podstawowa

1. R. Zenderowski, B. Koziński, Różnice kulturowe w biznesie, CeDeWu, Warszawa 2022
2. S. Huntington, Zderzenie cywilizacji, Wydawnictwo Zysk i S-ka, Warszawa 2018

Literatura uzupełniająca

1. M. Budzanowska-Drzewiecka, A. Marcinkowski, A. Motyl-Adamczyk, Różnice kulturowe w komunikacji biznesowej, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2016
2. F. Trompenaars, Siedem wymiarów kultury: znaczenie różnic kulturowych w działalności gospodarczej, Oficyna Ekonomiczna, Kraków 2002
3. Ch. Hampden-Turner, F. Trompenaars, Siedem kultur kapitalizmu, Wolters Kluwer, Warszawa 2012
4. G. Hofstede, G.J. Hofstede, M. Minkov, Kultury i organizacje, PWE, Warszawa 2011
5. B. Błaszczyk, English 4 IT. Praktyczny kurs języka angielskiego dla specjalistów IT i nie tylko. Helion 2022

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	15 godz.....	8 godz.
• Zapoznanie się z literaturą przedmiotu.....	5 godz.....	8 godz.
• Realizacja kursu e-learningowego.....	15 godz.....	15 godz.
• Samodzielne wykonanie zadań.....	8 godz.....	12 godz.
• Przygotowanie się do zaliczenia końcowego.....	7 godz.....	7 godz.
Suma obciążeń:.....	50 godz.....	50 godz.

Zaliczenie zajęć:

Warunkiem uzyskania pozytywnej oceny jest przyswojenie informacji na temat wpływu różnic kulturowych na działalność człowieka w środowisku międzynarodowym, procesów kulturowych towarzyszących jego aktywności oraz konsekwencji wpływu czynnika kulturowego na zachowanie i funkcjonowanie rynków międzynarodowych. Warunkiem zaliczenia przedmiotu jest aktywny udział w zajęciach, wykonanie ćwiczeń oraz pozytywna ocena z testu

Analiza matematyczna i algebra liniowa

Nazwa zajęć po angielsku: Analysis and Linear Algebra

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: I

Forma zaliczenia: Egzamin

Punkty ECTS: 5

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia z zakresu nauk podstawowych

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr Piotr Sługocki

Opis zajęć:

Najważniejszym czynnikiem rozwoju społeczeństw jest wiedza. Aby ją nabywać potrzebna jest umiejętność myślenia abstrakcyjnego. Myślenie abstrakcyjne jest fundamentem matematyki - królowej nauk. W trakcie zajęć student nabywa umiejętności posługiwania się podstawowym aparatem analizy matematycznej i algebry liniowej. W pierwszej części student poznaje podstawowe definicje i twierdzenia analizy matematycznej dotyczące ciągów, szeregów i funkcji jednej zmiennej. W drugiej części student zdobywa wiadomości z algebry liniowej, w tym z rachunku macierzowego i układów równań liniowych. W wykładach i ćwiczeniach korzysta się z prezentacji Wolfram, które student może również samodzielnie uruchamiać mając dostęp do Internetu.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia pojęcia z zakresu matematyki niezbędną do zrozumienia, formułowania i rozwiązywania zadań z zakresu analizy matematycznej i algebry liniowej	K_W01
rozpoznaje zastosowania rachunku macierzowego i układów równań liniowych w problemach informatycznych	K_W03
Umiejętności (student potrafi)	
oblicza granice ciągów, granice funkcji, pochodne oraz ekstrema funkcji jednej zmiennej	K_U01
wykorzystuje narzędzia cyfrowe do wizualizacji funkcji, macierzy, przekształceń liniowych i wyników obliczeń	K_U04
rozwiązuje zadania z zakresu działań na macierzach, wyznaczników i układów równań liniowych	K_U05
planuje własną pracę i realizuje samodzielne uczenie się matematyki	K_U19
Kompetencje społeczne (student jest gotów do)	
jest gotów oceniać poprawność własnych rozwiązań i krytycznie odnosić się do otrzymanych wyników	K_K01
jest gotów korzystać ze wsparcia prowadzącego i źródeł wiedzy w przypadku trudności z rozwiązywaniem zadań matematycznych	K_K02

Metody i formy kształcenia:

- Wykład informacyjny
- Metoda ćwiczeń
- Wykład problemowy
- Metoda symulacyjna

Metody oceniania:

- Częstkowe sprawdziany testowe - kolokwia
- Egzamin opisowy
- Aktywność w trakcie zajęć

- Burza mózgów
- Dyskusja
- Praca ze źródłem elektronicznym

Wykład konwersatoryjny (W)

1. Ciągi liczbowe i szeregi
2. Granica ciągu i granica funkcji w punkcie
3. Ciągłość funkcji
4. Pochodna funkcji i jej interpretacja geometryczna i fizyczna
5. Funkcje różniczkowalne i twierdzenia ich dotyczące - obliczanie pochodnych funkcji
6. Twierdzenie Bolzano, Rolle'a i Lagrange'a o wartości średniej
7. Ekstrema
8. Twierdzenie Weierstrassa, Fermata
9. Pochodne wyższych rzędów
10. Interpretacje geometryczne i fizyczne pochodnych wyższych rzędów
11. Wypukłość i wklęsłość funkcji
12. Pojęcie macierzy
13. Przekształcenie liniowe na płaszczyźnie
14. Macierz, wyznacznik macierzy i jego interpretacje geometryczne
15. Układy dwóch i trzech równań liniowych i ich interpretacje geometryczne
16. Działania na macierzach. Macierz odwrotna

Ćwiczenia (Ć)

1. Obliczanie granic ciągów liczbowych
2. Obliczanie granic funkcji w punkcie
3. Badanie ciągłości funkcji
4. Obliczanie pochodnych funkcji
5. Znajdowanie wartości ekstremalnych funkcji
6. Konstruowanie wykresów funkcji
7. Obliczanie wyznaczników
8. Działania na macierzach
9. Diagram i macierz przejścia

Literatura podstawowa

1. Krywicki W., Włodarski L., Analiza matematyczna w zadaniach, cz. I, PWN, m.in. wyd.2004, druk 2022

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Samodzielne wykonanie zadań.....	28 godz.....	28 godz.
• Przygotowanie się do kolokwium.....	32 godz.....	40 godz.
• Przygotowanie i udział w egzaminie.....	20 godz.....	33 godz.
Suma obciążeń:.....	125 godz.....	125 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest uzyskanie pozytywnej oceny z dwóch kolokwium oraz egzaminu. Pierwsze kolokwium jest po połowie zajęć w semestrze, drugie na końcu semestru. Przed kolokwiami studenci otrzymują spis tematów, których będą dotyczyły zadania. Za pojedyncze zadanie student może otrzymać: 0, 0,25, 0,5, 0,75, 1 punktu. Przykładowa skala ocen, gdy można było uzyskać 7 punktów jest przedstawiona poniżej.

<3,5	2
3,5	3--
3,75-4,25	3
4,5-5	3,5
5,25-5,75	4
6-6,5	4,5
6,75-7	5

Na egzaminie sprawdzana jest praktyczna znajomość pojęć i twierdzeń z wykładów oraz konstruowanie wykresów funkcji. Sposób oceny jest podobny jak w przypadku kolokwium. Aktywność studenta na ćwiczeniach jest dodatkowo premiowana.

Podstawy metod probabilistycznych i statystyki

Nazwa zajęć po angielsku: Basic Concepts of Probability and Statistics

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: II

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia z zakresu nauk podstawowych

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

prof. dr hab. Jadwiga Suchecka, dr Piotr Sługocki

Opis zajęć:

W życiu nic nie jest pewne. Poznanie metod probabilistycznych (rachunku prawdopodobieństwa) pozwala lepiej to zrozumieć. W pierwszej części zajęć student poznaje podstawowe pojęcia, wzory i sposoby uzyskiwania rozwiązań problemów z zakresu metod probabilistycznych dotyczących zdarzeń losowych i zmiennych losowych. W drugiej części student zdobywa wiadomości z statystyki opisowej, w tym dotyczące opisu statystycznego próby losowej.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
ma wiedzę z zakresu matematyki niezbędną do formułowania i rozwiązywania typowych zadań z zakresu metod probabilistycznych, w tym dot. kombinatoryki i statystyki opisowej	K_W01
charakteryzuje zależności między prawdopodobieństwem warunkowym, niezależnością zdarzeń, wzorem Bayesa i interpretacją informacji niepewnej	K_W02
rozpoznaje zastosowania metod probabilistycznych i statystyki opisowej w analizie danych, modelowaniu i rozwiązywaniu zadań informatycznych	K_W05
Umiejętności (student potrafi)	
oblicza liczby możliwości z wykorzystaniem permutacji, wariacji i kombinacji oraz stosuje klasyczną definicję prawdopodobieństwa	K_U01
oblicza średnią, medianę, dominantę, kwartyle i miary zróżnicowania oraz analizuje opis statystyczny próby	K_U03
wykorzystuje arkusz kalkulacyjny lub inne narzędzie cyfrowe do obliczeń probabilistycznych i statystycznych oraz prezentacji wyników	K_U04
rozwiązuje zadania z zakresu prawdopodobieństwa warunkowego, niezależności zdarzeń, prawdopodobieństwa całkowitego i wzoru Bayesa	K_U07
pogłębienie analitycznego rozumowania i całościowej charakterystyki przetwarzanych danych	U_U20
Kompetencje społeczne (student jest gotów do)	
jest gotów oceniać poprawność własnych rozwiązań oraz krytycznie odnosić się do wyników obliczeń probabilistycznych i statystycznych.	K_K01
jest gotów rozwijać umiejętności analizy danych i samodzielnego uczenia się z wykorzystaniem materiałów cyfrowych	U_K07

Metody i formy kształcenia:

- Wykład problemowy
- Studium przypadku
- Metoda ćwiczeń
- Burza mózgów
- Wykład informacyjny

Metody oceniania:

- Częstkowe sprawdziany pisemne - kolokwia
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

1. Doświadczenie z jednakowo prawdopodobnymi wynikami
2. Prawdopodobieństwo geometryczne
3. Kombinatoryka. Permutacje. Wariacje. Kombinacje
4. Algebra zdarzeń. Definicja klasyczna prawdopodobieństwa
5. Prawdopodobieństwo warunkowe. Zdarzenia niezależne
6. Prawdopodobieństwo zupełne
7. Wzór Bayesa. Wnioskowanie bayesowskie
8. Zmienna losowa i jej rozkład. Średnia i odchylenie standardowe rozkładu
9. Zagadnienie Bernoulliego
10. Statystyka. Populacja generalna. Badanie reprezentatywne
11. Charakterystyki próby

Ćwiczenia (Ć)

1. Pytanie kawalera de Mere jako problem rachunku prawdopodobieństwa
2. Zagadnienia kombinatoryki. Grafy i drzewa. Wyznaczanie ilości możliwości
3. Działania na zbiorach zdarzeń
4. Wykorzystanie klasycznej definicji prawdopodobieństwa
5. Prawdopodobieństwo warunkowe
6. Zdarzenia niezależne i ich wykorzystanie w praktyce
7. Prawdopodobieństwo całkowite jako suma prawdopodobieństw
8. Wykorzystanie informacji - wzór Bayesa
9. Wyznaczanie wartości oczekiwanej, wariancji i odchylenia standardowego zmiennej losowej
10. Rozkłady Bernoulliego w przykładach z życia
11. Obliczanie średniej, mediany, dominanty i kwartyli

Literatura podstawowa

1. W. Kryszczyński, J. Bartos, W. Dyczka, K. Królikowska, M. Wasilewski, Rachunek prawdopodobieństwa i statystyka matematyczna w zadaniach, cz. 1, PWN 2012
2. A. Serwatka, Rachunek prawdopodobieństwa i statystyka. Wydawnictwo AGH im. St. Staszica, Kraków. OPEN AGH: <https://epodreczniki.open.agh.edu.pl/handbook/2369/module/2383/reader>

Literatura uzupełniająca

1. A. Plucińska, E. Pluciński, Probabilistyka: Statystyka matematyczna. Procesy stochastyczne. Rachunek prawdopodobieństwa. WNT 2000
2. S. Ostaniec, Z. Rusnak, U. Siedlecka, Statystyka. Elementy teorii i zadania. Wydawnictwo Akademii Ekonomicznej im. Oskara Langego we Wrocławiu 2006
3. D. Foryth, Probability and Statistics for Computer Science, Springer International Publishing, 2019

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....45 godz.....24 godz.
- Zapoznanie się z literaturą przedmiotu.....15 godz.....24 godz.
- Przygotowanie się do kolokwiów.....15 godz.....27 godz.
- Suma obciążeń:.....75 godz.....75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest uzyskanie pozytywnej oceny z dwóch kolokwiów. Pierwsze jest po połowie zajęć w semestrze, drugie na końcu semestru. Przed kolokwiami studenci otrzymują spis tematów, których będą dotyczyły

zadania. Za pojedyncze zadanie student może otrzymać: 0, 0,25, 0,5, 0,75, 1 punktu. Przykładowa skala ocen, gdy można było uzyskać 3 punkty jest przedstawiona poniżej.

=< 1,5	ndst
1,75	dst
2	dst+
2,25-2,5	db
2,5	db
2,75-3	bdb

Aktywność studenta na ćwiczeniach jest dodatkowo premiowana.

Matematyka dyskretna

Nazwa zajęć po angielsku: Discrete mathematics

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: III

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia z zakresu nauk podstawowych

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr Piotr Sługocki

Opis zajęć:

Istnieje ścisły związek między znajomością matematyki, a innowacyjnością. Wykłady i ćwiczenia zaznajamiają studentów z podstawowymi pojęciami matematyki dyskretnej (zbiory, relacje, funkcje, logika, metody dowodzenia), dają wiedzę dotyczącą tego obszaru matematyki oraz jego wykorzystania w informatyce niezbędnej do układania i analizy algorytmów.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
ma wiedzę z zakresu matematyki dyskretnej niezbędną do formułowania i rozwiązywania typowych zadań z tego zakresu	K_W01
charakteryzuje związki pomiędzy pojęciami funkcji, relacji oraz zbioru; zna i rozumie ideę indukcji matematycznej; potrafi wykorzystać reguły wnioskowania do przeprowadzania dowodów	K_W02
rozpoznaje zastosowania matematyki dyskretnej w informatyce	K_W05
<i>Umiejętności (student potrafi)</i>	
stosuje działania na zbiorach, iloczyn kartezjański, prawa de Morgana i diagramy Venna do rozwiązywania zadań	K_U01
rozwiązuje zadania dotyczące funkcji, iniekcji, suriekcji, funkcji odwrotnej i kompozycji funkcji	K_U05
uzasadnia poprawność wybranych wnioskowań, dowodów i równoważności logicznych	K_U07
<i>Kompetencje społeczne (student jest gotów do)</i>	
jest gotów oceniać poprawność własnych rozwiązań i przyjmować informację zwrotną dotyczącą błędów w rozumowaniu formalnym	K_K01
jest gotów rozwijać umiejętności matematycznego i logicznego rozumowania z wykorzystaniem materiałów cyfrowych	U_K07

Metody i formy kształcenia:

- Wykład problemowy
- Metoda ćwiczeń

Metody oceniania:

- Częstkowe sprawdziany pisemne - kolokwia
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

- Zbiory - symbolika, definicje
- Prawa dot. zbiorów, w tym prawa de Morgana. Diagramy Venn'a

3. Funkcje. Obraz i przeciwobraz zbioru
4. Para uporządkowana i iloczyn kartezjański
5. Relacje. Dziedzina i przeciwdziedzina relacji
6. Relacja zwrotna, przeciwzwrotna, symetryczna i przechodnia
7. Relacja równoważności i klasa równoważności
8. Logika. Funktory zdaniotwórcze. Tablice prawdy
9. Równoważność funkcji zdaniowych
10. Prawa logiczne
11. Funkcja a relacja. Suriekcja, iniekcja, bijekcja, funkcja odwrotna
12. Metody dowodzenia

Ćwiczenia (Ć)

1. Wykazywanie równości zbiorów
2. Wykazywanie równości iloczynów kartezjańskich
3. Dowodzenie własności relacji
4. Badanie równoważności logicznej formuł
5. Wykorzystanie praw rachunku zdań. Tautologie
6. Badania dot. suriekcji, iniekcji, funkcji odwrotnej
7. Wykorzystanie różnych metod dowodzenia

Literatura podstawowa

1. Kenneth Ross, Charles Wright, Matematyka dyskretna, PWN Warszawa (pierwsze rozdziały) 2011

Literatura uzupełniająca

1. Helena Rasiowa, Wstęp do matematyki współczesnej, PWN Warszawa 2013

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Zapoznanie się z literaturą przedmiotu.....	12 godz.....	24 godz.
• Przygotowanie się do kolokwium.....	18 godz.....	27 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest uzyskanie pozytywnej oceny z dwóch kolokwium. Pierwsze jest po połowie zajęć w semestrze, drugie na końcu semestru. Przed kolokwiami studenci otrzymują spis tematów, których będą dotyczyły zadania. Za pojedyncze zadanie student może otrzymać: 0, 0,25, 0,5, 0,75, 1 punktu. Przykładowa skala ocen, gdy można było uzyskać 4 punkty jest przedstawiona poniżej.

0-2	ndst
2,25-2,5	dst
2,75	dst+
3-3,25	db
3,5	db+
3,75-4	bdb

Aktywność studenta na ćwiczeniach jest dodatkowo premiowana.

Fizyka

Nazwa zajęć po angielsku: PHYSICS

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: III

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 2

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia z zakresu nauk podstawowych

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

prof. SWPW dr hab. Władysław Błasiak

Opis zajęć:

Przedmiot „Fizyka” wprowadza studentów Informatyki w wybrane zagadnienia fizyki istotne dla rozumienia modelowania, symulacji, pomiaru oraz działania współczesnych technologii komputerowych. Kurs nie ma charakteru pełnego przeglądu fizyki ogólnej, lecz koncentruje się na tych treściach, które wspierają kompetencje przyszłego informatyka: analizę zjawisk, pracę z danymi, interpretację wyników oraz krytyczną ocenę modeli i narzędzi obliczeniowych.

Szczególne nacisk położony jest na zastosowania fizyki w technice komputerowej: chłodzenie układów elektronicznych, sensory, transmisję sygnałów, światłowody, zakłócenia elektromagnetyczne, układy półprzewodnikowe, elementarne zasady. Istotną częścią kursu są interaktywne symulacje komputerowe, proste eksperymenty wirtualne, analiza parametrów modeli, opracowanie wyników oraz problemy Fermiego.

Studenci uczą się przyjmowania założeń, szacowania wielkości, rozpoznawania ograniczeń symulacji, weryfikowania wyników oraz formułowania wniosków na podstawie danych.

Na zajęciach korzystamy z banku programów w Colorado University, służących do modelowania zjawisk fizycznych w zakresie fizyki klasycznej i kwantowej.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
Student wskazuje wybrane prawa fizyczne przydatne do opisu, wyjaśniania i przewidywania zjawisk oraz działania komputerów.	K_W01
Student charakteryzuje wybrane modele fizyczne i zależności między wielkościami fizycznymi, wskazując ich znaczenie dla symulacji komputerowej i interpretacji danych	K_W02
<i>Umiejętności (student potrafi)</i>	
Student pozyskuje i wykorzystuje źródła elektroniczne, dokumentację symulacji i dane pomiarowe do rozwiązania problemu fizyczno-obliczeniowego	K_U01
Student prezentuje wyniki pomiarów za pomocą doboru odpowiednich programów komputerowych oraz z wykorzystaniem AI	K_U04
<i>Kompetencje społeczne (student jest gotów do)</i>	
Student gotów jest do krytycznej oceny modeli, symulacji, przybliżeń, danych wejściowych i wyników generowanych przez narzędzia komputerowe lub AI	K_K01
Student potrafi korzystać z nowoczesnych technologii w celu kreatywnego rozwiązywania różnych problemów (np. rozwiązywania tzw. problemów Fermiego, tj. takich zadań w których nie ma z góry podanych wszystkich wartości potrzebnych do ich rozwiązania i trzeba je dobrać korzystając z nowoczesnych technologii).	U_K07

Metody i formy kształcenia:

- Wykład problemowy
- Metoda symulacyjna
- Praca ze źródłem elektronicznym
- Metoda ćwiczeń

Metody oceniania:

- Częstkowe sprawdziany testowe - kolokwia
- Praca domowa
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

1. Fizyka w pracy informatyka: modele rzeczywistości, symulacje zjawisk fizycznych, pomiary, błędy i niepewności pomiarowe.
2. Mechanika i ruch jako podstawa modelowania zjawisk dynamicznych. Równnia ruchu. Symulacje ruchu ciał w polu grawitacyjnym.
3. Energia, moc. Sprawność w systemach technicznych i komputerowych.
4. Drgania, rezonans i stabilność układów. Oscylator harmoniczny i anharmoniczny.
5. Termodynamika i przepływ ciepła w technice komputerowej. Temperatura, ciepło, przewodzenie, konwekcja, przegrzewanie układów, chłodzenie procesorów.
6. Elektromagnetyzm: prąd, pole elektryczne i magnetyczne, sygnał i jego zakłócenia. Równania Maxwella.
7. Fale, optyka i transmisja informacji, interferencja, dyfrakcja, światłowody.
8. Fizyczne podstawy elektroniki i półprzewodników. Przewodnictwo, dioda, tranzystor, układ scalony; wprowadzenie do fizyki sprzętu komputerowego.
9. Modelowanie i symulacje z wykorzystaniem metody Monte Carlo: parametry i założenia modelu, jego walidacja oraz ograniczenia symulacji. Rozwiązywanie problemów Fermiego.
10. Elementy fizyki kwantowej dla informatyka. Superpozycja, pomiar, kubit, bramka kwantowa, wynik probabilistyczny, wprowadzenie do idei komputerów kwantowych.

Ćwiczenia (Ć)

1. Symulacja ruchów ciał pod wpływem różnych sił. Rozwiązywanie równań ruchu. Symulacja ruchu planet w Układzie Słonecznym.
2. Drgania i rezonans w symulacji komputerowej. Badanie wpływu masy, sprężystości, tłumienia i wymuszenia na ruch. Wykresy fazowe oscylatora harmonicznego i anharmonicznego.
3. Rozwiązywanie problemów Fermiego. Formułowanie założeń, szacowanie, obliczenia przybliżone.
4. Metoda Monte Carlo. Wyznaczanie liczby π metodą Buffona.
5. Symulacja modelu gazu doskonałego. Temperatura, energia cząsteczek, ciśnienie, rozkłady, interpretacja wyników symulacji.
6. Symulacja rozkładu pola elektrycznego i magnetycznego: linie pola, potencjał.
7. Symulacja zjawisk optycznych. Odbicie, załamanie światła, całkowite wewnętrzne odbicie. Światłowody. Symulacja biegu promieni świetlnych w ośrodkach anizotropowych.
8. Opracowanie przykładowych wyników eksperymentalnych. Tabele wyników, wykresy, niepewności pomiarowe, korelacje, wnioski.

Literatura podstawowa

1. R. Resnick, D. Halliday, J. Walker. Podstawy Fizyki t. 1 - 5, PWN, Warszawa 2012

Literatura uzupełniająca

1. W. Błasiak, seria opowiadań zatytułowanych: "Słońce i My" /tekst dostępny na platformie WLODEK/.
2. W. Błasiak - Wszechświat i Mózg (wykład inauguracyjny) / tekst dostępny na platformie WLODEK

Źródła dodatkowe

1. SYMULACJE KOMPUTEROWE - COLORADO UNIVERSITY:
<https://phet.colorado.edu/en/simulations/filter?subjects=earth-science,physics&sort=alpha>
2. WYKŁADY z FIZYKI: https://www.youtube.com/playlist?list=PLRc6a9k_z6Mm96Elf-buW8P8OXNJFkwHQ
3. W. Błasiak, Wyzwalanie kreatywności na lekcjach fizyki, Hejnał Oświatowy, nr 12/228, 2023, ISSN 1233 -7609, p. 3 – 9 / materiał dostępny na platformie WLODEK

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego..... 30 godz..... 16 godz.
- Przygotowanie się do zajęć..... 5 godz..... 10 godz.
- Samodzielne wykonanie zadań..... 7 godz..... 10 godz.

- Wypełnienie dokumentacji zajęć..... 4 godz..... 6 godz.
- Zapoznanie się z literaturą przedmiotu..... 4 godz..... 8 godz.
- Suma obciążeń:..... 50 godz..... 50 godz.

Zaliczenie zajęć:

Ocena końcowa będzie wystawiona na podstawie wyników dwóch sprawdzianów pisemnych oraz pracy domowej złożonej na Platformie WLODEK. Sprawdziany pisemne będą odbywać się w trakcie zajęć w czasie ok. 15 - 20 minut. Będą dotyczyły zagadnień omawianych na wykładzie. Poniżej podane są przykłady takich zadań.

Sprawdziany i praca domowa będą oceniane w skali od 2 do 5 (wg. kryteriów podanych przy zadaniach). Końcowa ocena będzie średnią arytmetyczną ocen ze sprawdzianów oraz pracy domowej.

Za szczególną aktywność na zajęciach będzie można uzyskać podwyższenie oceny końcowej o 0,5 pkt. (dla ocen końcowych niższych od 4,5 pkt). Podwyższenie oceny za aktywność będzie uzasadnione przez prowadzącego zajęcia, w obecności wszystkich słuchaczy. Każdy taki przypadek zostanie odnotowany na platformie WLODEK.

Przykłady szczególnej aktywności: wykonanie i zaprezentowanie symulacji zjawiska fizycznego (np. efektu Magnusa) lub sformułowanie i rozwiązanie oryginalnego zadania typu Fermiego.

Niżej podano przykładową pracę domową oraz sprawdzian z poprzednich kursów.

Praca domowa.

Opisz 3 wybrane programy symulacyjne z banku programów Colorado University:

Adres strony:

<https://phet.colorado.edu/en/simulations/filter?subjects=earth-science,physics&sort=alpha&view=gri>

Dla każdego programu proszę:

1. Opisać istotę symulowanego zjawiska fizycznego.
2. Zaproponować wirtualny eksperyment (eksperymenty) możliwy do realizacji za pomocą danej symulacji
3. Wskazać zalety oraz ew. wady symulacji komputerowej i zaproponować jej udoskonalenie.

Kryteria oceny.

Za opisanie istoty zjawiska fizycznego – 9 pkt (po 3 pkt za każde zadanie)

Za propozycję wirtualnego eksperymentu – 9 pkt (po 3 pkt za każde zadanie)

Za wskazanie zalet i ew. wad symulacji i propozycję jej udoskonalenia – 9 pkt – (po 3 pkt za każde zadanie)

Razem 27 pkt.

0 – 12 pkt - ocena ndst (2)

13 – 17 pkt - ocena dst (3)

18 – 22 pkt – ocena db (4)

23 – 27 pkt – ocen bdb (5)

Sprawdzian.

Oszacuj ile energii mógłby pozyskać od Słońca Twój wykładowca nosząc fotowoltaiczny kombinezon przez jeden słoneczny dzień.

Kryteria oceny.

Poprawność merytoryczna idei rozwiązania - 3

Uzasadnienie dokonanych przybliżeń - 3

Poprawność merytoryczna i językowa komentarzy - 3

Razem 9 pkt.

0 – 4 pkt - ocena ndst

5 – 6 pkt - ocena dst.

7 – 8 pkt – ocena db

8 – 9 pkt – ocen bdb

Podstawy elektrotechniki i elektroniki

Nazwa zajęć po angielsku: Basics of electrical engineering and electronics

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: IV

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia z zakresu nauk podstawowych

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Laboratorium.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

prof. SWPW dr hab. Władysław Błasiak, mgr inż. Krzysztof Obst

Opis zajęć:

Przedmiot skierowany jest do studentów kierunku Informatyka na studiach inżynierskich. Na zajęciach wykładowych studenci poznają podstawowe elementy i układy elektroniczne oraz narzędzia sprzętowe i programowe niezbędne do wykonywania pomiarów typowych wielkości elektrycznych. W ramach zajęć laboratoryjnych studenci, podzieleni na małe grupy, realizują praktyczne zadania na stanowiskach ćwiczeniowych, zgodnie z instrukcją.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
zna i rozumie wybrane treści z zakresu elektrotechniki i elektroniki	K_W01
opisuje budowę i funkcjonowanie typowych elementów i układów elektronicznych	K_W02
charakteryzuje praktyczne zastosowania elektrotechniki i elektroniki w informatyce	K_W03
wymienia i wyjaśnia wybrane standardy oraz normy techniczne występujące w elektrotechnice i elektronice w obrębie zastosowań w informatyce	K_W06
Umiejętności (student potrafi)	
wyszukiuje w literaturze i źródłach elektronicznych informacji na temat budowy i podstawowych parametrów typowych elementów i układów elektronicznych	K_U01
planuje i przeprowadza pomiary z dziedziny elektrotechniki, interpretuje uzyskane wyniki i formułuje wnioski	K_U06
stosuje metody analityczne i eksperymentalne do rozwiązywania zadań inżynierskich z zakresu elektrotechniki i elektroniki	K_U07
dokonuje wstępnej analizy ekonomicznej doboru elementów, układów lub rozwiązań technicznych w zastosowaniu informatycznym	K_U09
współpracuje w zespole podczas realizacji ćwiczeń laboratoryjnych, dzieląc zadania pomiarowe, dokumentacyjne i analityczne	K_U18
Kompetencje społeczne (student jest gotów do)	
jest gotów rozwijać kompetencje techniczne z wykorzystaniem materiałów cyfrowych, kart katalogowych, instrukcji i nowoczesnych technologii elektronicznych	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Dyskusja

Metody oceniania:

- Przygotowanie do zajęć - wejściówki
- Sprawozdanie z realizacji zadania

- Praca ze źródłem elektronicznym
- Metoda laboratoryjna
- Zaliczenie testowe

Wykład konwersatoryjny (W)

1. Podstawowe pojęcia: ładunek elektryczny, obwód elektryczny, podstawowe prawa obwodu prądu stałego, moc i praca prądu elektrycznego.
2. Metody rozwiązywania obwodów elektrycznych: metoda superpozycji.
3. Źródła energii elektrycznej, przebiegi łączeniowe w obwodach RC i LC. Zjawisko indukcji elektromagnetycznej.
4. Klasyfikacja prądów zmiennych. Metoda liczb zespolonych. Wartość średnia i skuteczna. Elementy RLC w obwodach prądu zmiennego.
5. Diody półprzewodnikowe: charakterystyki i zastosowanie: dioda prostownicza, Zenera, LED, fotodioda
6. Tranzystory bipolarne i unipolarne: zasada funkcjonowania, charakterystyki statyczne, zastosowanie.
7. Podstawy techniki cyfrowej: podstawowe funkcje logiczne, realizacja układów logicznych. Układy TTL, CMOS, parametry techniczne.

Laboratorium (L)

1. Konfiguracja środowiska programistycznego
2. Programowanie wejść i wyjść cyfrowych
3. Obsługa sygnałów analogowych
4. Sterowanie elementami wykonawczymi
5. Współpraca z czujnikami
6. Obsługa wyświetlaczy i elementów sygnalizacyjnych
7. Komunikacja szeregową i magistralę sprzętową
8. Projekt systemu Internetu Rzeczy (IoT)

Literatura podstawowa

1. B. Bolanowski, Podstawy elektrotechniki: konspekt wykładu dla kierunku informatyka. Łódź WSEH 2002
2. Instrukcje do ćwiczeń dostępne na platformie Wlodek

Literatura uzupełniająca

1. P. Fabijański, Podstawy elektroniki. REA Warszawa 2006

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Zapoznanie się z literaturą przedmiotu.....	15 godz.....	15 godz.
• Przygotowanie się do zajęć.....	5 godz.....	15 godz.
• Wypełnienie dokumentacji zajęć.....	5 godz.....	15 godz.
• Przygotowanie się do zaliczenia końcowego.....	5 godz.....	6 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest uzyskanie zakładanych efektów uczenia się. Weryfikacja wiedzy teoretycznej polega na rozwiązaniu testu końcowego oraz udzieleniu odpowiedzi na pytania zadane przed rozpoczęciem ćwiczeń laboratoryjnych. Weryfikacja umiejętności polega na odpowiedzi na pytanie przed rozpoczęciem zajęć, wykonaniu ćwiczeń zgodnie z podaną instrukcją oraz przygotowanie sprawozdania z wykonania zadania. Warunkiem zaliczenia zajęć jest uzyskanie ponad 50% punktów ze wszystkich form sprawdzania wiedzy i umiejętności.

Wprowadzenie do informatyki

Nazwa zajęć po angielsku: Introduction to computer science

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: I

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 2

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny 15 / 8 godz 0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

prof. SWPW dr hab. Władysław Błasiak

Opis zajęć:

Wykład przedstawia historię technik obliczeniowych, poczynając od najstarszych maszyn liczących, poprzez kolejne generacje komputerów. Ekspozuje związki informatyki z matematyką, fizyką i rozwojem technologii (lampy, tranzystory, elementy scalone).

Omawia fizyczne granice stosowalności prawa Moora, rozwój oprogramowania użytkowego, rodzaje algorytmów, idea uczenia maszynowego oraz komputerów kwantowych.

Kurs kończy się rozważaniami na temat perspektyw sztucznej inteligencji oraz pozytywnego (medycyna, edukacja, transport, badania naukowe) i negatywnego wpływu technologii informacyjnych na życie społeczeństwa (cyberuzależnienia).

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wie na czym polega modelowanie procesów (w tym zjawisk przyrodniczych) za pomocą języka matematyki	K_W01
podaje argumenty wyjaśniające ograniczony zasięg prawa Moora	K_W02
przedstawia korzyści wynikające z korzystania z technologii informacyjnych (w tym AI) oraz związane z tym zagrożenia	K_W07
Umiejętności (student potrafi)	
analizuje i uzasadnia potrzebę prowadzenia bardzo szybkich obliczeń, np. oszacować czas w problemie komiwojażera dla n miast lub napisać ogólny algorytm poruszania się n cząstek gazu doskonałego	K_U01
opisuje wybrane zjawiska językiem matematyki oraz zaproponować sposób prezentacji wyników obliczeń i przebiegu zjawisk przy użyciu komputera	K_U02
ocenia użyteczność i przydatność ogólnie dostępnych aplikacji komputerowych	K_U04
projektuje symulację komputerową złożonego procesu lub zjawiska	K_U05
projektuje bramki logiczne	K_U11
Kompetencje społeczne (student jest gotów do)	
jest gotów oceniać wiarygodność informacji o nowych technologiach i ograniczenia własnej wiedzy o informatyce	K_K01
jest gotów do rozwiązywania problemów, np. jest gotów przeprowadzić test Turinga dla wybranej wersji AI	K_K02
jest gotów rozwijać wiedzę o nowych technologiach informacyjnych, np. rozumie konieczność	U_K07

Metody i formy kształcenia:

- Wykład problemowy
- Wykład informacyjny
- Dyskusja
- Burza mózgów
- Praca ze źródłem elektronicznym

Metody oceniania:

- Częstkowe sprawdziany pisemne - kolokwia

Wykład konwersatoryjny (W)

1. Przykłady ilustrujące konieczność stosowania szybkich metod obliczeniowych. Problem komiwojażera. Symulacje złożonych zjawisk przyrodniczych. Deterministyczny chaos. Pomiar medyczny i biometryczny.
2. Informatyka i jej związki z matematyką, fizyką oraz techniką. Dane i sposoby ich kodowania (system binarny, szesnastkowy i inne). Lampy elektronowe (dioda, trioda), tranzystory, elementy scalone, bramki logiczne, (konstrukcje bramek logicznych: NOT, AND, NAND, OR, NOR, XOR).
3. Krótka historia technologii informacyjnych. Najstarsze maszyny liczące i pierwsze programy (abakus, pałeczki Napiera, pascalina, pierwszy mechaniczny komputer Babbage'a, program Ady Lovelace). Pierwsza generacja komputerów - lampy elektronowe. Druga generacja - tranzystory. Trzecia generacja - układy scalone. Czwarta generacja - mikroprocesory i mikrokomputery.
4. Internet. Powstanie systemu stron WWW i jego konsekwencje. Podstawy programowania stron WWW. Internetowe banki danych. Wprowadzenie do algorytmów. Nauczanie zdalne.
5. Granice miniaturyzacji. Prawo Moora. Fizyczne granice stosowalności prawa Moora. I co dalej?
6. Komputery kwantowe. Krótki, jakościowy opis praw mechaniki kwantowej. Zasada działania komputera kwantowego. Kluczowe cechy komputerów kwantowych. Kubity. Splątanie kwantowe. Idea kryptografii kwantowej. Przegląd aktualnych postępów w zakresie budowy komputerów kwantowych.
7. Sztuczna inteligencja. Test Turinga (przykłady). Ocena aktualnych możliwości AI. Wizja przyszłości. Testowanie aktualnych granic możliwości chat.gpt.
8. Wybrane przykłady społecznych skutków rozwoju informatyki (wybory, giełda, bioinżynieria).

Literatura podstawowa

1. Lembas J., Kawa R., Wstęp do Informatyki, Wydawnictwo Naukowe PWN, 2017

Literatura uzupełniająca

1. Brookshear J. G., Brylow D., Informatyka w ogólnym zarysie, Wydawnictwo Naukowe PWN 2022
2. Mieścicki M., Wstęp do informatyki, Wydawnictwo BTC, 2013

Źródła dodatkowe

1. Cieciora M., Wybrane problemy społeczne i zawodowe informatyki, Otwarte Zasoby Edukacyjne - OZE, Warszawa 2021
2. Stefanowicz B. (red.), Wstęp do informatyki, SGH 1999
3. Strona internetowa profesora Ryszarda Tadeusiewicza: <https://www.cri.agh.edu.pl/uczelnia/tad/>
4. Strona internetowa Interactive Simulations for Science and Math: <https://phet.colorado.edu/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- | | | |
|---|--------------|----------|
| • Zajęcia z udziałem nauczyciela akademickiego..... | 15 godz..... | 8 godz. |
| • Zapoznanie się z literaturą przedmiotu..... | 17 godz..... | 20 godz. |
| • Przygotowanie się do zajęć..... | 8 godz..... | 12 godz. |
| • Przygotowanie się do kolokwium..... | 5 godz..... | 5 godz. |
| • Samodzielne wykonanie zadań..... | 5 godz..... | 5 godz. |
| Suma obciążeń:..... | 50 godz..... | 50 godz. |

Zaliczenie zajęć:

Warunkiem koniecznym uzyskania zaliczenia będzie uzyskanie pozytywnej oceny z co najmniej połowy częściowych sprawdzianów pisemnych. Sprawdziany te będą się odbywać na zajęciach (z wyjątkiem pierwszych ćwiczeń). Będą trwać ok. 10 - 15 minut. Zadanie/zadania będą związane z tematyką poprzedniego wykładu. Ocena końcowa zaliczenia będzie średnią arytmetyczną ocen ze wszystkich sprawdzianów.

Dla chętnych studentów będzie możliwość podwyższenia oceny końcowej o 1 stopień za przygotowanie i pozytywną ocenę przygotowanej symulacji komputerowej na temat:

- a) Wyznaczanie liczby π metoda Buffona.
- b) Symulacja przebiegu wybranego zjawiska fizycznego.
- c) Modelowanie rozkładu Gaussa.

Zgłoszone prace będą prezentowane i oceniane przez wykładowcę na ostatnich ćwiczeniach kursu.

Architektura systemów komputerowych

Nazwa zajęć po angielsku: Architecture of computer systems

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: I

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny..... 30 / 16 godz..... 0 / 0 godz.
- Ćwiczenia..... 15 / 8 godz..... 0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Maciej Krawczyk

Opis zajęć:

W ramach przedmiotu student uzyskuje wiedzę z zakresu budowy i przeznaczenia podstawowych bloków funkcjonalnych komputera, zasad ich współpracy, problemów zarządzania zasobami pamięciowymi, współpracy z urządzeniami zewnętrznymi itp. Ponadto uzyskuje umiejętność oceny wydajności różnych architektur komputerów, śledzenia lub napisania fragmentów programów assemblerowych. Zajęcia umożliwiają dalsze samodzielne pogłębianie wiedzy i śledzenie rozwoju technik stosowanych w komputerach i złożonych systemach informatycznych.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia podstawowe zagadnienia architektury systemów komputerowych oraz opisuje procesy zachodzące w procesorze, pamięci, magistralach i urządzeniach wejścia-wyjścia	K_W04
charakteryzuje model programowy współczesnych procesorów oraz rozpoznaje metody, techniki i narzędzia stosowane przy analizie architektury komputerów	K_W05
wymienia i wyjaśnia wybrane standardy i normy techniczne występujące w budowie oraz eksploatacji systemów komputerowych	K_W06
<i>Umiejętności (student potrafi)</i>	
wykorzystuje wiedzę z zakresu architektury systemów komputerowych do rozwiązywania problemów praktycznych związanych z konfiguracją, analizą wydajności i pracą niskopoziomową	K_U05
planuje i przeprowadza pomiary wydajności procesora oraz pamięci, interpretuje wyniki i formułuje wnioski	K_U06
ocenia sposób funkcjonowania wybranych rozwiązań sprzętowych, w szczególności procesora, pamięci podręcznej, pamięci operacyjnej i urządzeń komputerowych	K_U10
potrafi zaprojektować i skonstruować system komputerowy bazujący na powszechnie dostępnych komponentach elektronicznych	K_U11
stosuje właściwe technologie, standardy i normy inżynierskie przy rozwiązywaniu zadań dotyczących budowy i działania systemów komputerowych	K_U12
posługuje się specjalistyczną terminologią z zakresu architektury komputerów w komunikacji technicznej i dokumentowaniu wykonanych zadań	K_U14
<i>Kompetencje społeczne (student jest gotów do)</i>	

jest gotów krytycznie oceniać własną wiedzę i informacje dotyczące budowy oraz działania systemów komputerowych	K_K01
jest gotów rozwiązywać problemy poznawcze i praktyczne oraz zasięgać opinii ekspertów w przypadku trudności z samodzielnym rozwiązywaniem zadań z architektury komputerów	K_K02
jest gotów rozwijać wiedzę z zakresu architektury systemów komputerowych z wykorzystaniem nowoczesnych technologii, dokumentacji technicznej i źródeł elektronicznych	U_K07

Metody i formy kształcenia:

- Praca ze źródłem elektronicznym
- Metoda pokazu
- Metoda ćwiczeń
- Wykład problemowy

Metody oceniania:

- Zaliczenie testowe
- Praca domowa

Wykład konwersatoryjny (W)

1. Wprowadzenie w problematykę architektury komputerów. Struktura i działanie procesora (koncepcja von Neumana i Harvard).
2. Systemy liczbowe. Reprezentacja danych w systemach komputerowych.
3. Model programowy procesora. Koncepcje CISC i RISC.
4. Budowa jednostki wykonawczej, jednocyklowa, wielocyklowa, potokowa.
5. Potokowe jednostki wykonawcze w architekturach CISC i RISC.
6. Organizacja pamięci podręcznej (kieszeniowej).
7. Ewolucja struktury komputera od pamięciocentrycznej do wieloszynowej.

Ćwiczenia (Ć)

1. Budowa współczesnych komputerów. Zasady BHP obowiązujące podczas montażu i demontażu podzespołów komputerowych.
2. Struktura programu komputerowego (prefiks, kod, dane, stos, sarta, zasoby) w powiązaniu z architekturą x86/x64.
3. Narzędzia i metody pomiaru wydajności procesora oraz pamięci.
4. Wprowadzenie do programowania niskopoziomowego. Język Assembler.

Literatura podstawowa

1. W. Stallings, Organizacja i architektura systemu komputerowego Tom 1 - Wydanie XI, WNT

Literatura uzupełniająca

1. P. Metzger, Anatomia PC, Helion
2. A. Skorupski, Podstawy budowy i działania komputerów, WKŁ
3. D. A. Patterson, J. L. Hennessy, Computer Organization and Design
4. A. S. Tanenbaum, T. Austin, Structured Computer Organization
5. S. Kruk, Asembler Wykłady i ćwiczenia, Mikom, wybrane fragmenty

Źródła dodatkowe

1. <https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html#combined>
2. <http://www.cs.cmu.edu/~ralf/files.html>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....45 godz.....24 godz.
- Zapoznanie się z literaturą przedmiotu.....10 godz.....13 godz.
- Przygotowanie się do zajęć.....4 godz.....8 godz.
- Przygotowanie się do zaliczenia końcowego.....6 godz.....12 godz.
- Samodzielne wykonanie zadań.....10 godz.....18 godz.
- Suma obciążeń:.....75 godz.....75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia wykładu jest rozwiązywanie testów sprawdzających na platformie "Wlodek". Testy są udostępniane po każdym wykładzie. Testy są dostępne przez okres 7 dni. Odpowiedzi są oceniane w skali punktowej zależnej od wagi pytania. Obowiązuje następująca skala ocen dotycząca procentowej sumy punktów ze wszystkich testów:

51-60% - ocena dostateczna

61-70% - ocena dostateczna plus

71-80% - ocena dobra

81-90% - ocena dobra plus

91-100% - ocena bardzo dobra.

Warunkiem zaliczenia ćwiczeń jest samodzielne wykonanie pracy domowej zadawanej po każdym ćwiczeniu.

Sprawozdanie z pracy domowej musi być umieszczone na platformie Wlodek w ciągu 14 dni od ćwiczeń.

Sprawozdania są oceniane w skali punktowej od 0 do 5. Punkty są przeliczane na ocenę w sposób analogiczny, jak w przypadku wykładu.

Warunkiem zaliczenia przedmiotu jest uzyskanie oceny pozytywnej z wykładu i z ćwiczeń. Na ocenę końcową składają się w równym stopniu ocena z wykładu i ocena z ćwiczeń.

Systemy operacyjne

Nazwa zajęć po angielsku: Operating systems

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: II

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Wojciech Korzybski

Opis zajęć:

Zajęcia prezentują podstawową rolę i zadania systemu operacyjnego oraz sposoby zarządzania zasobami sprzętowymi komputera: procesorem, pamięcią operacyjną i urządzeniami wejścia-wyjścia. Szczegółowo została przedstawiona także koncepcja procesu oraz pliku i praktyczna realizacja systemu plików.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia rolę systemu operacyjnego oraz podstawowe mechanizmy zarządzania procesami, pamięcią, plikami, urządzeniami, użytkownikami i usługami systemowymi	K_W04
charakteryzuje podstawowe narzędzia i techniki pracy z systemami Windows i Linux, w tym powłoki, skrypty, system plików, uprawnienia, procesy i usługi	K_W05
rozpoznaje standardy i mechanizmy techniczne występujące w systemach operacyjnych, w tym struktury systemów plików, uprawnienia, konta użytkowników, logowanie zdarzeń i aktualizacje	K_W06
Umiejętności (student potrafi)	
dobiera właściwe narzędzia systemowe do realizacji zadań administracyjnych i diagnostycznych	K_U04
wykorzystuje wiedzę o systemach operacyjnych do wykonywania praktycznych zadań dotyczących plików, procesów, użytkowników, usług i konfiguracji środowiska	K_U05
analizuje działanie wybranych mechanizmów systemu operacyjnego oraz ocenia wpływ konfiguracji na bezpieczeństwo, wydajność i wygodę użytkownika	K_U10
rozwiązuje praktyczne zadania inżynierskie z użyciem poleceń systemowych, skryptów, dokumentacji technicznej i standardowych narzędzi systemowych	K_U12
organizuje własną pracę z systemem operacyjnym, dokumentuje wykonane czynności i przygotowuje skrypty ułatwiające powtarzalne zadania	K_U17
Kompetencje społeczne (student jest gotów do)	
jest gotów zasięgać opinii ekspertów i korzystać z dokumentacji w przypadku trudności z rozwiązaniem problemów systemowych	K_K02

Metody i formy kształcenia:

- Wykład informacyjny

Metody oceniania:

- Zaliczenie testowe

- Praca ze źródłem elektronicznym
- Metoda ćwiczeń
- Metoda pokazu
- Dyskusja
- Aktywność w trakcie zajęć
- Zaliczenie praktyczne

Wykład konwersatoryjny (W)

1. Informacje wstępne o systemach operacyjnych
2. Rola i zadania systemów operacyjnych w zakresie uprawnień i usług
3. Podział systemów operacyjnych
4. Koncepcja procesu i zasobu w systemach operacyjnych
5. Przerwania w systemach operacyjnych
6. Model architektury systemu operacyjnego
7. System plików i implementacje systemu plików
8. Podstawy bezpieczeństwa systemu operacyjnego

Ćwiczenia (Ć)

1. Przykładowe polecenia wiersza poleceń
2. Polecenia systemu operacyjnego i ich parametry
3. Programy wsadowe z wykorzystaniem poleceń systemu Windows
4. Programy wsadowe z parametrami
5. PowerShell - charakterystyka środowiska
6. Polecenia i aliasy w PowerShell
7. Zadania praktyczne do pracy na plikach i folderach

Literatura podstawowa

1. A. Silberschatz, G. Gagne, P. B. Galvin, Podstawy systemów operacyjnych Tom I, Wydawnictwo Naukowe PWN 2021
2. A. Silberschatz, G. Gagne, P. B. Galvin, Podstawy systemów operacyjnych Tom II, Wydawnictwo Naukowe PWN 2021

Literatura uzupełniająca

1. A. Silberschatz, J.L. Peterson, G. Gagne, Podstawy systemów operacyjnych. WNT, Warszawa 2005.
2. A. S. Tanenbaum, Systemy operacyjne, Wydanie III, Helion 2010
3. A. Bertram, PowerShell dla administratorów systemów. Prosta automatyzacja zadań, Helion 2018

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....60 godz.....32 godz.
- Zapoznanie się z literaturą przedmiotu.....15 godz.....25 godz.
- Przygotowanie się do zaliczenia końcowego.....15 godz.....28 godz.
- Samodzielne wykonanie zadań.....10 godz.....15 godz.
- Suma obciążeń:.....100 godz.....100 godz.

Zaliczenie zajęć:

Zaliczenie przedmiotu wymaga zdania testu komputerowego (min. 50%) z zakresu treści wykładowych oraz zaliczenia zadań ćwiczeniowych. Wymagane jest także przesyłanie zadań na platformę Wlodek, wykonywanych podczas zajęć ćwiczeniowych. Ocena końcowa jest średnią arytmetyczną z zaliczenia testu oraz oceny z ćwiczeń.

System operacyjny LINUX

Nazwa zajęć po angielsku: Linux System

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: III

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: kurs e-learningowy

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot "System Linux" przeznaczony jest dla studentów kierunku Informatyka, którzy znają już architekturę systemów komputerowych, rozumieją działanie systemów operacyjnych oraz potrafią programować. Celem przedmiotu jest wykształcenie umiejętności posługiwania się systemem operacyjnym LINUX, w tym administrowanie poszczególnymi obszarami systemu oraz programowanie w powłoce systemu (shellu). Cały kurs nastawiony jest na informatyczną obsługę systemu z konsoli znakowej, w minimalnym stopniu omawiając środowisko graficzne. Dzięki temu studenci nauczą się korzystać z dowolnego systemu Unix / Linux, niezależnie od producenta oraz dystrybucji. Zakres materiału odpowiada podstawowemu profilowi certyfikacyjnemu LPIC-1, opracowanemu przez Linux Professional Institute, który potwierdza umiejętność wykonywania zadań administracyjnych w wierszu poleceń, instalacji i konfiguracji komputera z Linuxem oraz podstawowej konfiguracji sieci. Umożliwi to studentom administrowanie systemem, dbanie o jego bezpieczeństwo oraz opracowywanie skryptów ułatwiających codzienną pracę informatyka.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
charakteryzuje budowę, zastosowania i rolę systemu Linux w środowiskach informatycznych	K_W03
wyjaśnia procesy działania systemu Linux: uruchamianie, zarządzanie procesami, usługami, plikami, uprawnieniami, użytkownikami, pakietami i logami	K_W04
identyfikuje polecenia, narzędzia i techniki stosowane w administracji Linuxem, pracy z powłoką, diagnozowaniu problemów i automatyzacji zadań	K_W05
rozpoznaje standardy, dobre praktyki i zasady bezpieczeństwa dotyczące systemu plików, uprawnień, kont użytkowników, usług, aktualizacji i licencji open-source	K_W06
wyjaśnia ekonomiczne, prawne i etyczne uwarunkowania wykorzystania systemu Linux, wolnego oprogramowania i licencji open-source w organizacji	K_W08
rozróżnia licencje i zasady legalnego wykorzystania oprogramowania instalowanego w systemie Linux	K_W09
Umiejętności (student potrafi)	
pozyskuje i wykorzystuje dokumentację (man, info), dokumentację dystrybucji i źródła techniczne do rozwiązania problemu administracyjnego	K_U01
dobiera polecenia, narzędzia i opcje systemowe do realizacji zadań w systemie Linux	K_U04
ocenia działanie systemu Linux, diagnozuje problemy z procesami, usługami, plikami, uprawnieniami i kontami oraz porównuje rozwiązania z innymi systemami	K_U10

konfiguruje użytkowników, grupy, uprawnienia, usługi, archiwizację, zdalny dostęp SSH i wybrane mechanizmy bezpieczeństwa systemu Linux	K_U11
komunikuje się z użytkownikami i administratorami systemu Linux, korzysta z SSH oraz dokumentuje wykonane czynności administracyjne z użyciem terminologii technicznej	K_U14
opracowuje skrypty powłoki automatyzujące powtarzalne czynności administracyjne i organizuje własną pracę administratora	K_U17
Kompetencje społeczne (student jest gotów do)	
jest gotów do zasięgania opinii ekspertów i korzystania z dokumentacji w przypadku trudności z samodzielnym rozwiązaniem problemu systemowego	K_K02
jest gotów do przestrzegania zasad etyki administratora, poufności danych, odpowiedzialnego korzystania z uprawnień i legalnego używania oprogramowania open-source	K_K05
jest gotów do samorozwoju w zakresie nowych dystrybucji, narzędzi administracyjnych, bezpieczeństwa i automatyzacji systemów Linux	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Wykład problemowy
- Praca ze źródłem elektronicznym
- Metoda sytuacyjna
- Metoda ćwiczeń

Metody oceniania:

- Częstkowe sprawdziany testowe - kolokwia
- Zaliczenie praktyczne
- Ocena aktywności w kursie

Wykład konwersatoryjny (W)

1. Architektura systemów Unix/Linux (kernel, przestrzeń użytkownika, powłoka, procesy, usługi, dystrybucje)
2. Linux w ekosystemie informatyki (open source, serwery, chmura, kontenery, bezpieczeństwo, DevOps)
3. System plików w linuxie (budowa systemu, pamięci zewnętrzne, pliki i katalogi, typu plików)
4. Praca z plikami i katalogami (podstawowe operacje, wyświetlanie i edycja plików tekstowych, sortowanie, wycinanie, filtry i potoki)
5. Bieżąca praca w systemie (użytkownicy i grupy, prawa dostępu, archiwizacja i kompresja danych, komunikacja w systemie)
6. Systemowe środowiska pracy (zmienne, konfiguracja systemu, procesy i wielozadaniowość, automatyzacja zadań, sieć komputerowa, instalacja oprogramowania)
7. Skrypty powłoki (interakcja i obliczenia, konstrukcja skryptu, warunki, pętle, parametry, obsługa błędów)

Ćwiczenia (Ć)

1. Wykorzystanie serwera ćwiczeniowego suse.wlodkowic.pl. Konfiguracja programu komunikacyjnego.
2. Logowanie się do systemu. Zapoznanie się ze środowiskiem pracy.
3. Wydawanie poleceń. Analiza komunikatów o błędach. Pomoc systemowa.
4. Wyświetlanie podstawowych informacji o systemie i użytkownikach.
5. Praca z pamięciami zewnętrznymi. Praca z plikami i katalogami.
6. Przeglądanie zawartości plików tekstowych oraz edycja plików tekstowych.
7. Operacje na plikach tekstowych: sortowanie, wyszukiwanie, wycinanie, łączenie.
8. Wykorzystanie mechanizmów strumieni i potoków.
9. Zarządzanie użytkownikami i grupami. Zasady bezpieczeństwa w systemie: ustalanie i zmiana praw dostępu do plików i katalogów.
10. Archiwizacja i kompresja danych.
11. Tworzenie i wykorzystanie dowiązań do plików (linków).
12. Wykorzystanie zmiennych w systemie Linux. Konfiguracja systemu Linux.
13. Nadzór nad procesami, usługami i logami. Wykorzystanie sieci komputerowej.
14. Instalacja i aktualizacja oprogramowania (pakiety) w konsoli oraz w środowisku graficznym.

15. Interakcja pomiędzy użytkownikiem i systemem.
16. Budowa skryptów wykonywanych w powłoce. Parametry, warunki, pętle, walidacja danych.
17. Skrypty administracyjne.

Literatura podstawowa

1. Linux Essentials wersja 1.6 Polski. <https://learning.lpi.org/pdfstore/LPI-Learning-Material-010-160-pl.pdf>
2. LPIC-1 Exam 101, <https://learning.lpi.org/pdfstore/LPI-Learning-Material-101-500-en.pdf>
3. R. Żak, System operacyjny Linux, Novum 2014
4. kurs e-learningowy na platformie WLODEK

Literatura uzupełniająca

1. LPIC-1 Exam 102, <https://learning.lpi.org/pdfstore/LPI-Learning-Material-102-500-en.pdf>
2. E. Nemeth, G. Snyder, T.R. Hein, B. Whaley, D. Mackin: Unix i Linux. Przewodnik administratora systemów. Wydanie V, Helion 2018
3. M. Sobell, Praktyczny przewodnik Linux programowanie w powłoce. Helion 2014
4. C. Flynt, S. Lakshman, S. Tushar, Skrypty powłoki systemu Linux. Receptury. Helion 2012
5. M. McCallister, SUSE Linux 10. Księga eksperta, Helion 2006

Źródła dodatkowe

1. Linux Professional Institute, <https://www.lpi.org/>
2. <http://www.linux.pl/>
3. <https://www.suse.com/>
4. A. S. Tanenbaum, Systemy operacyjne, Wydanie III, Helion 2010

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Zapoznanie się z literaturą przedmiotu.....	5 godz.....	10 godz.
• Realizacja kursu e-learningowego.....	20 godz.....	30 godz.
• Przygotowanie się do kolokwium.....	5 godz.....	11 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest, co najmniej w stopniu podstawowym, przyswojenie wiadomości na temat systemu Linux, zdobycie umiejętności pracy z systemem i zarządzania nim oraz nabycie kompetencji polegających na samodzielnym rozwiązywaniu problemów zawodowych.

Na każdych zajęciach praktycznych (ćwiczeniowych) studenci przystępują do praktycznego testu (treść na platformie, realizacja na serwerze Linux) obejmującego materiał z poprzednich zajęć z wykładów oraz ćwiczeń.

W domu studenci realizują kurs e-learningowy oraz odpowiadają na pytania kontrolne. Wypełniają również krzyżówki oraz udzielają się na forum dyskusyjnym rozwiązując zadania „dla dociekliwych”.

Ocena końcowa wyliczana jest średnia ważona z kolokwium testowych (80%), punktów uzyskanych za pytania kontrolne w lekcjach (15%) oraz wyniku krzyżówek (5%). Punkty z rozwiązanych zadań „dla dociekliwych” doliczane są do całej puli.

Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi:

do 50% - ocena niedostateczna,

51% - 60% - ocena dostateczna,

61% - 70% - ocena dostateczna plus,

71% - 80% - ocena dobra

81% - 90% - ocena dobra plus,

powyżej 90% - ocena bardzo dobra.

Osoby, które nie uzyskają zaliczenia zgodnie z powyższym algorytmem, uczestniczą w zaliczeniu poprawkowym.

Podstawy programowania

Nazwa zajęć po angielsku: Fundamentals of Programming

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: I

Forma zaliczenia: Egzamin

Punkty ECTS: 5

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Marcin Zawadzki

Opis zajęć:

Przedmiot wprowadzający do programowania komputerowego w języku C# z wykorzystaniem środowiska Visual Studio. Zajęcia obejmują fundamentalne koncepcje programistyczne: zmienne, typy danych, operatory, instrukcje warunkowe, pętle, tablice, metody (funkcje) oraz struktury. Studenci rozwijają umiejętności logicznego myślenia, analizy problemu i tworzenia działających programów konsolowych. Ponad połowę czasu zajęć stanowią aktywności praktyczne realizowane przy komputerze, ukierunkowane na nabywanie umiejętności zawodowych programisty. Przedmiot kończy się zaliczeniem praktycznym polegającym na samodzielnym napisaniu programu integrującego nabyte umiejętności.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Wyjaśnia pojęcia związane z algorytmiką i programowaniem komputerowym (algorytm, program, kompilator, zmienna, typ danych, proces tworzenia programu od analizy problemu do implementacji) niezbędne do formułowania i rozwiązywania zadań programistycznych.	K_W01
Identyfikuje metody, techniki i narzędzia (Visual Studio, kompilator C#, konsola, platforma e-learningowa) stosowane do tworzenia, kompilowania i uruchamiania programów konsolowych.	K_W05
Identyfikuje i stosuje zasady kodowania w języku C# określone w Microsoft C# Coding Conventions, w szczególności zasady nazewnictwa (PascalCase, camelCase), formatowania i czytelności kodu.	K_W06
Umiejętności (student potrafi)	
Wykorzystuje komunikaty kompilatora, dokumentację języka C# oraz przykłady kodu do samodzielnego identyfikowania i rozwiązywania problemów programistycznych.	K_U01
Posługuje się środowiskiem Visual Studio 2022 do tworzenia, kompilowania, uruchamiania i identyfikacji błędów w programach konsolowych C#.	K_U04
Implementuje program rozwiązujący zadany problem praktyczny z użyciem zmiennych, instrukcji warunkowych, pętli, tablic i metod z parametrami.	K_U05
Planuje dane testowe dla swojego programu (wartości poprawne, graniczne, błędne), przeprowadza testy uruchomieniowe, interpretuje wyniki działania programu (poprawne/błędne wyjście) i wyciąga wnioski prowadzące do korekty kodu.	K_U06
Projektuje i realizuje aplikację konsolową składającą się z wielu współpracujących metod (z parametrami, zwracaniem wartości i przeciążaniem) zgodnie z postawionymi wymaganiami	K_U11

funkcjonalnymi.	
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny jakości własnego kodu, identyfikowania w nim błędów i poprawiania rozwiązań na podstawie informacji zwrotnej od prowadzącego.	K_K01
Student jest gotów do samodzielnego poszukiwania rozwiązań problemów programistycznych w dokumentacji i materiałach dydaktycznych oraz zasięgania opinii prowadzącego w przypadku trudności.	K_K02
Student jest gotów do samodzielnego i rzetelnego wykonywania zadań programistycznych, przestrzegając zasad uczciwości akademickiej (samodzielność rozwiązań, niekopiowanie cudzego kodu) oraz odpowiedzialnego podejścia do oddawania prac.	K_K05

Metody i formy kształcenia:

- Wykład informacyjny
- Pogadanka
- Metoda pokazu
- Metoda ćwiczeń
- Wykład problemowy
- Praca ze źródłem elektronicznym

Metody oceniania:

- Zaliczenie praktyczne
- Aktywność w trakcie zajęć
- Egzamin opisowy

Wykład konwersatoryjny (W)

1. Wprowadzenie do programowania: pojęcie algorytmu, program, kompilator, kod źródłowy. Środowisko Visual Studio 2022, struktura projektu konsolowego w języku C#.
2. Typy danych (int, double, char, bool, string), zmienne, stałe, operatory arytmetyczne, logiczne i relacyjne. Konwersje typów, formatowanie wyników.
3. Instrukcje warunkowe: if-else, if-else-if, instrukcja wielokrotnego wyboru switch. Walidacja danych wejściowych.
4. Instrukcje iteracyjne: pętle for, while, do-while. Pętle zagnieżdżone. Rysowanie wzorów tekstowych.
5. Tablice jednowymiarowe: deklaracja, inicjalizacja, operacje na elementach, przeszukiwanie, właściwość Length. Losowanie liczb (klasa Random).
6. Metody (funkcje): definiowanie, parametry, wartość zwracana, przeciążanie metod. Zmienne lokalne i globalne. Zasięg nazw.
7. Struktury (struct): definiowanie typów złożonych, tablice struktur. Podstawy operacji na plikach tekstowych.
8. Podsumowanie i integracja zagadnień. Budowa programu z menu wyboru łączącego wszystkie poznane konstrukcje języka C#.

Ćwiczenia (Ć)

1. Konfiguracja środowiska Visual Studio 2022. Pierwszy program konsolowy. Kompilacja, uruchamianie, odczyt komunikatów błędów kompilatora.
2. Deklaracja zmiennych, typy danych, operatory. Wczytywanie danych z konsoli (Console.ReadLine), wypisywanie wyników (Console.WriteLine), formatowanie.
3. Instrukcje warunkowe (if-else, switch). Zabezpieczanie programu przed błędnymi danymi wejściowymi (walidacja).
4. Pętle (for, while, do-while). Programy rysujące wzory tekstowe (prostokąt, trójkąt z gwiazdek). Budowa menu programu.
5. Tablice: tworzenie, wypełnianie, odwoływanie do elementów, wypisywanie. Operacje: minimum, maksimum, suma, średnia.
6. Metody: definiowanie z parametrami i wartością zwracaną. Przeciążanie metod. Podział programu na mniejsze funkcje.
7. Struktury: definiowanie własnych typów złożonych. Tablice struktur. Zapis i odczyt danych z pliku.
8. Zaliczenie praktyczne: samodzielne napisanie programu konsolowego integrującego poznane zagadnienia (menu, metody, tablice, walidacja danych).

Literatura podstawowa

1. Matulewski J., „C#. Lekcje programowania”. Wydawnictwo Helion, 2022.
2. Bhargava A., „Algorytmy. Ilustrowany przewodnik”. Wydawnictwo Helion, 2022.

Literatura uzupełniająca

1. Oficjalna dokumentacja języka C# i platformy .NET (learn.microsoft.com/dotnet/csharp)
2. Price M.J., „C# 12 and .NET 8 - Modern Cross-Platform Development Fundamentals”. Packt Publishing, 2023.

Źródła dodatkowe

1. Microsoft Learn - interaktywne moduły szkoleniowe C# (learn.microsoft.com/training/paths/csharp-first-steps)
2. Exercism - interaktywna ścieżka nauki C# (exercism.org/tracks/csharp)
3. Materiały dydaktyczne na platformie e-learningowej (wlodek.wlodkowic.pl)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	60 godz.....	32 godz.
• Zapoznanie się z literaturą przedmiotu.....	15 godz.....	15 godz.
• Samodzielne wykonanie zadań.....	30 godz.....	53 godz.
• Przygotowanie i udział w egzaminie.....	10 godz.....	10 godz.
• Przygotowanie się do zajęć.....	10 godz.....	15 godz.
Suma obciążeń:.....	125 godz.....	125 godz.

Zaliczenie zajęć:

Ćwiczenia: Warunkiem przystąpienia do zaliczenia praktycznego jest systematyczne uczestnictwo w zajęciach potwierdzone przesłaniem rozwiązań zadań programistycznych na platformę e-learningową. Ocena z ćwiczeń wystawiana jest na podstawie wyniku zaliczenia praktycznego (Termin I / Termin II) polegającego na samodzielnym napisaniu programu konsolowego w języku C#.

Egzamin: Egzamin w formie praktycznej - samodzielne napisanie programu rozwiązującego postawione zadania z wykorzystaniem poznanych konstrukcji języka C# (zmienne, instrukcje warunkowe, pętle, tablice, metody z parametrami). Poprawna implementacja programu potwierdza zarówno znajomość zagadnień teoretycznych (typy danych, operatory, struktury sterujące, algorytmy) jak i umiejętność ich praktycznego zastosowania. Forma praktyczna egzaminu jest adekwatna do specyfiki przedmiotu realizowanego na profilu praktycznym kierunku Informatyka.

Kryteria oceniania zaliczenia praktycznego:

- Poprawne rozwiązanie zadań wymagających zastosowania zmiennych, pętli, tablic i metod z parametrami: ocena dostateczna (3.0)
- Poprawne rozwiązanie zadań wymagających zastosowania zmiennych, pętli, tablic i metod z parametrami oraz częściowe rozwiązanie zadań wymagających współpracy wielu metod: ocena dostateczna plus (3.5)
- Poprawne rozwiązanie zadań wymagających wielometodowej struktury programu ze współpracą metod: ocena dobra (4.0)
- Poprawne rozwiązanie zadań wymagających wielometodowej struktury programu ze współpracą metod oraz częściowe rozwiązanie zadań wymagających przeciążania metod i złożonej logiki: ocena dobra plus (4.5)
- Poprawne rozwiązanie wszystkich zadań łącznie z wymagającymi przeciążania metod, złożonej walidacji i pełnej integracji konstrukcji: ocena bardzo dobra (5.0)

Próg osiągnięcia efektów uczenia się: Każdy efekt uczenia się uznaje się za osiągnięty po uzyskaniu minimum oceny dostatecznej (3.0) z zaliczenia praktycznego.

Weryfikacja kompetencji społecznych:

- PEU_K01 (krytyczna ocena kodu): weryfikowana przez iteracyjne poprawianie rozwiązań zadań na platformie w trakcie semestru.
- PEU_K02 (poszukiwanie rozwiązań): weryfikowana przez samodzielne rozwiązywanie zadań z wykorzystaniem udostępnionych materiałów.
- PEU_K03 (etyka, samodzielność): weryfikowana przez samodzielne wykonanie programu zaliczeniowego przy komputerze pod nadzorem prowadzącego.

Algorytmy i złożoność

Nazwa zajęć po angielsku: Algorithms and complexity

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: I

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Wojciech Korzybski

Opis zajęć:

W ramach przedmiotu student poznaje podstawowe pojęcia, metody i techniki projektowania algorytmów oraz sposoby ich zapisu, analizy i implementacji. Omawiane są algorytmy iteracyjne i rekurencyjne, podstawowe struktury algorytmiczne, wybrane metody projektowania algorytmów, w tym strategia „dziel i zwyciężaj”, a także algorytmy wyszukiwania i sortowania. Istotnym elementem zajęć jest analiza poprawności oraz złożoności czasowej i pamięciowej algorytmów, pozwalająca ocenić efektywność różnych sposobów rozwiązania tego samego problemu. Zajęcia mają charakter wprowadzający i praktyczny. Student uczy się formułować problem algorytmiczny, dobierać sposób reprezentacji algorytmu, tworzyć schematy blokowe i pseudokod, implementować wybrane rozwiązania oraz testować ich działanie dla różnych danych wejściowych. Przedmiot przygotowuje do dalszego kształcenia w zakresie programowania, struktur danych, baz danych, sztucznej inteligencji, optymalizacji i projektowania efektywnych rozwiązań informatycznych.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia podstawowe pojęcia algorytmu, danych wejściowych, danych wyjściowych, specyfikacji, poprawności i złożoności obliczeniowej	K_W01
charakteryzuje zależności między strukturą algorytmu, liczbą operacji, rozmiarem danych i efektywnością rozwiązania	K_W02
rozpoznaje podstawowe metody, techniki i narzędzia stosowane przy projektowaniu, zapisie, analizie i implementacji algorytmów	K_W05
<i>Umiejętności (student potrafi)</i>	
formułuje problem algorytmiczny, dobiera sposób jego reprezentacji i wykorzystuje wiedzę do zaproponowania rozwiązania	K_U01
implementuje wybrane algorytmy iteracyjne, rekurencyjne, sortujące i wykorzystujące strategię „dziel i zwyciężaj”	K_U05
analizuje poprawność oraz złożoność czasową i pamięciową wybranych algorytmów	K_U06
stosuje metody analityczne i eksperymentalne do porównywania działania algorytmów dla różnych danych wejściowych	K_U07
rozwiązuje praktyczne zadania algorytmiczne z użyciem właściwych narzędzi, schematów blokowych lub pseudokodu	K_U12
<i>Kompetencje społeczne (student jest gotów do)</i>	
jest gotów zasięgać opinii ekspertów i korzystać z dokumentacji lub literatury, gdy nie potrafi	K_K01

Metody i formy kształcenia:

- Wykład informacyjny
- Metoda ćwiczeń
- Metoda pokazu
- Praca ze źródłem elektronicznym

Metody oceniania:

- Aktywność w trakcie zajęć
- Zaliczenie praktyczne
- Zaliczenie testowe

Wykład konwersatoryjny (W)

1. Wprowadzenie do teorii algorytmów
2. Podstawowe struktury algorytmiczne i specyfikacja algorytmów
3. Przegląd metod i technik projektowania algorytmów
4. Elementy analizy algorytmów
5. Algorytmy iteracyjne i rekurencyjne
6. Algorytmy dotyczące strategii "dziel i zwyciężaj"
7. Podstawowe algorytmy sortowania
8. Zaawansowane algorytmy sortowania

Ćwiczenia (Ć)

1. Wprowadzenie do algorytmów i budowanie sieci działań
2. Elementy schematu blokowego i zasady budowania algorytmów
3. Implementacja wybranych schematów blokowych
4. Analiza schematów blokowych
5. Budowanie złożonych schematów blokowych (iteracyjnych i rekurencyjnych)
6. Algorytmy dotyczące strategii "dziel i zwyciężaj"
7. Algorytmy sortujące: przez wybór, bąbelkowe i przez wstawianie
8. Algorytmy sortujące typu: QuickSort, MergeSort

Literatura podstawowa

1. Wróblewski P.: Algorytmy. Struktury danych i techniki programowania. Helion 2019.

Literatura uzupełniająca

1. Kluczewski J., Smyczek M. (red.): Algorytmika dla studenta i technika programisty, ITSTART 2023

Źródła dodatkowe

1. Flowgorithm <http://www.flowgorithm.org/>
2. OpenDSA <https://opensa-server.cs.vt.edu/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- | | | |
|---|--------------|----------|
| • Zajęcia z udziałem nauczyciela akademickiego..... | 45 godz..... | 24 godz. |
| • Zapoznanie się z literaturą przedmiotu..... | 15 godz..... | 26 godz. |
| • Samodzielne wykonanie zadań..... | 15 godz..... | 25 godz. |
| Suma obciążeń:..... | 75 godz..... | 75 godz. |

Zaliczenie zajęć:

Zaliczenie ćwiczeń polega na wykonaniu zadań praktycznych poprzez realizację schematów blokowych przykładowych algorytmów omawianych na zajęciach.
Zaliczenie wykładu wymaga wykonania z wynikiem pozytywnym (ponad 50%) testu komputerowego.

Języki i paradygmaty programowania I

Nazwa zajęć po angielsku: Programming Languages and Paradigms I

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: II

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 5

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....45 / 24 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Marcin Zawadzki

Opis zajęć:

Przedmiot stanowi kontynuację kursu „Podstawy programowania” i rozszerza umiejętności programowania proceduralnego w języku C# z wykorzystaniem środowiska Visual Studio. Zajęcia obejmują zaawansowane operacje na tablicach wielowymiarowych (macierze), metody z różnymi typami parametrów, rekurencję, struktury danych, operacje na plikach tekstowych oraz wybrane algorytmy. Studenci realizują zadania o rosnącej złożoności - od operacji na macierzach, przez szyfry i konwersje systemów liczbowych, po kompletne aplikacje konsolowe (np. system zarządzania danymi). Ćwiczenia stanowią dominującą formę zajęć, ukierunkowaną na samodzielne rozwiązywanie problemów programistycznych.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
Wyjaśnia pojęcia związane z tablicami wielowymiarowymi, rekurencją, strukturami danych i algorytmami (sortowanie, konwersje, szyfrowanie) niezbędne do rozwiązywania zadań programistycznych.	K_W01
Charakteryzuje zasady organizacji i struktury programu w języku C# (podział na metody, czytelność, konwencje nazewnicze) stosowane przy tworzeniu wielometodowych aplikacji konsolowych.	K_W03
Identyfikuje metody i techniki (metody z parametrami różnych typów, rekurencja, operacje na plikach, kolekcje) stosowane do rozwiązywania problemów programistycznych w języku C#.	K_W05
<i>Umiejętności (student potrafi)</i>	
Analizuje problem algorytmiczny i implementuje jego rozwiązanie z użyciem tablic wielowymiarowych, rekurencji, struktur i operacji na plikach.	K_U01
Posługuje się środowiskiem Visual Studio do tworzenia, kompilowania, uruchamiania i identyfikacji błędów w wielometodowych programach konsolowych C#.	K_U04
Projektuje i implementuje aplikację konsolową rozwiązującą złożony problem praktyczny (system zarządzania danymi, gra, szyfr) z użyciem struktur, tablic, metod i plików.	K_U05
Planuje dane testowe, weryfikuje poprawność programu na przypadkach brzegowych (puste tablice, wartości graniczne, błędne dane) i wyciąga wnioski prowadzące do korekty kodu.	K_U06
Projektuje i realizuje kompletną aplikację konsolową składającą się ze struktur danych, wielu współpracujących metod, menu użytkownika i operacji na plikach.	K_U11
<i>Kompetencje społeczne (student jest gotów do)</i>	

Student jest gotów do krytycznej oceny jakości, czytelności i efektywności własnych rozwiązań programistycznych oraz poprawiania ich na podstawie informacji zwrotnej.	K_K01
Student jest gotów do samodzielnego poszukiwania rozwiązań problemów algorytmicznych w dokumentacji i materiałach dydaktycznych oraz zasięgania opinii prowadzącego w przypadku trudności.	K_K02
Student jest gotów do samodzielnego i rzetelnego wykonywania zadań programistycznych, przestrzegając zasad uczciwości akademickiej (samodzielność rozwiązań, niekopiowanie cudzego kodu) oraz odpowiedzialnego podejścia do oddawania prac.	K_K05

Metody i formy kształcenia:

- Wykład informacyjny
- Praca ze źródłem elektronicznym
- Metoda ćwiczeń
- Metoda pokazu
- Pogadanka
- Wykład problemowy

Metody oceniania:

- Zaliczenie praktyczne
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

1. Powtórzenie i rozszerzenie: zmienne, operatory, pętle, tablice jednowymiarowe. Wprowadzenie do tablic wielowymiarowych.
2. Tablice dwuwymiarowe (macierze): deklaracja, operacje na elementach, iloczyn skalarny wektorów, transpozycja macierzy, sortowanie elementów.
3. Metody (funkcje) zaawansowane: typy metod, parametry, wartości zwracane, metody wbudowane i użytkownika. Budowa programu z menu.
4. Rekurencja: definicja, silnia iteracyjna vs rekurencyjna, zastosowania rekurencji w algorytmach.
5. Konwersje systemów liczbowych (binarny, dziesiętny), Schemat Hornera. Instrukcja wielokrotnego wyboru (switch).
6. Struktury (struct): definiowanie typów złożonych, tablice struktur, kolekcje (List). Operacje na plikach tekstowych (zapis, odczyt).
7. Algorytmy: sortowanie elementów tablicy, wyszukiwanie minimum, szyfr Cezara. Podsumowanie i integracja zagadnień.

Ćwiczenia (Ć)

1. Powtórzenie tablic jednowymiarowych. Operacje: tworzenie, wypełnianie, odwoływanie, Length.
2. Tablice dwuwymiarowe: deklaracja, wczytywanie elementów, wypisywanie macierzy.
3. Operacje na macierzach: transpozycja, iloczyn skalarny, wyszukiwanie minimum w tablicy.
4. Pętle zaawansowane: zagnieżdżone pętle, generowanie wzorów, konwersje systemów liczbowych.
5. Metody: definiowanie z parametrami, wartość zwracana, podział programu na funkcje.
6. Metody zaawansowane: budowa menu programu, przeciążanie, współpraca wielu metod.
7. Rekurencja w praktyce: silnia, potęga, inne zastosowania algorytmiczne.
8. Instrukcja switch. Walidacja danych wejściowych. Testowanie programu: projektowanie przypadków testowych, analiza błędów i poprawa rozwiązania.
9. Struktury: definiowanie, tablice struktur, operacje CRUD na zbiorach danych.
10. Kolekcje (List) i ich zastosowanie w zarządzaniu danymi.
11. Pliki tekstowe: zapis i odczyt danych, integracja z programem (np. zapis stanu gry/danych).
12. Algorytmy: sortowanie tablicy, szyfr Cezara, wyszukiwanie najczęstszego elementu.
13. Projekt integracyjny: aplikacja konsolowa łącząca struktury, metody, pliki i menu (np. system zarządzania biblioteką).
14. Projekt integracyjny (kontynuacja): testowanie, poprawki, rozbudowa funkcjonalności.
15. Zaliczenie praktyczne: samodzielne napisanie programu konsolowego integrującego poznane zagadnienia.

Literatura podstawowa

1. Marcin Jamro, "Struktury danych i algorytmy w języku C#. Projektowanie efektywnych aplikacji". Wydawnictwo Helion, 2019

2. Price M.J., "C# 12 and .NET 8 - Modern Cross-Platform Development Fundamentals". Packt, 2023.

Literatura uzupełniająca

1. Rodenburg J., „Code like a Pro in C#”. Manning Publications, 2021.
2. Joseph Albahari, "C# 12 w pigułce. Kompendium programisty", Wydawnictwo Helion, 2024
3. Jennifer Greene, Andrew Stellman, "C#. Rusz głową! Wydanie III". Wydawnictwo Helion, 2014
4. Oficjalna dokumentacja C# i .NET (learn.microsoft.com/dotnet/csharp)
5. Bhargava A., "Algorytmy. Ilustrowany przewodnik". Wydawnictwo Helion, 2022.

Źródła dodatkowe

1. Microsoft Docs - Dokumentacja języka C#
2. Pluralsight - Kursy online z C#
3. Stack Overflow - Forum programistyczne
4. Microsoft Learn - moduły szkoleniowe C# (learn.microsoft.com/training)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	60 godz.....	32 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	10 godz.
• Samodzielne wykonanie zadań.....	35 godz.....	63 godz.
• Przygotowanie się do zaliczenia końcowego.....	10 godz.....	10 godz.
• Przygotowanie się do zajęć.....	10 godz.....	10 godz.
Suma obciążeń:.....	125 godz.....	125 godz.

Zaliczenie zajęć:

Ćwiczenia: Warunkiem przystąpienia do zaliczenia praktycznego jest systematyczne uczestnictwo w zajęciach potwierdzone przesłaniem rozwiązań zadań programistycznych na platformę e-learningową. Ocena końcowa z przedmiotu wystawiana jest na podstawie wyniku zaliczenia praktycznego (Termin I / Termin II) polegającego na samodzielnym napisaniu programu konsolowego w języku C#. Poprawna implementacja programu potwierdza zarówno znajomość zagadnień teoretycznych z wykładu jak i umiejętność ich praktycznego zastosowania.

Kryteria oceniania zaliczenia praktycznego:

- Poprawne rozwiązanie zadań wymagających zastosowania tablic, pętli i metod z parametrami: ocena dostateczna (3.0)
- Poprawne rozwiązanie zadań wymagających zastosowania tablic, pętli i metod z parametrami oraz częściowe rozwiązanie zadań wymagających rekurencji lub operacji na strukturach: ocena dostateczna plus (3.5)
- Poprawne rozwiązanie zadań wymagających wielometodowej struktury programu z rekurencją lub strukturami danych: ocena dobra (4.0)
- Poprawne rozwiązanie zadań wymagających wielometodowej struktury programu z rekurencją lub strukturami danych oraz częściowe rozwiązanie zadań wymagających integracji wielu konstrukcji (struktury + pliki + menu): ocena dobra plus (4.5)
- Poprawne rozwiązanie wszystkich zadań łącznie z wymagającymi pełnej integracji konstrukcji (struktury, pliki, menu, walidacja): ocena bardzo dobra (5.0)

Próg osiągnięcia efektów uczenia się: Efekty wiedzy (PEU_W01-W03) i umiejętności (PEU_U01-U05) uznaje się za osiągnięte po uzyskaniu minimum oceny dostatecznej (3.0) z zaliczenia praktycznego. Kompetencje społeczne (PEU_K01-K03) uznaje się za osiągnięte po spełnieniu warunków określonych dla przypisanych im metod weryfikacji, odpowiednio do zakresu danego efektu, w tym poprzez systematyczne przysyłanie i poprawianie rozwiązań, samodzielną pracę oraz - w przypadku wystąpienia trudności - zasięganie opinii prowadzącego.

Weryfikacja kompetencji społecznych:

- PEU_K01 (krytyczna ocena kodu): weryfikowana przez iteracyjne poprawianie rozwiązań zadań na platformie w trakcie semestru (historia kolejnych wersji/przesłań).
- PEU_K02 (poszukiwanie rozwiązań): weryfikowana przez samodzielne rozwiązywanie zadań z wykorzystaniem udostępnionych materiałów; w przypadku trudności - konsultacja z prowadzącym.
- PEU_K03 (etyka, samodzielność): weryfikowana przez samodzielne wykonanie programu zaliczeniowego przy komputerze pod nadzorem prowadzącego.

Języki i paradygmaty programowania II

Nazwa zajęć po angielsku: Programming Languages and Paradigms II

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: III

Forma zaliczenia: Egzamin

Punkty ECTS: 6

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....45 / 24 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Marcin Zawadzki

Opis zajęć:

Przedmiot stanowi kontynuację kursu „Języki i paradygmaty programowania I” i rozwija umiejętności programowania w języku C# z wykorzystaniem środowiska Visual Studio, ze szczególnym uwzględnieniem programowania imperatywnego i obiektowego oraz elementów paradygmatu funkcyjnego realizowanych z wykorzystaniem LINQ. Studenci pogłębiają znajomość algorytmów sortowania i wyszukiwania, poznają programowanie obiektowe (klasy, dziedziczenie, polimorfizm), kolekcje generyczne (List<T>), LINQ jako narzędzie deklaratywnego przetwarzania danych oraz zaawansowane operacje na plikach tekstowych. Zajęcia kończą się indywidualnym projektem programistycznym (np. gra konsolowa RPG) integrującym wszystkie nabyte umiejętności. Ze względu na praktyczny profil kształcenia zasadnicza część osiągania efektów uczenia się realizowana jest podczas ćwiczeń i projektu, natomiast wykład konwersatoryjny pełni funkcję wprowadzającą, porządkującą i problemową.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Wyjaśnia pojęcia związane z algorytmami sortowania (bąbelkowe, przez wstawianie, szybkie), rekurencją, złożonością algorytmów oraz strukturami danych (kolekcje generyczne) niezbędne do formułowania i rozwiązywania zadań programistycznych.	K_W01
Charakteryzuje zasady programowania obiektowego (enkapsulacja, dziedziczenie, polimorfizm) oraz ich zastosowanie w organizacji i strukturze większych projektów programistycznych.	K_W03
Identyfikuje metody i techniki (kolekcje generyczne, LINQ, operacje na plikach, programowanie obiektowe) stosowane do projektowania i implementacji aplikacji w języku C#.	K_W05
Umiejętności (student potrafi)	
Analizuje problem algorytmiczny i implementuje jego rozwiązanie z użyciem odpowiednich algorytmów sortowania, struktur obiektowych, kolekcji i operacji na plikach.	K_U01
Posługuje się środowiskiem Visual Studio do tworzenia, kompilowania, debugowania i uruchamiania wieloplukowych projektów C# wykorzystujących klasy i dziedziczenie.	K_U04
Projektuje i implementuje aplikację konsolową rozwiązującą złożony problem (gra, system zarządzania danymi) z zastosowaniem klas, dziedziczenia, kolekcji, plików i menu.	K_U05
Planuje dane testowe, porównuje efektywność różnych algorytmów sortowania, weryfikuje poprawność programu na przypadkach brzegowych i wyciąga wnioski prowadzące do optymalizacji kodu.	K_U06

Projektuje i realizuje kompletny projekt programistyczny (grę konsolową RPG) integrujący klasy z dziedziczeniem, system zapisu/odczytu stanu z pliku, menu użytkownika i logikę gry.	K_U11
Planuje etapy realizacji indywidualnego projektu programistycznego, dekomponuje problem na moduły i zarządza postępowaniem prac w ramach wyznaczonego harmonogramu.	K_U17
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny jakości, czytelności i efektywności własnych rozwiązań programistycznych oraz poprawiania ich na podstawie informacji zwrotnej.	K_K01
Student jest gotów do samodzielnego poszukiwania rozwiązań problemów projektowych w dokumentacji i materiałach dydaktycznych oraz zasięgania opinii prowadzącego w przypadku trudności.	K_K02
Student jest gotów do samodzielnego i rzetelnego wykonywania zadań programistycznych i projektu, przestrzegając zasad uczciwości akademickiej (samodzielność rozwiązań, niekopiowanie cudzego kodu) oraz odpowiedzialnego podejścia do oddawania prac.	K_K05

Metody i formy kształcenia:

- Pogadanka
- Praca ze źródłem elektronicznym
- Metoda pokazu
- Metoda ćwiczeń
- Wykład problemowy
- Metoda projektu

Metody oceniania:

- Rozliczenie projektu
- Zaliczenie praktyczne
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

1. Powtórzenie i rozszerzenie: metody zaawansowane, rekurencja, struktury, kolekcje. Wprowadzenie do algorytmów sortowania.
2. Algorytmy sortowania: bąbelkowe, przez wstawianie, przez wybieranie, szybkie. Wizualizacja procesu sortowania.
3. Algorytmy wyszukiwania i przetwarzania danych. LINQ jako element programowania funkcyjnego - deklaratywne filtrowanie, sortowanie i transformacja kolekcji.
4. Programowanie obiektowe - klasy: pola, właściwości, konstruktory, metody. Enkapsulacja.
5. Programowanie obiektowe - dziedziczenie: klasa bazowa i pochodna, słowo kluczowe base, nadpisywanie metod (virtual/override).
6. Programowanie obiektowe - polimorfizm: metody wirtualne, rzutowanie typów, zastosowania praktyczne.
7. Operacje na plikach tekstowych: zaawansowany zapis/odczyt, serializacja stanu programu. Integracja zagadnień semestru.

Ćwiczenia (Ć)

1. Powtórzenie: tablice wielowymiarowe, macierze - iloczyn skalarny, transpozycja, operacje.
2. Metody zaawansowane: wielometodowa struktura programu, menu, współpraca metod.
3. Rekurencja w praktyce: silnia, potęga, zastosowania algorytmiczne.
4. Algorytmy sortowania (implementacja): sortowanie bąbelkowe, przez wstawianie, przez wybieranie.
5. Algorytmy sortowania (zaawansowane): sortowanie szybkie, wizualizacja procesu w konsoli.
6. Kolekcje generyczne (List<T>): tworzenie, operacje CRUD, zastosowania w zarządzaniu danymi.
7. LINQ jako element paradygmatu funkcyjnego: deklaratywne filtrowanie, sortowanie i przetwarzanie kolekcji. Zadania praktyczne.
8. Programowanie obiektowe - klasy: definiowanie klas, tworzenie obiektów, konstruktory, metody.
9. Programowanie obiektowe - dziedziczenie: klasa bazowa, klasy pochodne, konstruktory z base.
10. Programowanie obiektowe - polimorfizm: virtual/override, zastosowania (np. system produktów w sklepie).
11. Pliki tekstowe zaawansowane: zapis i odczyt złożonych struktur danych, parsowanie, obsługa błędów I/O.
12. Zadania integracyjne: gra w Życie (automaty komórkowe), kółko i krzyżyk, histogram wartości.
13. Projekt - planowanie: wybór tematu (gra RPG), dekompozycja na moduły, określenie wymagań funkcjonalnych.
14. Projekt - implementacja: klasy postaci (dziedziczenie), system walki, ekwipunek, zapis/odczyt stanu gry.

15. Projekt - finalizacja: testowanie, poprawki, prezentacja działającego projektu.

Projekt (P)

1. Definiowanie wymagań projektu (gra konsolowa RPG): klasy postaci, system walki turowej, ekwipunek, złoto i sklep.
2. Implementacja projektu: struktura klas z dziedziczeniem, mechanika gry, system zapisu/odczytu stanu z pliku.
3. Testowanie, prezentacja projektu i obrona rozwiązań implementacyjnych.

Literatura podstawowa

1. Price M.J., "C# 12 i .NET 8 dla programistów aplikacji wieloplatformowych. Twórz aplikacje, witryny WWW oraz serwisy sieciowe za pomocą ASP.NET Core 8, Blazor i EF Core 8. Wydanie VIII". Helion 2024
2. Jamro M., "Struktury danych i algorytmy w języku C#. Projektowanie efektywnych aplikacji". Wydawnictwo Helion, 2019

Literatura uzupełniająca

1. Price M.J., "C# 12 and .NET 8 - Modern Cross-Platform Development Fundamentals". Packt, 2023.
2. Oficjalna dokumentacja C# i .NET (learn.microsoft.com/dotnet/csharp)
3. Bhargava A., "Algorytmy. Ilustrowany przewodnik". Wydawnictwo Helion, 2022.

Źródła dodatkowe

1. Microsoft Learn - moduły szkoleniowe C# (learn.microsoft.com/training)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	10 godz.
• Przygotowanie się do zajęć.....	10 godz.....	10 godz.
• Samodzielne wykonanie zadań.....	15 godz.....	35 godz.
• Przygotowanie projektu.....	40 godz.....	55 godz.
Suma obciążeń:.....	150 godz.....	150 godz.

Zaliczenie zajęć:

Przedmiot składa się z trzech komponentów: wykładu konwersatoryjnego, ćwiczeń i projektu. Warunkiem przystąpienia do egzaminu praktycznego jest uzyskanie zaliczenia z ćwiczeń (systematyczne przysyłanie rozwiązań) oraz pozytywne rozliczenie projektu.

Egzamin praktyczny (Termin I / Termin II) polega na samodzielnym napisaniu programu konsolowego w języku C# integrującego zagadnienia z całego semestru. Ocena z egzaminu jest oceną końcową z przedmiotu.

Kryteria oceniania egzaminu praktycznego:

- Poprawne rozwiązanie zadań wymagających zastosowania tablic, pętli, metod i kolekcji: ocena dostateczna (3.0)
- Poprawne rozwiązanie zadań wymagających zastosowania tablic, pętli, metod i kolekcji oraz częściowe rozwiązanie zadań wymagających programowania obiektowego (klasy, dziedziczenie): ocena dostateczna plus (3.5)
- Poprawne rozwiązanie zadań wymagających programowania obiektowego z dziedziczeniem i polimorfizmem oraz zastosowania rekurencji, z uzasadnieniem wyboru algorytmu z uwzględnieniem jego złożoności: ocena dobra (4.0)
- Poprawne rozwiązanie zadań wymagających programowania obiektowego z dziedziczeniem, polimorfizmem i rekurencją oraz częściowe rozwiązanie zadań wymagających integracji wielu konstrukcji (klasy + pliki + LINQ + menu): ocena dobra plus (4.5)
- Poprawne rozwiązanie wszystkich zadań łącznie z wymagającymi pełnej integracji konstrukcji (programowanie obiektowe, pliki, LINQ, menu, walidacja): ocena bardzo dobra (5.0)

Warunki zaliczenia projektu:

Projekt indywidualny (gra konsolowa RPG lub równoważna aplikacja) oceniany na podstawie:

- Działający program spełniający wymagania funkcjonalne
- Zastosowanie programowania obiektowego (klasy z dziedziczeniem)
- Jakość i czytelność kodu (nazewnictwo, podział odpowiedzialności klas, unikanie duplikacji)
- Zapis i odczyt stanu programu z pliku
- Menu użytkownika i logika aplikacji
- Prezentacja i obrona rozwiązań implementacyjnych

Próg osiągnięcia efektów uczenia się: Efekty wiedzy (PEU_W01-W03) i umiejętności (PEU_U01-U06) uznaje się za osiągnięte po uzyskaniu minimum oceny dostatecznej (3.0) z egzaminu praktycznego oraz pozytywnym rozliczeniu projektu. Kompetencje społeczne (PEU_K01-K03) uznaje się za osiągnięte po spełnieniu warunków określonych dla przypisanych im metod weryfikacji, odpowiednio do zakresu danego efektu, w tym poprzez systematyczne przysyłanie

i poprawianie rozwiązań, samodzielną pracę nad projektem oraz - w przypadku wystąpienia trudności - zasięganie opinii prowadzącego.

Weryfikacja kompetencji społecznych:

- PEU_K01 (krytyczna ocena kodu): weryfikowana przez iteracyjne poprawianie rozwiązań zadań i projektu na platformie w trakcie semestru.
- PEU_K02 (poszukiwanie rozwiązań): weryfikowana przez samodzielną realizację projektu z wykorzystaniem udostępnionych materiałów; w przypadku trudności - konsultacja z prowadzącym.
- PEU_K03 (etyka, samodzielność): weryfikowana przez samodzielne wykonanie programu egzaminacyjnego przy komputerze pod nadzorem prowadzącego oraz indywidualną realizację projektu.

Programowanie obiektowe

Nazwa zajęć po angielsku: Object-Oriented Programming

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: IV

Forma zaliczenia: Egzamin

Punkty ECTS: 5

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....45 / 24 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Marcin Zawadzki

Opis zajęć:

Przedmiot stanowi kontynuację ścieżki programistycznej rozpoczętej w „Podstawach programowania” oraz „Językach i paradygmatach programowania I–II” i jest w całości poświęcony paradygmatowi obiektowemu. Studenci pogłębiają znajomość programowania obiektowego w języku C# z wykorzystaniem środowiska Visual Studio: projektują i implementują systemy złożone z wielu klas z zastosowaniem enkapsulacji, dziedziczenia, polimorfizmu, klas abstrakcyjnych, interfejsów, kompozycji oraz typów wyliczeniowych. Zapoznają się z zasadami SOLID oraz delegatami jako mechanizmem przekazywania zachowania i pracy z kolekcjami. Zajęcia kończą się zespołowym projektem programistycznym integrującym wszystkie nabyte umiejętności. Ze względu na praktyczny profil kształcenia zasadnicza część osiągania efektów uczenia się realizowana jest podczas ćwiczeń i projektu, natomiast wykład konwersatoryjny pełni funkcję wprowadzającą, porządkującą i problemową.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Wyjaśnia zasady enkapsulacji (modyfikatory dostępu, właściwości) oraz ich rolę w projektowaniu klas odpornych na błędne użycie.	K_W03
Wyjaśnia mechanizmy dziedziczenia i polimorfizmu (klasy abstrakcyjne, metody wirtualne, nadpisywanie) oraz ich zastosowanie w budowie hierarchii klas.	K_W03
Wyjaśnia mechanizmy dziedziczenia i polimorfizmu (klasy abstrakcyjne, metody wirtualne, nadpisywanie) oraz ich zastosowanie w budowie hierarchii klas.	K_W03
Opisuje rolę kolekcji obiektowych (List<T>), typów wyliczeniowych (enum) oraz delegatów (Func, Action, Predicate) jako narzędzi stosowanych przy budowie aplikacji obiektowych w języku C#.	K_W05
Wyjaśnia zasady SOLID jako zbiór reguł poprawiających jakość, czytelność i modyfikowalność kodu obiektowego oraz wskazuje ich zastosowanie w praktyce programistycznej.	K_W05
Opisuje interfejsy jako kontrakty definiujące zachowanie obiektów oraz kompozycję jako alternatywę dla dziedziczenia w projektowaniu relacji między klasami.	K_W05
Opisuje rolę kolekcji obiektowych (List<T>), typów wyliczeniowych (enum) oraz delegatów (Func, Action, Predicate) jako narzędzi stosowanych przy budowie aplikacji obiektowych w języku C#.	K_W05
Wyjaśnia zasady SOLID jako zbiór reguł poprawiających jakość, czytelność i modyfikowalność kodu obiektowego oraz wskazuje ich zastosowanie w praktyce	K_W05

programistycznej.	
Umiejętności (student potrafi)	
Analizuje wymagania funkcjonalne i projektuje strukturę klas (pola, właściwości, metody, konstruktory) odpowiednią do rozwiązywanego problemu.	K_U01
Posługuje się środowiskiem Visual Studio do tworzenia, kompilowania, debugowania i uruchamiania wieloplukowych projektów obiektowych C# z podziałem klas na oddzielne pliki.	K_U04
Implementuje hierarchię klas z zastosowaniem dziedziczenia, polimorfizmu i klas abstrakcyjnych do rozwiązania problemu wymagającego wielopoziomowej abstrakcji.	K_U05
Implementuje system obiektowy z zastosowaniem kompozycji, interfejsów, kolekcji obiektowych i walidacji danych do rozwiązania złożonego problemu praktycznego.	K_U05
Implementuje system obiektowy z zastosowaniem kompozycji, interfejsów, kolekcji obiektowych i walidacji danych do rozwiązania złożonego problemu praktycznego.	K_U05
Planuje zestaw danych testowych dla programu obiektowego, przeprowadza testy weryfikujące poprawność relacji między klasami, interpretuje uzyskane wyniki i wyciąga wnioski prowadzące do korekty struktury klas lub logiki metod.	K_U06
Projektuje i realizuje kompletny projekt zespołowy w C# integrujący hierarchię klas, operacje CRUD, walidację danych, obsługę wyjątków i dokumentację techniczną.	K_U11
Planuje i realizuje projekt w zespole 2-4 osobowym: dekomponuje problem na moduły, rozdziela zadania, koordynuje pracę z wykorzystaniem narzędzi do zarządzania zadaniami i prezentuje własny wkład podczas obrony.	K_U17
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny jakości, czytelności i struktury własnych rozwiązań obiektowych oraz poprawiania ich na podstawie informacji zwrotnej od prowadzącego i współpracowników.	K_K01
Student jest gotów do samodzielnego poszukiwania rozwiązań problemów projektowych w dokumentacji i materiałach dydaktycznych, a w przypadku trudności - do odpowiedzialnego korzystania z pomocy prowadzącego i współpracowników w zespole.	K_K02

Metody i formy kształcenia:

- Metoda pokazu
- Metoda ćwiczeń
- Praca ze źródłem elektronicznym
- Metoda projektu
- Wykład problemowy

Metody oceniania:

- Aktywność w trakcie zajęć
- Egzamin opisowy
- Rozliczenie projektu
- Zaliczenie praktyczne

Wykład konwersatoryjny (W)

1. Programowanie obiektowe jako paradygmat: obiekt, klasa, odpowiedzialność, stan, zachowanie. Pola, właściwości, konstruktory.
2. Enkapsulacja: modyfikatory dostępu, prywatne pola, właściwości z getterami i setterami. Walidacja danych w klasach.
3. Dziedziczenie: klasa bazowa i pochodna, słowo kluczowe base, nadpisywanie metod. Klasy abstrakcyjne i metody abstrakcyjne.
4. Polimorfizm: metody wirtualne (virtual/override), rzutowanie typów, zastosowania praktyczne. Interfejsy.
5. Kompozycja i agregacja: obiekty jako pola klas, relacje między klasami. Typy wyliczeniowe (enum).
6. Kolekcje obiektowe (List<T>): przechowywanie i zarządzanie zbiorami obiektów. Przeciążanie metod i operatorów.
7. Delegaty (Func, Action, Predicate): mechanizmy przekazywania zachowania. Zasady SOLID - wprowadzenie do dobrych praktyk projektowania obiektowego.

Ćwiczenia (Ć)

1. Projektowanie klas: definiowanie pól, właściwości, konstruktorów i metod. Tworzenie obiektów.
2. Enkapsulacja w praktyce: prywatne pola, getter/setter, walidacja danych wejściowych.
3. Klasy z logiką: implementacja klasy reprezentującej obiekt matematyczny (wektor, macierz) z operacjami.
4. Własne kolekcje: implementacja klasy zarządzającej tablicą obiektów (dodawanie, usuwanie, wyszukiwanie).
5. Dziedziczenie: klasa bazowa i pochodne, konstruktory z base, nadpisywanie metod.
6. Polimorfizm i klasy abstrakcyjne: hierarchie klas z metodami abstrakcyjnymi (np. system płatności, hierarchia pojazdów).
7. Interfejsy: definiowanie kontraktów, implementacja w klasach, zastosowania praktyczne.
8. Kompozycja: obiekty jako składniki innych obiektów. Typy wyliczeniowe (enum) w modelowaniu dziedziny.
9. System obiektowy: projektowanie wieloklasowej aplikacji z enkapsulacją, dziedziczeniem i kompozycją (np. system rezerwacji hotelowej).
10. System obiektowy zaawansowany: klasy abstrakcyjne, interfejsy, walidacja, obsługa wyjątków (np. system obsługi przesyłek).
11. Kolekcje obiektowe (List<T>): zarządzanie zbiorami obiektów, filtrowanie, wyszukiwanie.
12. Delegaty: Func, Action, Predicate, Comparison, Converter - zastosowania praktyczne w zarządzaniu kolekcjami.
13. Zapis i odczyt obiektów z plików tekstowych. Serializacja stanu aplikacji.

Projekt (P)

1. Planowanie: wybór tematu, dekompozycja na moduły, podział zadań w zespole, ustalenie wymagań funkcjonalnych.
2. Projektowanie: struktura klas z dziedziczeniem i interfejsami, podział odpowiedzialności, plan implementacji.
3. Implementacja: hierarchia klas, operacje CRUD, walidacja danych, obsługa wyjątków.
4. Integracja i testowanie: łączenie modułów, testowanie na danych, naprawa błędów, dokumentacja techniczna.
5. Prezentacja i obrona: prezentacja projektu, indywidualna obrona rozwiązań, odpowiedzi na pytania prowadzącego.

Literatura podstawowa

1. Price M.J., "C# 12 and .NET 8 - Modern Cross-Platform Development Fundamentals". Packt, 2023.
2. Martin R.C., "Czysty kod. Podrecznik dobrego programisty". Wydawnictwo Helion, 2018.

Literatura uzupełniająca

1. Gamma E. i in., "Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku". Helion, 2010.
2. Oficjalna dokumentacja C# i .NET (learn.microsoft.com/dotnet/csharp)

Źródła dodatkowe

1. Microsoft Learn - Object-Oriented Programming in C# (learn.microsoft.com/dotnet/csharp/fundamentals/tutorials/oop)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Zapoznanie się z literaturą przedmiotu.....	6 godz.....	6 godz.
• Samodzielne wykonanie zadań.....	10 godz.....	20 godz.
• Przygotowanie projektu.....	25 godz.....	40 godz.
• Przygotowanie i udział w egzaminie.....	9 godz.....	19 godz.
Suma obciążeń:.....	125 godz.....	125 godz.

Zaliczenie zajęć:

Przedmiot składa się z trzech komponentów: wykładu konwersatoryjnego, ćwiczeń i projektu. Warunkiem przystąpienia do egzaminu praktycznego jest uzyskanie zaliczenia z ćwiczeń (systematyczne przysyłanie rozwiązań zadań obiektowych) oraz pozytywne rozliczenie projektu zespołowego.

Egzamin praktyczny (Termin I / Termin II) polega na samodzielnym napisaniu aplikacji konsolowej w języku C# w czasie 90 minut, z wymaganiem umieszczenia każdej klasy w oddzielnym pliku .cs. Egzamin weryfikuje indywidualne umiejętności programowania obiektowego.

Ocena końcowa z przedmiotu uwzględnia wynik egzaminu praktycznego oraz rozliczenie projektu zespołowego.

Projekt stanowi warunek konieczny uzyskania oceny końcowej. Ocena z projektu może podwyższyć ocenę końcową o pół stopnia (względem oceny z egzaminu), jeżeli projekt spełnia wszystkie kryteria zaliczeniowe i dodatkowo wyróżnia się co najmniej dwoma z następujących elementów: wzorowa struktura klas z wyraźnym podziałem odpowiedzialności, zastosowanie interfejsów w sposób uzasadniony architekturą, kompletna obsługa wyjątków i walidacja, lub wyróżniająca się dokumentacja techniczna.

Kryteria oceniania egzaminu praktycznego:

- Poprawna realizacja klas z polami, właściwościami, konstruktorami i metodami oraz zastosowanie dziedziczenia: ocena dostateczna (3.0)
- Poprawna realizacja klas z dziedziczeniem oraz zastosowanie kolekcji obiektowych (List<T>) i typów wyliczeniowych (enum): ocena dostateczna plus (3.5)
- Poprawna realizacja systemu wieloklasowego z dziedziczeniem, polimorfizmem i kolekcjami obiektowymi: ocena dobra (4.0)
- Poprawna realizacja systemu wieloklasowego z dziedziczeniem, polimorfizmem, kolekcjami oraz zastosowanie co najmniej jednego interfejsu lub klasy abstrakcyjnej w sposób uzasadniony strukturą rozwiązania: ocena dobra plus (4.5)
- Poprawna realizacja kompletnego systemu obiektowego z zastosowaniem dziedziczenia, polimorfizmu, interfejsów/klas abstrakcyjnych, kolekcji, enum i walidacji danych: ocena bardzo dobra (5.0)

Warunki zaliczenia projektu zespołowego:

Projekt realizowany w zespołach 2-4 osobowych, oceniany na podstawie:

- Działający program spełniający wymagania funkcjonalne
- Zastosowanie programowania obiektowego (hierarchia klas z dziedziczeniem, interfejsy)
- Jakość i czytelność kodu (nazewnictwo, podział odpowiedzialności klas, enkapsulacja)
- Walidacja danych i obsługa błędów
- Dokumentacja techniczna projektu
- Indywidualna obrona: każdy członek zespołu wskazuje własny fragment kodu, uzasadnia decyzje projektowe i odpowiada na pytania prowadzącego

Zasady uczciwości akademickiej: Egzamin praktyczny odbywa się samodzielnie pod nadzorem prowadzącego. Wkład każdego studenta w projekt zespołowy jest weryfikowany indywidualnie podczas obrony. Kopiowanie rozwiązań skutkuje niezaliczeniem.

Próg osiągnięcia efektów uczenia się: Poszczególne efekty uczenia się uznaje się za osiągnięte na podstawie spełnienia kryteriów przypisanych do danego PEU w matrycy weryfikacji. Efekty wiedzy i umiejętności weryfikowane egzaminem praktycznym (PEU_W01-W04, PEU_U01-U05) wymagają uzyskania co najmniej oceny dostatecznej (3.0). Efekty realizowane poprzez projekt (PEU_W05, PEU_U06, PEU_U07) są uznawane za osiągnięte po pozytywnym rozliczeniu projektu i indywidualnej obronie. Kompetencje społeczne (PEU_K01-K02) uznaje się za osiągnięte po spełnieniu warunków określonych dla przypisanych im metod weryfikacji.

Weryfikacja kompetencji społecznych:

- PEU_K01 (krytyczna ocena kodu): weryfikowana przez iteracyjne poprawianie rozwiązań zadań obiektowych na platformie w trakcie semestru oraz przez indywidualną obronę projektu (uzasadnienie decyzji, wskazanie słabości rozwiązania).
- PEU_K02 (poszukiwanie rozwiązań, współpraca): weryfikowana przez samodzielną realizację zadań z wykorzystaniem materiałów, konsultacje z prowadzącym i współpracownikami w zespole, oraz przez udokumentowany wkład w projekt zespołowy.

Technologie sieciowe

Nazwa zajęć po angielsku: Network Technologies

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: I

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Bogusław Kowalski

Opis zajęć:

W ramach przedmiotu student poznaje podstawowe pojęcia, mechanizmy i technologie wykorzystywane w projektowaniu, konfiguracji, diagnostyce i zabezpieczaniu sieci komputerowych. Omawiane są modele komunikacji sieciowej, urządzenia sieciowe, topologie LAN, adresacja IP, podstawowe usługi i protokoły sieciowe, komunikacja przewodowa i bezprzewodowa, routing, przełączanie, diagnostyka łączności oraz zasady bezpiecznego funkcjonowania sieci lokalnych i dostępu do Internetu.

Zajęcia mają charakter teoretyczno-praktyczny. Celem zajęć jest uzyskanie przez studentów: wiedzy na temat zasad funkcjonowania i projektowania sieci komputerowych LAN i WAN wraz ze standardami i protokołami komunikacyjnymi stosowanymi w tych sieciach; umiejętności pracy z dokumentacją techniczną, materiałami elektronicznymi oraz środowiskami laboratoryjnymi pozwalającymi projektować, konfigurować i testować sieci komputerowe.

Szczególny nacisk położony jest na praktyczne zastosowanie technologii sieciowych w pracy informatyka oraz na odpowiedzialne i bezpieczne korzystanie z zasobów sieciowych.

Program przedmiotu bazuje treściach obejmujących zagadnienia wymagane w procesie przygotowania do certyfikacji Cisco 200-301 CCNA oraz materiałach CISCO Networking Academy (Networking Basics oraz CCNA: Introduction to Networks).

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Student wyjaśnia zastosowania technologii sieciowych w systemach informatycznych, usługach internetowych i środowiskach rozproszonych.	K_W03
Student charakteryzuje procesy zachodzące w cyklu życia urządzeń sieciowych – od instalacji, przez konfigurację, aż po użycie	K_W04
Student identyfikuje metody i narzędzia stosowane do konfigurowania, monitorowania i diagnozowania sieci komputerowych	K_W05
Student rozpoznaje standardy i dokumenty techniczne stosowane w sieciach, w tym IEEE, RFC, modele ISO/OSI i TCP/IP oraz zasady adresacji IP.	K_W06
Student proponuje zastosowanie rozwiązań sieciowych wspierających rozwój indywidualnej działalności gospodarczej z wykorzystaniem usług chmurowych	K_W10
Umiejętności (student potrafi)	
Student pozyskuje i wykorzystuje dokumentację techniczną do rozwiązywania problemów związanych z projektowaniem i optymalizacją infrastruktury sieciowej	K_U01
Student selekcjonuje dokumentację techniczną i źródła online w celu efektywnego rozwiązywania problemów konfiguracji sieci	K_U03

Student dobiera narzędzia diagnostyczne i komunikacyjne do typowych zadań administracji siecią (np. analiza logów systemowych w celu oceny przyczyn awarii i formułowania rekomendacji naprawczych)	K_U04
Student projektuje adresację IPv4/IPv6 i prostą topologię sieci LAN spełniającą zadane wymagania funkcjonalne	K_U05
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny własnej wiedzy i umiejętności z zakresu konfiguracji i diagnozowania sieci oraz identyfikuje obszary wymagające doskonalenia	K_K01
Student jest gotów do zasięgania opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemu sieciowego (budowa własnej sieci LAN, analiza funkcjonowania sieci LAN)	K_K02
Student jest gotów do inicjowania działań związanych z wdrażaniem rozwiązań sieciowych wspierających dostępność usług publicznych i edukacyjnych	K_K03
Student jest gotów przestrzegać zasad etyki zawodowej i norm bezpieczeństwa przy realizacji projektów sieciowych	K_K05

Metody i formy kształcenia:

- Wykład problemowy
- Burza mózgów
- Metoda ćwiczeń
- Studium przypadku

Metody oceniania:

- Praca pisemna
- Częstkowe sprawdziany pisemne - kolokwia
- Przygotowanie do zajęć - wejściówki
- Sprawozdanie z realizacji zadania

Wykład konwersatoryjny (W)

1. Wprowadzenie do technologii sieciowych Ewolucja sieci komputerowych: od ARPANET do Internetu Rzeczy (IoT) Typy i klasyfikacje sieci (LAN, WAN, WLAN, MAN) Rola standardów i organizacji normalizacyjnych (IEEE, IETF, ISO)
2. Adresacja i routowanie w sieciach IP Klasy adresów IPv4 i IPv6 Maski podsieci i VLSM Mechanizmy routingu (statyczny vs dynamiczny) Ćwiczenie konwersatoryjne: analiza tras w tracert i ip route
3. Model ISO/OSI i TCP/IP – porównanie i znaczenie warstw Funkcje warstw modelu OSI Praktyczne odwzorowanie protokołów na model TCP/IP Dyskusja: czy model OSI ma dziś znaczenie praktyczne?
4. Protokoły komunikacyjne w warstwach sieciowej i transportowej TCP, UDP – różnice, zastosowania, nagłówki ICMP, ARP, DHCP – rola pomocnicza Analiza nagłówków pakietów – studium przypadku
5. Sprzęt sieciowy i topologie Switch, router, access point, firewall – funkcje i różnice Topologie fizyczne i logiczne Porównanie wydajności topologii w małej firmie (dyskusja z analizą przypadku)
6. Protokoły warstwy aplikacji HTTP/S, FTP, DNS, SMTP, POP3, IMAP – zasady działania Rozszerzenia protokołów, np. DNSSEC, HTTPS Analiza sesji HTTP za pomocą Wireshark – konwersatorium
7. Bezpieczeństwo sieci komputerowych Zagrożenia: sniffing, spoofing, DoS/DDoS, MITM Mechanizmy ochrony: szyfrowanie, VPN, firewall, ACL Rozmowa kierowana: czy bezpieczeństwo to tylko kwestia sprzętu?

Ćwiczenia (Ć)

1. Rozwijanie praktycznych umiejętności konfiguracji usług sieciowych i protokołów komunikacyjnych, analiza ruchu sieciowego, rozwiązywanie zadań problemowych
2. Zasady projektowania sieci IP – struktura adresowa sieci IP
3. Analiza działania sieci LAN na bazie dostępnych narzędzi systemowych
4. Zapoznanie się z budową sieci LAN (Local Area Network) na przykładzie sieci Uczelni - składniki sieci LAN. Wizyta w węzłach sieci uczelnianej
5. Elementy pasywne sieci komputerowej – światłowody multimodowe – pomiary okablowania światłowodowego. Badanie tłumienności
6. Podstawy i zasada funkcjonowania routerów na przykładzie Cisco
7. Przełączniki - elementy funkcjonowania warstwy łącza danych sieci LAN
8. Bezpieczeństwo w sieciach komputerowych – systemy otwarte. Badania i testy bezpieczeństwa systemów sieciowych

Literatura podstawowa

1. Free Computer Networking Course by Cisco: Networking Basics - <https://www.netacad.com/courses/networking-basics?courseLang=en-US> (wybrane działy)
2. CCNA: Introduction to Networks - <https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US> (wybrane działy)
3. Andrew S. Tanenbaum, David J. Wetherall, Sieci komputerowe
4. Praca zbiorowa, Vademecum teleinformatyka część I Warszawa, IDG 2002-2006

Literatura uzupełniająca

1. Stevens W. R. Biblia TCP/IP
2. K. Krysiak, Sieci komputerowe – Kompendium, Helion, Gliwice 2006
3. Hunt C. TCP/IP - Administracja Sieci
4. Sportack M., Sieci Komputerowe – Księga Eksperta

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	60 godz.....	32 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	20 godz.
• Przygotowanie się do kolokwium.....	10 godz.....	12 godz.
• Przygotowanie się do zajęć.....	10 godz.....	12 godz.
• Przygotowanie się do zaliczenia końcowego.....	5 godz.....	12 godz.
• Samodzielne wykonanie zadań.....	5 godz.....	12 godz.
Suma obciążeń:.....	100 godz.....	100 godz.

Zaliczenie zajęć:

Zaliczenie części wykładowej następuje na podstawie otrzymania przez studenta co najmniej 51 punktów w czasie dwóch kolokwium i końcowego zaliczenia

- Kolokwium nr 1 0 – 20 punktów
- Kolokwium nr 2 0 – 20 punktów
- Zaliczenie 0 – 60 punktów

Zasady oceniania słuchaczy :

0 – 50 punktów =	2.0
51 – 60 punktów =	3.0
61 – 70 punktów =	3.5
71 – 80 punktów =	4.0
81 – 90 punktów =	4.5
91 – 100 punktów =	5.0

Dokładne terminy pierwszego i drugiego kolokwium są ustalane na pierwszym wykładzie. Student, który otrzymał co najmniej 31 punktów (suma z dwóch kolokwium) ma prawo przystąpić do zaliczenia zerowego - ustnego.

W czasie wykładu student może otrzymać punkty za aktywność.

Zaliczenie laboratorium następuje na podstawie pozytywnych ocen otrzymanych przez studenta z wejściówek z każdych ćwiczeń laboratoryjnych oraz po otrzymaniu kompletu sprawozdań z przedmiotowych zajęć laboratoryjnych

Projektowanie sieci komputerowych

Nazwa zajęć po angielsku: Computer Network Design

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: II

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....15 / 8 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Bogusław Kowalski

Opis zajęć:

Celem przedmiotu jest rozwijanie praktycznych umiejętności w zakresie projektowania, konfiguracji i zarządzania sieciami komputerowymi. Studenci nauczą się tworzyć efektywne, bezpieczne i skalowalne rozwiązania sieciowe oraz stosować najlepsze praktyki techniczne i bezpieczeństwa.

W ramach przedmiotu studenci uzyskują wiedzę na temat zaawansowanych elementów dotyczących funkcjonowania sieci komputerowych z uwzględnieniem znajomości protokołów komunikacyjnych oraz omówieniem dostępnych technologii sieciowych dostępnych w sieciach lokalnych i rozległych. Szczególny nacisk położony jest na technologie używane do budowy bezpiecznych sieci teleinformatycznych oraz omówienie trendów i kierunków rozwoju sieci teleinformatycznych.

Zajęcia kierowane są do studentów II semestru kierunku Informatyka (studia I stopnia, profil praktyczny). Od uczestników wymaga się podstawowej znajomości systemów operacyjnych oraz architektury komputerów.

W ramach przedmiotu studenci:

- poznają architekturę i topologie sieci komputerowych,
- nauczą się projektować małe i średnie sieci komputerowe,
- będą rozwijać umiejętności praktycznej konfiguracji urządzeń sieciowych,
- przygotują projekt sieci wraz z dokumentacją,
- nauczą się analizować aspekty bezpieczeństwa i niezawodności sieci.

Przedmiot stanowi kontynuację zajęć „Technologie sieciowe” również w zakresie przygotowania do egzaminu CCNA 200-301, a zarazem stanowi podstawę do dalszych kursów z zakresu administracji systemami i bezpieczeństwa sieci.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Student wyjaśnia zastosowania projektowania sieci komputerowych w systemach informatycznych, usługach internetowych i środowiskach organizacyjnych.	K_W03
Student charakteryzuje procesy funkcjonowania urządzeń, usług i systemów sieciowych oraz ich znaczenie dla eksploatacji infrastruktury teleinformatycznej.	K_W04
Student identyfikuje metody, techniki i narzędzia stosowane w projektowaniu, analizie, symulacji, konfiguracji i monitorowaniu sieci komputerowych.	K_W05
Student rozpoznaje standardy i normy techniczne stosowane w projektowaniu sieci, w tym IEEE 802.x, RFC, modele OSI/TCP-IP, Ethernet, Wi-Fi, VLAN i IPv6	K_W06
Umiejętności (student potrafi)	
Student pozyskuje i wykorzystuje dokumentację techniczną, standardy, dokumentację producentów i wyniki pomiarów do rozwiązywania problemu projektowego.	K_U01
Student dobiera narzędzia do symulacji, monitorowania, testowania i dokumentowania	K_U04

działania projektowanej sieci.	
Student projektuje adresację IPv4/IPv6, VLAN-y, topologię logiczną i fizyczną sieci dla zadanych wymagań użytkowych, technicznych i bezpieczeństwa.	K_U05
Student konfiguruje i weryfikuje wybrane elementy zaprojektowanej sieci w środowisku rzeczywistym, wirtualnym lub symulacyjnym.	K_U07
Student porównuje warianty projektowanego rozwiązania sieciowego z uwzględnieniem podstawowych kryteriów technicznych i ekonomicznych.	K_U09
Student projektuje sieć komputerową zgodnie z zadanymi wymaganiami oraz realizuje i weryfikuje jej wybrane elementy w środowisku rzeczywistym lub symulacyjnym.	K_U11
Student opracowuje dokumentację techniczną projektu sieci, obejmującą diagramy logiczne i fizyczne, plan adresacji, konfiguracje urządzeń, uzasadnienie doboru rozwiązań, mechanizmy bezpieczeństwa oraz scenariusze rozbudowy.	K_U12
Student planuje i organizuje pracę własną lub zespołową przy przygotowaniu projektu sieci komputerowej.	K_U17
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny własnego projektu, konfiguracji, dokumentacji i przyjętych założeń bezpieczeństwa.	K_K01
Student jest gotów do zasięgania opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemu sieciowego.	K_K02
Student jest gotów do samorozwoju w zakresie nowych technologii sieciowych i standardów technicznych z poszanowaniem własności intelektualnej.	U_K07

Metody i formy kształcenia:

- Metoda projektu
- Burza mózgów
- Dyskusja
- Wykład problemowy
- Metoda ćwiczeń

Metody oceniania:

- Rozliczenie projektu
- Aktywność w trakcie zajęć
- Zaliczenie praktyczne

Wykład konwersatoryjny (W)

1. Proces projektowania sieci: wymagania, ograniczenia, interesariusze, założenia projektowe.
2. Projekt hierarchiczny i modułowy: access, distribution, core; segmentacja i odporność.
3. Standardy sieciowe (IEEE 802.x, Ethernet, Wi-Fi, VLAN)
4. Projektowanie topologii fizycznych i logicznych (LAN, WAN, WLAN)
5. Urządzenia sieciowe i ich rola w projekcie (routery, przełączniki, firewalle)
6. Adresacja IP, VLSM i planowanie podsieci.
7. Protokoły routingu (RIP, OSPF, EIGRP, BGP) i ich zastosowanie w projektach
8. Jakość usług (QoS) – zapewnienie wydajności i priorytetyzacja ruchu.
9. Bezpieczeństwo w projektowaniu sieci – uwierzytelnianie, szyfrowanie, kontrola dostępu
10. Dokumentacja projektowa – schematy, opisy funkcjonalne, kosztorysy.

Ćwiczenia (Ć)

1. Rozwijanie umiejętności w zakresie konfiguracji i projektowania elementów sieci, rozwiązywanie problemów w grupach, ćwiczenia praktyczne przy komputerze
2. Symulacja działania sieci z QoS i analiza wydajności.
3. Projekt adresacji IP z podziałem na podsieci i VLAN-y.
4. Projekt i implementacja sieci bezprzewodowej z zabezpieczeniami
5. Analiza ruchu sieciowego przy pomocy narzędzi monitorujących (Wireshark).

Projekt (P)

1. Opracowanie projektu sieci komputerowej, przygotowanie dokumentacji technicznej i prezentacja rezultatów
2. Opracowanie dokumentacji technicznej dla małej sieci firmowej.
3. Studium przypadku – analiza istniejącej sieci i propozycja jej modernizacji.
4. Projekt zespołowy: zaprojektowanie i obrona koncepcji sieci komputerowej dla wybranej organizacji.

Literatura podstawowa

1. James F. Kurose, Keith W. Ross, Sieci komputerowe. Ujęcie całościowe. Helion 2023
2. Andrew S. Tanenbaum, David J. Wetherall, Sieci komputerowe
3. CCNA: Introduction to Networks - <https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US> (wybrane tematy)

Literatura uzupełniająca

1. Cisco Campus LAN and Wireless LAN Solution Design Guide - projektowanie hierarchiczne i WLAN - <https://www.cisco.com/c/en/us/td/docs/solutions/CVD/Campus/cisco-campus-lan-wlan-design-guide.html>
2. Barrie Sosinsky, „Sieci komputerowe. Biblia”
3. Sportack M., Sieci Komputerowe – Księga Eksperta
4. Wendell Odom, CCNA 200-301 Official Cert Guide
5. A. W. Butrimenko, „Projektowanie i eksploatacja sieci komputerowych”

Źródła dodatkowe

1. IT Professional – miesięcznik

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Przygotowanie projektu.....	20 godz.....	20 godz.
• Samodzielne wykonanie zadań.....	20 godz.....	41 godz.
• Zapoznanie się z literaturą przedmiotu.....	15 godz.....	15 godz.
Suma obciążeń:.....	100 godz.....	100 godz.

Zaliczenie zajęć:

1. Złożenie kompletnego projektu zaliczeniowego, zgodnego z wytycznymi przedstawionymi przez prowadzącego, obejmującego m.in.:
 - a) dokumentację techniczną zaprojektowanej sieci,
 - b) diagramy logiczne i fizyczne,
 - c) konfiguracje urządzeń (realne lub symulowane),
 - d) analizę zastosowanych rozwiązań i ich uzasadnienie,
 - e) ocenę bezpieczeństwa oraz zgodności ze standardami.
2. Pozytywna ocena z obrony projektu, podczas której student:
 - a) prezentuje główne założenia i funkcjonalność projektu,
 - b) odpowiada na pytania problemowe związane z zastosowaną topologią, doborem protokołów i urządzeń,
 - c) uzasadnia dokonane wybory technologiczne oraz wskazuje możliwe scenariusze rozbudowy i optymalizacji sieci.

Dodatkowo:

Brana pod uwagę jest aktywność w trakcie zajęć projektowych i wykładów konwersatoryjnych, warunkiem dopuszczenia do obrony projektu jest terminowe oddanie pracy oraz spełnienie minimalnych kryteriów formalnych i merytorycznych.

Administracja sieci komputerowych

Nazwa zajęć po angielsku: Administration of computer networks

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: III

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Wojciech Korzybski

Opis zajęć:

W ramach przedmiotu omawiane są podstawowe zagadnienia z dziedziny administrowania komputerową siecią lokalną. Student poznaje zasady administrowania siecią komputerową opartą na środowisku Windows Server. Omawiane są usługi katalogowe, role serwera, zasady grupy, wirtualizacja, przechowywanie danych, mechanizmy optymalizacji i konserwacji systemu, aktualizowanie serwera oraz rozwiązywanie typowych problemów administracyjnych. Szczególny nacisk położony jest na rozumienie roli usług takich jak Active Directory, DNS i DHCP w organizacji, zarządzaniu i utrzymaniu infrastruktury sieciowej.

W części praktycznej studenci wykonują przykładowe zadania administratora sieciowego systemu Windows. Student konfiguruje wybrane usługi Windows Server, dodaje role serwera, zarządza uprawnieniami NTFS, stosuje przykładowe zasady grupy, wykorzystuje mechanizmy przechowywania danych, monitoruje wydajność oraz analizuje dzienniki zdarzeń.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia zagadnienia z dziedziny funkcjonowania sieci komputerowych i ich administrowania	K_W04
identyfikuje narzędzia i techniki administrowania środowiskiem Windows Server	K_W05
rozpoznaje standardy, dobre praktyki i zasady bezpieczeństwa stosowane w administracji domeną, kontami, zasobami, logami i dostępem zdalnym	K_W06
Umiejętności (student potrafi)	
dobiera narzędzia administracyjne do realizacji typowych zadań administratora sieci Windows	K_U04
konfiguruje konta użytkowników, grupy; uprawnienia NTFS	K_U05
stosuje standardy i dobre praktyki administracji sieciami komputerowymi	K_U12
stosuje rozwiązania i dobre praktyki poznane w środowisku zawodowym/uczelnianym podczas diagnozowania i utrzymania infrastruktury sieciowej	K_U13
Kompetencje społeczne (student jest gotów do)	
jest gotów do przestrzegania zasad etyki zawodowej, poufności danych, ograniczania uprawnień i odpowiedzialnego korzystania z kont administracyjnych	K_K05
jest gotów do samorozwoju w zakresie nowych technologii administracji systemami, tożsamością i usługami sieciowymi	U_K07

Metody i formy kształcenia:

Metody oceniania:

- Wykład informacyjny
- Metoda ćwiczeń
- Dyskusja
- Aktywność w trakcie zajęć
- Zaliczenie testowe
- Zaliczenie praktyczne

Wykład konwersatoryjny (W)

1. Wprowadzenie do problematyki administrowania siecią LAN
2. Przegląd architektury systemu Windows Server
3. Narzędzia administracyjne w Windows Server
4. Zarządzanie stacjami roboczymi i Serverami Windows
5. Usługi katalogowe - Active Directory
6. Zarządzanie kontami użytkowników i grup
7. Systemy plików w Windows Server i zarządzanie zasobami
8. Udostępnianie danych i zabezpieczenia zasobów

Ćwiczenia (Ć)

1. Udostępnianie folderów i uprawnienia do nich
2. Zarządzanie kontami i grupami w systemie Windows
3. Uprawnienia zabezpieczeń lokalnych NTFS
4. Polecenia sieciowe systemu operacyjnego
5. Skrypty logowania i ich wykorzystanie
6. Zdalny dostęp w Windows Server
7. Narzędzia administracyjne systemu Windows Server
8. Prezentacja procesu instalowania sieci lokalnej

Literatura podstawowa

1. Dokumentacja systemu Windows Server - <https://learn.microsoft.com/pl-pl/windows-server/>
2. Mrosek B., Chrobok K., Administrowanie sieciowymi systemami operacyjnymi Windows Server i Linux Server dla każdego Technika i studenta IT. ITSTART 2022

Literatura uzupełniająca

1. Krause J., Windows Server 2022 dla profesjonalistów. Profesjonalna administracja środowiskiem Windows Server. Wydanie IV. Helion 2024
2. Wołk K., Biblia Windows Server 2016. Podręcznik Administratora, Helion 2016
3. Stanek W.R.: Vademecum Administratora. Windows Server 2012, Helion 2012

Źródła dodatkowe

1. Windows Server 2022 Step by step installation - <https://www.youtube.com/watch?v=LbDqYtDkqLw>
2. Windows Server 2022 course/training: Learn how to use Windows Server 2022 for administration - <https://www.youtube.com/watch?v=ubDXqG7UXrA>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- | | | |
|---|---------------|-----------|
| • Zajęcia z udziałem nauczyciela akademickiego..... | 60 godz..... | 32 godz. |
| • Zapoznanie się z literaturą przedmiotu..... | 10 godz..... | 15 godz. |
| • Samodzielne wykonanie zadań..... | 10 godz..... | 33 godz. |
| • Przygotowanie się do zaliczenia końcowego..... | 20 godz..... | 20 godz. |
| Suma obciążeń:..... | 100 godz..... | 100 godz. |

Zaliczenie zajęć:

Zaliczenie wykładów odbywa się na podstawie testu komputerowego, który należy zaliczyć na mi. 50%. Zaliczenie ćwiczeń polega na praktycznym wykonaniu zadania zaliczeniowego z zagadnień realizowanych na zajęciach. Brana jest pod uwagę aktywność na zajęciach, weryfikowana poprzez realizowane w trakcie zajęć zadania przesyłane na platformę Wlodek. Ocena końcowa jest średnią arytmetyczną z zaliczenia wykładu i ćwiczeń.

Bezpieczeństwo systemów informatycznych i kryptografia

Nazwa zajęć po angielsku: Cybersecurity and Cryptography

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: IV

Forma zaliczenia: Egzamin

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Agnieszka Polska-Bachurzevska

Opis zajęć:

Przedmiot rozwija umiejętność praktycznego zabezpieczania systemów informatycznych oraz świadomego stosowania kryptografii — od zrozumienia mechanizmów, przez ich samodzielną implementację i konfigurację, po wykrywanie i wykorzystywanie błędów w rozwiązaniach nieprawidłowo zaprojektowanych. Zajęcia mają charakter warsztatowy i wymagający: studenci nie tylko poznają teorię (modele bezpieczeństwa — poufność, integralność, dostępność; uwierzytelnianie i autoryzacja; analiza ryzyka), lecz przede wszystkim pracują na realnych narzędziach i danych, samodzielnie łamiąc słabe zabezpieczenia i budując poprawne.

Studenci ćwiczą myślenie adwersaryjne (jak myśli atakujący) i inżynierskie (jak projektować obronę): analizują szyfry symetryczne i asymetryczne, funkcje skrótu i mechanizmy przechowywania haseł, badają protokół TLS/PKI, modelują zagrożenia (threat modeling), analizują rzeczywiste incydenty oraz podatności aplikacji webowych (OWASP Top 10). Każdy blok kończy się weryfikowalnym zadaniem praktycznym.

Kurs jest celowo wymagający — obowiązkowe laboratoria, częściowe kolokwia oraz semestralny projekt bezpieczeństwa nie pozwalają zaliczyć przedmiotu bez realnego opanowania materiału. Celem jest, aby absolwent potrafił samodzielnie ocenić ryzyko, dobrać i poprawnie wdrożyć mechanizmy zabezpieczające oraz krytycznie zweryfikować cudze rozwiązania, posługując się aktualnymi standardami (NIST, OWASP, MITRE ATT&CK) i dobrymi praktykami inżynierskimi.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Student wyjaśnia praktyczne zastosowania mechanizmów bezpieczeństwa w systemach informatycznych, w tym sposoby ochrony danych, kontroli dostępu oraz zabezpieczania usług IT.	K_W03
Student charakteryzuje podstawowe procesy zachodzące w systemach informatycznych i sieciowych w kontekście bezpieczeństwa, w tym uwierzytelniania, autoryzacji oraz przepływu danych.	K_W04
Student identyfikuje metody, techniki i narzędzia stosowane w analizie bezpieczeństwa systemów informatycznych oraz identyfikuje zagrożenia i podatności.	K_W05
Student rozpoznaje wybrane standardy i dobre praktyki w zakresie bezpieczeństwa informacji oraz stosuje je w systemach informatycznych.	K_W06
Student wyjaśnia społeczne konsekwencje naruszeń bezpieczeństwa informacji, socjotechniki, nadzoru, wycieków danych i błędów użytkowników.	K_W07
Student charakteryzuje podstawowe uwarunkowania prawne i etyczne związane z ochroną danych i bezpieczeństwem informacji.	K_W08
Student charakteryzuje wybrane zagadnienia kryptografii współczesnej, w tym szyfrowanie symetryczne i asymetryczne, funkcje skrótu, podpis cyfrowy, TLS, przechowywanie haseł.	U_W11

<i>Umiejętności (student potrafi)</i>	
Student analizuje incydenty bezpieczeństwa, interpretuje ich przebieg oraz formułuje wnioski dotyczące przyczyn i skutków zdarzeń.	K_U02
Student dobiera i ocenia informacje oraz źródła wiedzy w celu analizy zagrożeń i podejmowania decyzji dotyczących bezpieczeństwa systemów IT.	K_U03
Student dobiera odpowiednie metody i narzędzia do analizy oraz zabezpieczania systemów informatycznych w kontekście konkretnych scenariuszy.	K_U04
Student rozwiązuje praktyczne problemy związane z bezpieczeństwem systemów informatycznych, w tym identyfikować zagrożenia i proponować środki zaradcze.	K_U05
Student stosuje metody analityczne do oceny ryzyka oraz projektuje rozwiązania zwiększające poziom bezpieczeństwa systemów.	K_U07
Student uwzględnia aspekty techniczne, organizacyjne, prawne, etyczne i czynniki ludzkie podczas analizy bezpieczeństwa systemu.	K_U08
Student przeprowadza krytyczną analizę funkcjonowania systemów informatycznych i sieciowych pod kątem bezpieczeństwa oraz ocenia zastosowane rozwiązania.	K_U10
Student stosuje standardy i dobre praktyki bezpieczeństwa przy rozwiązywaniu zadań inżynierskich oraz projektowaniu systemów IT.	K_U12
Student komunikuje wyniki analiz bezpieczeństwa w sposób zrozumiały i uporządkowany, wykorzystując odpowiednią terminologię techniczną.	K_U14
Student uczestniczy w dyskusji dotyczącej rozwiązań z zakresu bezpieczeństwa systemów informatycznych oraz uzasadnia swoje stanowisko.	K_U15
<i>Kompetencje społeczne (student jest gotów do)</i>	
Student jest gotów do krytycznej oceny rozwiązań z zakresu bezpieczeństwa systemów informatycznych oraz identyfikacji ich ograniczeń.	K_K01
Student jest gotów do samodzielnego rozwiązywania problemów z zakresu bezpieczeństwa oraz korzystania z wiedzy eksperckiej w sytuacjach złożonych.	K_K02
Student jest gotów do podejmowania decyzji dotyczących bezpieczeństwa systemów informatycznych z uwzględnieniem ich wpływu na organizację.	K_K04
Student jest gotów do przestrzegania zasad etyki zawodowej w obszarze bezpieczeństwa informacji oraz ochrony danych.	K_K05

Metody i formy kształcenia:

- Wykład informacyjny
- Wykład problemowy
- Dyskusja
- Studium przypadku
- Burza mózgów
- Metoda ćwiczeń
- Metoda projektu
- Metoda laboratoryjna
- Metoda symulacyjna
- Grywalizacja

Metody oceniania:

- Zaliczenie opisowe
- Egzamin opisowy
- Zaliczenie praktyczne
- Częstkowe sprawdziany pisemne - kolokwia
- Przygotowanie do zajęć - wejściówki
- Rozliczenie projektu
- Sprawozdanie z realizacji zadania

Wykład konwersatoryjny (W)

1. Modele i cele bezpieczeństwa (CIA, AAA); myślenie ryzykiem i modelowanie zagrożeń (STRIDE)
2. Kryptografia symetryczna: szyfry blokowe i strumieniowe, tryby pracy (ECB/CBC/GCM) i ich typowe błędy
3. Kryptografia asymetryczna i PKI: RSA/ECC, wymiana kluczy, podpis cyfrowy, certyfikaty
4. Funkcje skrótu i bezpieczne przechowywanie haseł (SHA-2/3, HMAC, bcrypt/scrypt/Argon2, sól i pieprz)
5. Bezpieczeństwo protokołów: TLS 1.3, PKI w praktyce, ataki na kanał komunikacyjny (MITM)

6. Uwierzytelnianie i kontrola dostępu: MFA, OAuth2/OIDC/SSO, model Zero Trust, Conditional Access
7. Utwarczanie systemów i sieci (hardening): segmentacja, logowanie, detekcja
8. Bezpieczeństwo aplikacji webowych: OWASP Top 10, wstrzyknięcia, sesje, secure coding
9. Zarządzanie podatnościami i reagowanie na incydenty (NIST SP 800-61, MITRE ATT&CK)
10. Czynniki ludzkie, aspekty prawne i etyczne: socjotechnika, ochrona danych/RODO, legalność testów

Ćwiczenia (Ć)

1. Lab. kryptografii I: implementacja i łamanie słabych szyfrów; ECB vs CBC/GCM
2. Lab. kryptografii II: haszowanie i łamanie haseł (John/hashcat), dobór parametrów Argon2/bcrypt
3. Analiza TLS/PKI: inspekcja certyfikatów i handshake'u, wykrywanie słabych konfiguracji
4. Threat modeling wybranego systemu: model STRIDE + rejestr ryzyk
5. Uwierzytelnianie i dostęp: konfiguracja MFA/OAuth2, analiza polityk Zero Trust / Conditional Access
6. Bezpieczeństwo aplikacji webowej: wykorzystanie podatności OWASP w środowisku treningowym i ich naprawa
7. Analiza incydentu: rekonstrukcja ataku z logów, mapowanie technik na MITRE ATT&CK
8. Projekt semestralny — cz. 1: analiza zagrożeń i dobór zabezpieczeń zadanego systemu
9. Projekt semestralny — cz. 2: implementacja/konfiguracja, testy i obrona (prezentacja)

Literatura podstawowa

1. W. Stallings, Lawrie Brown, "Computer Security: Principles and Practice", Pearson
2. William Stallings, "Cryptography and Network Security: Principles and Practice", Pearson
3. A. Białas, "Bezpieczeństwo informacji i usług w nowoczesnej instytucji i firmie" Wydawnictwo WNT 2017
4. CERT Polska / NASK – aktualne przykłady incydentów, ostrzeżenia i materiały edukacyjne (<https://cert.pl/>, <https://www.nask.pl/>)

Literatura uzupełniająca

1. Ross Anderson, "Security Engineering: A Guide to Building Dependable Distributed Systems"
2. OWASP Foundation, "OWASP Top 10 – Web Application Security Risks"
3. P. Cofta, "Zarządzanie bezpieczeństwem informacji"
4. MITRE ATT&CK – baza taktyk i technik atakujących, pomocna przy opisie scenariuszy ataku.

Źródła dodatkowe

1. Microsoft Learn – Microsoft Entra Conditional Access.
2. Microsoft Learn – Zero Trust i zabezpieczanie dostępu w Microsoft 365.
3. NIST Cybersecurity Framework 2.0 – zarządzanie ryzykiem cyberbezpieczeństwa.
4. NIST SP 800-61 Rev. 3 – rekomendacje dotyczące reagowania na incydenty.
5. OWASP Top 10:2025 – najważniejsze ryzyka bezpieczeństwa aplikacji webowych.
6. RFC 9106 – Argon2 jako funkcja do bezpiecznego haszowania haseł.
7. ENISA – materiały dotyczące cyberhigieny i świadomości bezpieczeństwa.

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	60 godz.....	32 godz.
• Zapoznanie się z literaturą przedmiotu.....	6 godz.....	12 godz.
• Samodzielne wykonanie zadań.....	12 godz.....	24 godz.
• Przygotowanie i udział w egzaminie.....	6 godz.....	8 godz.
• Przygotowanie projektu.....	12 godz.....	16 godz.
• Przygotowanie się do kolokwium.....	4 godz.....	8 godz.
Suma obciążeń:.....	100 godz.....	100 godz.

Zaliczenie zajęć:

Zaliczenie przedmiotu wymaga łącznego spełnienia wszystkich poniższych warunków — nie jest możliwe zaliczenie bez opanowania części praktycznej.

1. Laboratoria: obecność obowiązkowa (dopuszczalna 1 nieobecność, każdą kolejną należy odrobić zadaniem uzupełniającym). Przed wybranymi laboratoriami odbywają się krótkie wejściówki sprawdzające przygotowanie. Z każdego laboratorium student oddaje sprawozdanie/rozwiązanie podlegające ocenie. Warunek: oddanie i zaliczenie

wszystkich sprawozdań oraz uzyskanie co najmniej 50% punktów z wejściówek.

2. Kolokwia: w trakcie semestru odbywają się 2 kolokwia teoretyczno-praktyczne. Warunek: średnia $\geq 50\%$ (dopuszczalne jedno podejście poprawkowe).

3. Projekt semestralny: analiza zagrożeń, dobór i wdrożenie zabezpieczeń oraz testy; oddanie dokumentacji i obrona (prezentacja). Warunek: ocena pozytywna ($\geq 50\%$). Projekt jest obowiązkowy i nie podlega pełnej kompensacji innymi elementami.

4. Egzamin pisemny: warunkiem dopuszczenia jest zaliczenie laboratoriów, kolokwiów i projektu. Egzamin ma formę problemową (pytania otwarte + analiza scenariuszy incydentów). Warunek: $\geq 50\%$.

Ocena końcowa jest średnią ważoną: egzamin 35%, projekt 30%, kolokwia 20%, laboratoria (sprawozdania + wejściówki) 15%. Aktywność merytoryczna na zajęciach może podnieść ocenę końcową o maksymalnie 0,5 stopnia.

Skala ocen (procent $\rightarrow$ ocena): 0–49% — 2,0; 50–60% — 3,0; 61–70% — 3,5; 71–80% — 4,0; 81–90% — 4,5; 91–100% — 5,0. Warunkiem oceny pozytywnej jest uzyskanie $\geq 50\%$ w każdym z elementów obowiązkowych (laboratoria, kolokwia, projekt, egzamin).

Cyfrowa dokumentacja techniczna

Nazwa zajęć po angielsku: Digital technical documentation

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: II

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 2

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Wojciech Korzybski

Opis zajęć:

W treściach przedmiotu studenci są zapoznawani z wieloma metodami i narzędziami do tworzenia dokumentacji cyfrowej, przydatnymi w pracy zawodowej informatyka. W ramach zajęć ćwiczeniowych studenci poznają sposób budowy schematów blokowych, diagramów sieci i planów pomieszczeń.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
charakteryzuje metody, techniki i narzędzia tworzenia cyfrowej dokumentacji technicznej, w tym schematów blokowych, diagramów sieci, diagramów UML/BPMN, diagramów architektury i dokumentacji projektowej	K_W05
rozpoznaje wybrane standardy, notacje i konwencje dokumentowania systemów informatycznych, sieci, procesów, interfejsów API i infrastruktury technicznej	K_W06
Umiejętności (student potrafi)	
dobiera źródła, wzorce, dokumentację narzędziową i informacje techniczne potrzebne do przygotowania dokumentacji cyfrowej	K_U03
dobiera właściwe narzędzia informatyczne do opracowania dokumentacji graficznej, tekstowej i projektowej	K_U04
tworzy dokumentację techniczną z użyciem poznanych narzędzi i rozwiązuje praktyczne problemy dokumentowania systemów informatycznych	K_U05
projektuje strukturę dokumentacji systemu informatycznego lub sieci teleinformatycznej, używając właściwych metod, technik i narzędzi	K_U11
Kompetencje społeczne (student jest gotów do)	
jest gotów krytycznie oceniać jakość, kompletność, aktualność i jednoznaczność dokumentacji technicznej	K_K01
jest gotów wykorzystywać dokumentację techniczną jako narzędzie organizacji pracy, komunikacji z interesariuszem i rozwoju działań projektowych lub przedsiębiorczych	K_K04
jest gotów przestrzegać zasad etyki zawodowej, prawa autorskiego i legalnego wykorzystania szablonów, ikon, symboli oraz materiałów źródłowych w dokumentacji	K_K05

Metody i formy kształcenia:

- Metoda ćwiczeń

Ćwiczenia (Ć)

Metody oceniania:

- Zaliczenie praktyczne

1. Rola dokumentacji technicznej w pracy informatyka
2. Wprowadzenie: Zasady dobrej dokumentacji: cel, odbiorca, zakres, wersjonowanie, aktualność, jednoznaczność
3. Zapoznanie z oprogramowaniem do tworzenia dokumentacji
4. Tworzenie schematów blokowych algorytmów
5. Diagramy sieci: topologia logiczna i fizyczna
6. Schemat sieci w budynku: punkty dystrybucyjne, okablowanie, urządzenia aktywne
7. Elementy szafy serwerowej i stojaków sieciowych
8. Opisy i bazy danych w graficznej dokumentacji
9. Projekt zaliczeniowy

Literatura podstawowa

1. Doug Lowe: Sieci komputerowe dla bystrzaków, HELION 2019

Literatura uzupełniająca

1. Andrei Besedin: How to Learn Microsoft Visio Quickly!, PublishDrive, 2018
2. Materiały na platformie WLODEK

Źródła dodatkowe

1. https://www.google.pl/books/edition/Microsoft_Visio_2016_Step_By_Step/kPA0CwAAQBAJ?hl=pl&gbpv=1&dq=visio&printsec=frontcover

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....30 godz.....16 godz.
- Zapoznanie się z literaturą przedmiotu.....5 godz.....16 godz.
- Przygotowanie się do zaliczenia końcowego.....5 godz.....8 godz.
- Przygotowanie projektu.....10 godz.....10 godz.
- Suma obciążeń:.....50 godz.....50 godz.

Zaliczenie zajęć:

Zaliczenie przedmiotu następuje na podstawie samodzielnego projektu przygotowanego przez studenta. Przy ocenie końcowej brana jest także pod uwagę obecność i aktywność studentów na zajęciach.

Bazy danych

Nazwa zajęć po angielsku: Databases

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: II

Forma zaliczenia: Egzamin

Punkty ECTS: 6

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Andrzej Ulicki

Opis zajęć:

Celem przedmiotu jest zaznajomienie studentów z relacyjnym typem baz danych. W wykładzie zawarte będą szczegółowe funkcjonalności baz danych w oparciu o silnik Microsoft SQL Server. Na ćwiczeniach studenci będą mieli okazję do praktycznej realizacji zadań w oparciu o bazy MS SQL z wykorzystaniem środowiska graficznego i konsolowego. Dodatkowym elementem będzie zadanie praktyczne realizowane przez każdego ze studentów.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia zastosowanie relacyjnych i nierelacyjnych baz danych w systemach informatycznych oraz rolę danych w aplikacjach	K_W03
charakteryzuje procesy działania systemu bazodanowego: przechowywanie danych, transakcje, współbieżność, integralność, bezpieczeństwo i dostęp klient-serwer	K_W04
identyfikuje metody, techniki i narzędzia projektowania, implementacji, zapytań, administracji i dokumentowania baz danych	K_W05
rozpoznaje standardy i dobre praktyki dotyczące SQL, modelowania danych, normalizacji, integralności, uprawnień i bezpieczeństwa baz danych	K_W06
<i>Umiejętności (student potrafi)</i>	
pozyskuje i wykorzystuje dokumentację systemu bazodanowego, SQL i narzędzi projektowych do rozwiązania problemu bazodanowego	K_U01
opracowuje strukturę relacyjnej bazy danych wykorzystując diagramy ERD i reguły normalizacyjne	K_U02
stosuje język SQL do tworzenia struktur danych, wykonywania zapytań, złączeń, agregacji, transakcji i modyfikacji danych; wykorzystuje narzędzia wspierające pracę programistyczną typu CASE	K_U04
projektuje strukturę relacyjnej bazy danych z użyciem ERD, reguł integralności i normalizacji	K_U05
opracowuje bazę danych i jej dokumentację techniczną	K_U11
stosuje standardy i dobre praktyki bezpieczeństwa, dokumentowania, testowania oraz utrzymania projektu bazodanowego	K_U12
prezentuje projekt bazodanowy, uzasadnia decyzje projektowe i komunikuje ograniczenia	K_U14

rozwiązania z użyciem terminologii baz danych	
Kompetencje społeczne (student jest gotów do)	
jest gotów do krytycznej oceny poprawności modelu danych, zapytań, dokumentacji i bezpieczeństwa własnego projektu	K_K01
jest gotów do działania przedsiębiorczego przy ocenie możliwości wykorzystania projektu bazodanowego w rozwiązaniu użytkowym lub komercyjnym	K_K04

Metody i formy kształcenia:

- Wykład problemowy
- Dyskusja
- Studium przypadku
- Metoda ćwiczeń
- Metoda projektu

Metody oceniania:

- Rozliczenie projektu
- Egzamin testowy
- Zaliczenie praktyczne

Wykład konwersatoryjny (W)

1. Podstawowa terminologia stosowana w bazach danych: dane, informacja, system baz danych, DBMS.
2. Model relacyjny: relacja, klucz, zależności, ograniczenia integralności.
3. Modelowanie diagramów ERD.
4. Tworzenie struktur danych w relacyjnej bazie danych z wykorzystaniem języka SQL.
5. Definicja modelu relacyjnego.
6. Proces normalizacji w bazach danych.
7. Wprowadzenie do języka SQL. Podstawowe operacje na danych w języku SQL.
8. Techniki złączeń i programowanie proceduralne w języku SQL.
9. Technologia klient – serwer w dostępie do bazy danych.
10. Transakcja i jej własności.
11. Zagadnienia bezpieczeństwa w bazach danych.
12. Organizacja plików w bazach danych.
13. Nierelacyjne bazy danych (NoSQL).

Ćwiczenia (Ć)

1. Projektowanie diagramów DFD i ERD.
2. Tworzenie struktur danych w relacyjnej bazie danych z wykorzystaniem języka SQL.
3. Struktury jeden do jeden, jeden do wiele oraz wiele do wiele w relacyjnych bazach danych.
4. Narzędzia CASE do tworzenia struktur danych w relacyjnych bazach danych.
5. Programowanie w języku SQL i języku wbudowanym w bazie danych.
6. Język SQL, tworzenie statycznych i dynamicznych zapytań SQL.

Projekt (P)

1. Projekt - opis problemu i wymagań użytkownika.
2. Opracowanie diagramu przepływu danych (DFD), zdefiniowanie encji (obiekty) i opracowanie diagram encji (ERD).
3. Opracowanie logicznej struktury tabel, przeprowadzenie analizy zależności funkcyjnych i normalizacji tabel.
4. Opracowanie dokumentacji opisującą proces tworzenia aplikacji bazodanowej.
5. Opracowanie dokumentacji użytkownika oraz technicznej umożliwiającej rozbudowę opracowanego projektu.

Literatura podstawowa

1. Ramez Elmasri, Shamkant B. Navathe, Wprowadzenie do systemów baz danych. Wydanie VII, Helion 2019
2. Rojek-Mikołajczak Izabela, Bazy danych kurs podstawowy dla inżynierów informatyków. WAB 2004
3. Microsoft Learn Podstawy bazy danych - <https://learn.microsoft.com/pl-pl/shows/dbfundamentals/>
4. Dokumentacja techniczna SQL Server - <https://learn.microsoft.com/pl-pl/sql/sql-server/?view=sql-server-ver16>

Literatura uzupełniająca

1. Larry Rockoff, Język SQL. Przyjazny podręcznik. Wydanie II, Helion 2017
2. Jeffrey D. Ullman, Jennifer Widom, Podstawowy wykład z systemów baz danych, WNT 2000

Źródła dodatkowe

1. <https://www.w3schools.com/sql/>
2. <https://www.postgresql.org/docs/>
3. <https://www.mongodb.com/>
4. <https://firebase.google.com/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	20 godz.
• Przygotowanie się do zajęć.....	20 godz.....	25 godz.
• Wypełnienie dokumentacji zajęć.....	15 godz.....	25 godz.
• Przygotowanie projektu.....	20 godz.....	25 godz.
• Przygotowanie się do zaliczenia końcowego.....	10 godz.....	15 godz.
Suma obciążeń:.....	150 godz.....	150 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie zaliczenia ze wszystkich trzech form prowadzenia zajęć: co najmniej 50% punktów z egzaminu testowego z wykładu, co najmniej 50% punktów ze sprawozdań z realizowanych ćwiczeń i co najmniej 50% punktów z projektu ocenianego na podstawie złożonej i zaprezentowanej bazy danych wraz z dokumentacją. Ocena końcowa wyliczana jest na podstawie znormalizowanych liczb punktów z poszczególnych form zajęć z odpowiadającymi im wagami.

Inżynieria oprogramowania

Nazwa zajęć po angielsku: Software Engineering

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: III

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Mariusz Sochacki

Opis zajęć:

Inżynieria oprogramowania to dziedzina informatyki zajmująca się procesem tworzenia, rozwoju i zarządzania systemami oprogramowania. Celem przedmiotu jest przedstawienie studentom metodologii, technik i narzędzi niezbędnych do skutecznego projektowania, implementacji, testowania i utrzymywania systemów oprogramowania. Przedmiot "Inżynieria Oprogramowania" przeznaczony jest dla studentów kierunku Informatyka, którzy przynajmniej w podstawowym zakresie znają metody wytwarzania oprogramowania. Celem przedmiotu jest poznanie nowoczesnych metod wytwarzania oprogramowania, zgodnie ze standardami określonymi przez międzynarodowe organizacje zajmujące się dziedziną inżynierii oprogramowania. Kurs jest nastawiony na poznanie metod tworzenia dokumentacji od etapu analizy do rozwoju oprogramowania. Po ukończeniu kursu studenci zapoznają się ze znajomością wykorzystania MS Visio w zakresie budowy diagramów UML, wykorzystania Enterprise Architect w dokumentowaniu i wytwarzaniu oprogramowania. Ponadto studenci zapoznają się z wykorzystaniem wzorców projektowych w wytwarzaniu oprogramowania, budowy systemów rozproszonych oraz metodologiami prowadzenia projektów informatycznym. Umożliwi to studentom, uczestniczenie w projektach informatycznych mających na celu wytwarzanie oprogramowania, przygotowanie do funkcji analityka systemów informatycznych.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia zastosowania inżynierii oprogramowania w analizie, projektowaniu, implementacji, testowaniu, wdrażaniu i utrzymaniu systemów informatycznych	K_W03
charakteryzuje procesy zachodzące w cyklu życia oprogramowania, w tym analizę wymagań, projektowanie, implementację, testowanie, wdrażanie, utrzymanie i rozwój	K_W04
identyfikuje metody, techniki i narzędzia inżynierii oprogramowania, w tym UML, repozytoria kodu, narzędzia CASE, systemy zarządzania zadaniami	K_W05
rozpoznaje standardy i notacje stosowane w inżynierii oprogramowania	K_W06
wyjaśnia ekonomiczne, prawne i etyczne uwarunkowania wytwarzania oprogramowania, w tym odpowiedzialność za jakość, bezpieczeństwo, dane i decyzje projektowe	K_W08
<i>Umiejętności (student potrafi)</i>	
pozyskuje i wykorzystuje dokumentację standardów, bibliotek, narzędzi i API z portali międzynarodowych organizacji stanowiących standardy notacji UML (OMG) oraz innych źródeł	K_U01

integruje informacje z wymagań, dokumentacji, modeli, testów i przeglądów oraz uzasadnia decyzje projektowe	K_U02
dobiera narzędzia do realizacji typowych zadań dla inżynierii oprogramowania - prowadzenia projektów wdrożeniowych, utrzymaniowych i rozwojowych	K_U04
ocenia system informatyczny, szczególnie w zakresie wykorzystania odpowiednich technik analitycznych, podejścia do fazy testów, prac programistycznych, bezpieczeństwa i utrzymania	K_U10
projektuje systemy oraz określa zasady wytwarzania oprogramowania poprzez właściwe wykorzystanie wzorców oraz standardów analizy i wytwarzania oprogramowania	K_U11
stosuje standardy, wzorce, praktyki dokumentowania, testowania, przeglądu i utrzymania oprogramowania w zadaniu inżynierskim	K_U12
opracowuje dokumentację techniczną i komunikuje decyzje projektowe z użyciem specjalistycznej terminologii inżynierii oprogramowania	K_U14
Kompetencje społeczne (student jest gotów do)	
jest gotów do rozwiązywania problemów praktycznych, korzystając samodzielnie z dostępnych materiałów lub korzystając z konsultacji wykładowcy oraz przygotowuje się do właściwej realizacji zadań podczas testów	K_K02
jest gotów do dzielenia się swoją wiedzą i umiejętnościami w zespole projektowym oraz w środowisku studenckim	K_K03
jest gotów do przestrzegania zasad etyki zawodowej, odpowiedzialnego korzystania z oprogramowania, licencji, wzorców, kodu	K_K05
jest gotowy do wykorzystania nowoczesnych technologii informacyjnych do ciągłego zdobywania informacji na tematy zawodowe w związku z błyskawiczną zmianą wykorzystywanych w informatyce technologii oraz oprogramowania	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Praca ze źródłem elektronicznym
- Metoda sytuacyjna
- Burza mózgów
- Metoda ćwiczeń
- Metoda symulacyjna
- Wykład problemowy

Metody oceniania:

- Zaliczenie testowe
- Zaliczenie praktyczne
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

1. Wprowadzenie do Inżynierii Oprogramowania. Budowa wymagań.
2. Wykorzystanie języka UML - modele behawioralne
3. Wykorzystanie języka UML - modele strukturalne
4. Wykorzystanie wzorców projektowych
5. Walidacja i testowanie oprogramowania
6. Wykorzystanie API
7. Wprowadzenie do zarządzania projektem informatycznym
8. Rozwój oprogramowania

Ćwiczenia (Ć)

1. Ćwiczenia z tworzenia specyfikacji wymagań
2. Ćwiczenia z wykorzystania narzędzi CASE.
3. Ćwiczenia z wykorzystania diagramów UML behawioralnych
4. Ćwiczenia z wykorzystania diagramów UML strukturalnych

5. Ćwiczenia z testowania oprogramowania
6. Ćwiczenia z inspekcji kodu
7. Ćwiczenia wspierające zarządzanie projektem
8. Komunikacja między systemami rozproszonymi

Literatura podstawowa

1. Hofmeister C., Nord R., Soni D., Inżynieria programowania. Tworzenie architektury oprogramowania. WNT 2006
2. Sommerville I.: Software Engineering, Tenth Edition, Pearson Education Limited 2016
3. Karl E Wieggers, Joy Beatty, Specyfikacja oprogramowania. Inżynieria wymagań. Helion 2014
4. Gamma E., Helm R., Johnson R., Vlissides J., Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku. Helion 2010

Literatura uzupełniająca

1. Koszłajda A.: Od pomysłu do przemysłu Projekty IT w praktyce, Helion, Gliwice, 2019
2. Wrycza S.: UML 2.1, Helion, Gliwice, 2006
3. Lehman, M.M.; Belady, L.A.: Program evolution : processes of software change. London: Academic Press Inc, 1985
4. Autor: Sander Rossel, Continuous Integration, Delivery, and Deployment. Reliable and faster software releases with automating builds, tests, and deployment, Helion 2017

Źródła dodatkowe

1. <https://support.office.com/pl-pl/article/diagramy-uml-w-programie-visio-ca4e3ae9-d413-4c94-8a7a-38dac30cbcd6>
2. <https://www.omg.org/>
3. <https://www.omg.org/spec/UML>
4. <https://docs.github.com/en/get-started/start-your-journey/hello-world>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	60 godz.....	32 godz.
• Zapoznanie się z literaturą przedmiotu.....	20 godz.....	20 godz.
• Przygotowanie się do zajęć.....	5 godz.....	18 godz.
• Przygotowanie się do zaliczenia końcowego.....	6 godz.....	12 godz.
• Samodzielne wykonanie zadań.....	4 godz.....	8 godz.
• Wypełnienie dokumentacji zajęć.....	5 godz.....	10 godz.
Suma obciążeń:.....	100 godz.....	100 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen ze wszystkich obowiązkowych form weryfikacji efektów uczenia się. Ocena końcowa wyliczana jest jako średnia ważona z wyniku zaliczenia testowego (40%) oraz zaliczenia ćwiczeń (60%) . Dodatkowo mogą być wliczone wyniki przesłanych na platformę Wlodek zadania.

Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi:

do 50% - ocena niedostateczna,

51% - 60% - ocena dostateczna,

61% - 70% - ocena dostateczna plus,

71% - 80% - ocena dobra

81% - 90% - ocena dobra plus,

powyżej 90% - ocena bardzo dobra.

Projektowanie UX/UI

Nazwa zajęć po angielsku: UX/UI Design

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: IV

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot przeznaczony jest dla studentów studiów inżynierskich na kierunku Informatyka. Celem przedmiotu jest rozwinięcie umiejętności analizowania doświadczeń użytkownika w kontekście projektów informatycznych oraz projektowania interfejsów użytkownika. Studenci zapoznają się m.in. z procesem projektowania UX/UI, zasadami dostępności, heurystykami Nielsena oraz narzędziami prototypowania. Zajęcia realizowane są w formie wykładów konwersatoryjnych, ćwiczeń oraz projektu, umożliwiając zdobycie wiedzy, rozwój umiejętności analityczno-projektowych oraz kształtowanie postawy nastawionej na potrzeby użytkownika, szczególnie w ramach jego interakcji z systemem. Studenci uczą się współpracy z biznesem, odkrywania potrzeb klienta, opracowania wizji systemu, zapoznają się z nowoczesnymi metodami współpracy z klientem m.in. przy pomocy tablicy Kanban, wykorzystują m.in. szablony mapy empatii, szablony persony oraz matrycę propozycji wartości, budują diagramy z wykorzystaniem języka UML, projektują interfejs użytkownika.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia znaczenie projektowania komunikacji człowieka z systemem informatycznym oraz rolę UX/UI w jakości systemu informatycznego	K_W03
charakteryzuje proces projektowania systemów interaktywnych, w tym analizę potrzeb użytkownika, architekturę informacji, prototypowanie, testowanie i iterację projektu	K_W04
identyfikuje metody, techniki i narzędzia projektowania UX/UI, w tym persony, mapę empatii, user journey, user flows, wireframes, prototypy, testy użyteczności i design systems	K_W05
rozpoznaje standardy i dobre praktyki projektowania interfejsów, w tym WCAG 2.2, EN 301 549, ISO 9241-210, WAI-ARIA APG oraz wytyczne systemów projektowych	K_W06
rozdziela zasady licencjonowania zasobów graficznych, ikon, fontów, zdjęć, bibliotek UI i szablonów używanych w projektach interfejsów	K_W09
Umiejętności (student potrafi)	
integruje informacje z analizy użytkowników, wymagań biznesowych, heurystyk, standardów dostępności i wyników testów przy opracowaniu projektu UX/UI	K_U02
dobiera narzędzia prototypowania, modelowania, dokumentowania i testowania interfejsu do wymagań projektu informatycznego	K_U04
planuje i przeprowadza test użyteczności lub audyt heurystyczny zaprojektowanego interfejsu, interpretuje wyniki i wyciąga wnioski projektowe	K_U06

uwzględnia aspekty pozatechniczne projektowania interfejsu: potrzeby użytkowników, dostępność, inkluzywność, ograniczenia poznawcze, etykę i kontekst użycia	K_U08
projektuje interfejs użytkownika systemu informatycznego na podstawie potrzeb użytkowników, architektury informacji, user flows, prototypów i wyników testów	K_U11
planuje i organizuje pracę własną oraz zespołu projektowego z użyciem Kanban, backlogu, kryteriów akceptacji i dokumentacji projektowej	K_U17
Kompetencje społeczne (student jest gotów do)	
jest gotów do zasięgania opinii ekspertów z zakresu UX, dostępności, psychologii poznawczej, prawa autorskiego lub technologii front-end w przypadku trudności projektowych	K_K02
jest gotów do inicjowania działań projektowych na rzecz dostępnych i inkluzywnych interfejsów ograniczających wykluczenie cyfrowe	K_K03
jest gotów do samorozwoju w zakresie nowych metod, standardów, narzędzi i technologii projektowania UX/UI	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Dyskusja
- Praca ze źródłem elektronicznym
- Studium przypadku
- Metoda ćwiczeń
- Metoda projektu

Metody oceniania:

- Ocena aktywności w kursie
- Praca pisemna
- Zaliczenie testowe
- Rozliczenie projektu

Wykład konwersatoryjny (W)

1. UX/UI w projektach informatycznych: rola w jakości produktu, użyteczności, dostępności i wartości biznesowej.
2. Proces projektowania zorientowanego na użytkownika (Human-Centered Design i ISO 9241-210)
3. Rozpoznawanie potrzeb klienta i ich dokumentowanie z wykorzystaniem design thinking, wywiadów, ankiet, obserwacji, person, map empatii, map ścieżki użytkownika, matrycy propozycji wartości.
4. Architektura informacji: struktura treści, nawigacja, user flows, sitemap, card sorting.
5. Projektowanie interakcji: stany komponentów, informacja zwrotna, sugerowany sposób użycia elementów (afordancje), zapobieganie błędom, krótkie komunikaty interfejsu.
6. UI design: typografia, kolor, kontrast, siatki, hierarchia wizualna, Gestalt, responsywność.
7. Heurystyki Nielsena, zasady Normana i wzorce projektowe UI.
8. Dostępność cyfrowa: WCAG 2.2, EN 301 549, EAA, WAI-ARIA APG, projektowanie dla technologii wspomagających.
9. Design systems: komponenty, warianty, tokeny, style, biblioteki, spójność i przekazanie projektu do front-endu.
10. Prototypowanie: low-fidelity, mid-fidelity, high-fidelity, prototypy interaktywne i testowalne.
11. Testowanie użyteczności: scenariusze, zadania, metryki, obserwacje, raport, iteracja.
12. Etyka UX: dark patterns, manipulacja, prywatność, inkluzywność, licencje zasobów i odpowiedzialne użycie AI.

Ćwiczenia (Ć)

1. Analiza problemu projektowego: interesariusze, użytkownicy, cele biznesowe i ograniczenia.
2. Persony, mapa empatii, user journey, value proposition.
3. User stories, kryteria akceptacji i backlog projektu UX/UI.
4. Architektura informacji: sitemap, user flows, struktura nawigacji.
5. Makiety low-fidelity i audyt heurystyczny.
6. Projekt wizualny: typografia, kolor, kontrast, komponenty, stany i responsywność.
7. Prototyp high-fidelity w Figmie lub narzędziu równoważnym.
8. Audyt dostępności według WCAG 2.2.
9. Test użyteczności: scenariusz, zadania, obserwacja, wnioski.
10. Iteracja projektu po testach i przygotowanie dokumentacji dla zespołu front-end.

Projekt (P)

1. Wybór problemu i określenie celów UX.
2. Analiza użytkowników i kontekstu użycia.
3. Opracowanie person, user journey i wymagań.
4. Architektura informacji i user flows.
5. Makiety low-fidelity.
6. Prototyp high-fidelity.
7. Audyt dostępności WCAG 2.2 i korekta projektu.
8. Test użyteczności z użytkownikami lub test ekspercki z jasno opisanym protokołem.
9. Dokumentacja projektu: decyzje projektowe, komponenty, stany, ograniczenia, wyniki testów.
10. Prezentacja projektu.

Literatura podstawowa

1. Materiały na platformie WLODEK
2. C. Badura, UXUI Design zoptymalizowany. Manual Book, Helion 2019
3. M. Ritter, C. Winterbottom, UX w projektowaniu witryn internetowych, Helion 2018

Literatura uzupełniająca

1. M. Bartyzel, Oprogramowanie szyte na miarę. Jak rozmawiać z klientem, który nie wie, czego chce. Wydanie II rozszerzone, Helion OnePress 2015
2. C. Badura, UXUI Design zoptymalizowany. Workshop Book, Helion 2019
3. D. Norman, Dizajn na co dzień, Karakter 2018
4. B. Michalska-Dominiak, P. Grocholiński, Poradnik design thinking, Onepress 2019
5. S. Krug: Nie każ mi myśleć, O życiowym podejściu do funkcjonalności stron internetowych, Wyd. III. Helion 2014
6. S. Krug, Przetestuj ją sam! O funkcjonalności stron internetowych. Helion 2010

Źródła dodatkowe

1. <https://www.webusability.pl/>
2. <https://www.nngroup.com/articles/ten-usability-heuristics/>
3. medium.muz.li/gestalt-principles-in-ui-design-6b75a41e9965
4. <https://www.checklist.design/>
5. <https://www.w3.org/TR/WCAG22/>
6. <https://www.w3.org/WAI/ARIA/apg/>
7. <https://www.iso.org/standard/77520.html>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	60 godz.....	32 godz.
• Zapoznanie się z literaturą przedmiotu.....	6 godz.....	14 godz.
• Przygotowanie projektu.....	20 godz.....	28 godz.
• Wypełnienie dokumentacji zajęć.....	5 godz.....	12 godz.
• Przygotowanie się do zaliczenia końcowego.....	4 godz.....	6 godz.
• Realizacja kursu e-learningowego.....	5 godz.....	8 godz.
Suma obciążeń:.....	100 godz.....	100 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest przyswojenie wiadomości na temat UX/UI, zdobycie umiejętności projektowania wrażeń użytkownika oraz interfejsu użytkownika oraz nabycie kompetencji polegających na samodzielnym i grupowym rozwiązywaniu problemów zawodowych.

Studenci mają dostęp do materiałów na platformie WLODEK, które poszerzają treści teoretyczne oraz praktyczne przekazywane na zajęciach. Przerabiając kolejne materiały, odpowiadają na pytania kontrolne oraz przesyłają wykonane na zajęciach lub samodzielnie zadania. Pod koniec semestru przewidziane jest zaliczenie testowe treści teoretycznych, złożenie kompletnej dokumentacji z zajęć ćwiczeniowych oraz przesłanie i zaprezentowanie projektu zespołowego.

Ocena końcowa wyliczana jest średnia ważona z pytań kontrolnych umieszczonych w lekcjach (10%), z wyniku

zaliczenia testowego (30%), dokumentacji ćwiczeniowej (25%) oraz oceny z projektu (35%).

Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi:

do 50% - ocena niedostateczna,

51% - 60% - ocena dostateczna,

61% - 70% - ocena dostateczna plus,

71% - 80% - ocena dobra

81% - 90% - ocena dobra plus,

powyżej 90% - ocena bardzo dobra.

Realizacja projektu informatycznego

Nazwa zajęć po angielsku: Implementation of an IT project

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: IV

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Marcin Zawadzki

Opis zajęć:

Przedmiot „Realizacja projektu informatycznego” integruje umiejętności nabyte na wcześniejszych kursach programowania, baz danych i technologii internetowych w ramach realizacji zespołowego projektu informatycznego. Studenci planują, projektują, implementują, testują i wdrażają aplikację webową, stosując metodyki zwinne (Scrum), narzędzia zarządzania projektem oraz praktyki inżynierii oprogramowania. Kurs obejmuje cykl życia projektu, zarządzanie backlogiem, harmonogramowanie, pracę z repozytorium Git, dokumentację techniczną, testy, przegląd kodu i prezentację wyników. Ponad 60% czasu zajęć stanowią aktywności praktyczne ukierunkowane na wytworzenie działającego produktu programistycznego.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
Wyjaśnia cykl życia projektu informatycznego, role projektowe, zakres, harmonogram, ryzyka, jakość i kryteria odbioru produktu.	K_W01
Porównuje podejścia predykcyjne, zwinne i hybrydowe oraz identyfikuje narzędzia wspierające planowanie, monitorowanie i realizację projektu.	K_W02
Wyjaśnia standardy dokumentowania wymagań, API, testów, repozytorium, jakości kodu i wdrożenia.	K_W04
Wyjaśnia ekonomiczne, prawne i etyczne uwarunkowania realizacji projektu informatycznego.	K_W05
<i>Umiejętności (student potrafi)</i>	
Planuje zakres, harmonogram, zasoby, backlog, kamienie milowe i ryzyka projektu informatycznego.	K_U01
Dobiera źródła, wymagania, dokumentację techniczną i wzorce rozwiązań potrzebne do realizacji projektu.	K_U02
Realizuje projekt informatyczny zgodnie z przyjętymi wymaganiami, harmonogramem i standardami dokumentacji, integrując komponenty systemu.	K_U03
Stosuje narzędzia do zarządzania projektem, kontroli wersji, dokumentowania, testowania i prezentacji wyników.	K_U11
<i>Kompetencje społeczne (student jest gotów do)</i>	
Współdziała w zespole projektowym, dokumentując własny wkład, komunikację i decyzje projektowe.	K_K01

Student jest gotów do krytycznej oceny postępu prac, jakości produktu, dokumentacji i własnego wkładu w projekt.	K_K02
Student jest gotów do zasięgania opinii ekspertów w przypadku trudności projektowych, technicznych lub organizacyjnych.	K_K03
Student jest gotów do działania przedsiębiorczego przy definiowaniu wartości projektu i sposobu jego wdrożenia.	K_K04
Student jest gotów do przestrzegania zasad etyki zawodowej, prawa autorskiego, licencji i odpowiedzialnego wykorzystania kodu oraz narzędzi AI.	K_K05

Metody i formy kształcenia:

- Wykład informacyjny
- Wykład problemowy
- Metoda projektu
- Studium przypadku
- Burza mózgów
- Praca ze źródłem elektronicznym

Metody oceniania:

- Zaliczenie opisowe
- Rozliczenie projektu

Wykład konwersatoryjny (W)

1. Cykl życia projektu i planowanie: cel, interesariusze, metodyki (Scrum, Kanban), role, backlog, harmonogram, ryzyka.
2. Wymagania, dokumentacja i architektura: user stories, kryteria akceptacji, projekt techniczny, REST API, OpenAPI, model danych.
3. Testowanie i zarządzanie zmianą: testy funkcjonalne, integracyjne, code review, CI/CD, zarządzanie konfliktem, retrospektywa.
4. Wdrożenie i utrzymanie: deployment, dokumentacja techniczna, instrukcja uruchomienia, aspekty prawne, etyka, licencje, AI.

Ćwiczenia (Ć)

1. Formowanie zespołów: wybór tematu, podział ról, karta projektu, repozytorium Git, backlog.
2. Sprint 1 Planning: planowanie pierwszego sprintu, user stories, estymacja, Definition of Done.
3. Sprint 1 Review: demo MVP, przegląd kodu, feedback, retrospektywa.
4. Sprint 2 Review: rozbudowa funkcjonalności, integracja, testy, dokumentacja postępów.
5. Sprint 3 i testowanie: finalizacja, testy odbiorcze, audyt jakości, przygotowanie do wdrożenia.
6. Dokumentacja i code review: dokumentacja OpenAPI, README, raport wkładu, prezentacja końcowa.

Literatura podstawowa

1. Schwaber K., Sutherland J., "The Scrum Guide" (scrumguides.org), 2020.
2. Wiegers K., Beatty J., "Software Requirements". Microsoft Press.

Literatura uzupełniająca

1. Oficjalna dokumentacja ASP.NET Core (learn.microsoft.com/aspnet/core)
2. Oficjalna dokumentacja Git (git-scm.com/doc)
3. GitHub Docs — Pull Requests i Code Review (docs.github.com)

Źródła dodatkowe

1. OpenAPI Specification (swagger.io/specification)
2. Atlassian — poradniki Scrum i Kanban (atlassian.com/agile)
3. PMBOK Guide — Project Management Institute (pmi.org)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....45 godz.....24 godz.
- Zapoznanie się z literaturą przedmiotu.....5 godz.....5 godz.
- Przygotowanie projektu.....25 godz.....36 godz.
- Samodzielne wykonanie zadań.....0 godz.....10 godz.
- Suma obciążeń:.....75 godz.....75 godz.

Zaliczenie zajęć:

Ostateczna ocena z przedmiotu jest średnią ważoną z komponentu wykładowego i ćwiczeniowego (projektowego).
Warunkiem koniecznym jest uzyskanie pozytywnych ocen z obu komponentów (minimum 51%).

Wagi poszczególnych komponentów:

Wykład: 20%

Ćwiczenia (projekt): 80%

Warunki zaliczenia wykładu:

Zaliczenie testowe końcowe (metodyki, narzędzia, standardy): 100% oceny z wykładu.

Weryfikuje efekty: PEU_W01–PEU_W04.

Warunki zaliczenia ćwiczeń (projektu):

Rozliczenie projektu zespołowego, na które składa się:

- Działająca aplikacja (25%)
- Dokumentacja (backlog, harmonogram, OpenAPI, README) (20%)
- Jakość kodu i testy (15%)
- Repozytorium Git — historia commitów, pull requesty, code review (15%)
- Raport indywidualnego wkładu (15%)
- Prezentacja i obrona projektu (10%)

Weryfikuje efekty: PEU_U01–PEU_U05, PEU_K01–PEU_K04.

Próg osiągnięcia efektów uczenia się:

Każdy efekt uczenia się uznaje się za osiągnięty, jeżeli student uzyskał minimum 51% punktów z przypisanych do tego efektu zadań weryfikujących.

Skala ocen:

91% – 100%: bardzo dobry (5.0)

81% – 90%: dobry plus (4.5)

71% – 80%: dobry (4.0)

61% – 70%: dostateczny plus (3.5)

51% – 60%: dostateczny (3.0)

50% i poniżej: niedostateczny (2.0)

Podstawy technologii internetowych

Nazwa zajęć po angielsku: Basics of Web Technology

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: II

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: kurs e-learningowy

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

W ramach zajęć "Podstawy technologii internetowych" studenci zdobywają wiedzę na temat technologii umożliwiających budowanie responsywnych i interaktywnych stron internetowych oraz nabywają umiejętności ich wykonania przy pomocy języka HTML, stylów kaskadowych CSS, języka JavaScript oraz dostępnych bibliotek i frameworków typu Bootstrap. Studenci w trakcie zajęć poznają podstawowe zasady projektowania stron internetowych oraz ćwiczą praktycznie ich wykorzystanie. Przygotowany kurs e-learningowy zawiera informacje teoretyczne, opis znaczników i poleceń, przykłady ich wykorzystania, pytania kontrolne oraz praktyczne przykłady do wykonania. Studenci korzystają z dostępnych narzędzi typu Notepad++ czy Visual Studio Code. Po ukończeniu zajęć studenci dysponują podstawową wiedzą z zakresu budowania responsywnych stron internetowych i są przygotowani do realizacji przedmiotu „Programowanie aplikacji internetowych” w zakresie budowy widoków dla użytkowników.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia zastosowanie technologii WWW w budowie stron internetowych i interfejsu użytkownika	K_W03
identyfikuje techniki, narzędzia i środowiska stosowane do tworzenia stron internetowych, w tym HTML, CSS, JavaScript, DevTools, edytor kodu, walidatory i frameworki CSS	K_W05
rozpoznaje standardy i dobre praktyki technologii internetowych, w tym HTML Living Standard, CSS, ECMAScript, WCAG 2.2, zasady semantyki, responsywności i walidacji kodu	K_W06
Umiejętności (student potrafi)	
pozyskuje i wykorzystuje dokumentację MDN, W3C, WHATWG, ECMAScript i dokumentację frameworków do rozwiązywania problemów podczas budowy strony	K_U01
dobiera narzędzia, biblioteki i techniki front-endowe do wymagań strony internetowej	K_U04
projektuje i wykonuje responsywną stronę internetową z użyciem semantycznego HTML, CSS, JavaScript i wybranego frameworka lub własnego układu responsywnego	K_U11
stosuje standardy i dobre praktyki W3C, WCAG, ECMAScript oraz zasady poprawności, dostępności, responsywności i podstaw bezpieczeństwa front-endu	K_U12
Kompetencje społeczne (student jest gotów do)	
jest gotowy do krytycznej oceny jakości własnej strony pod kątem poprawności, dostępności, responsywności, użyteczności i zgodności ze standardami	K_K01
jest gotów do samorozwoju w zakresie zmieniających się standardów, narzędzi i praktyk technologii internetowych	U_K07

Metody i formy kształcenia:

- Studium przypadku
- Praca ze źródłem elektronicznym
- Metoda ćwiczeń
- Wykład problemowy

Metody oceniania:

- Częstkowe sprawdziany testowe - kolokwia
- Zaliczenie praktyczne
- Ocena aktywności w kursie
- Praca domowa

Wykład konwersatoryjny (W)

1. Jak działa WWW: klient-serwer, HTTP/HTTPS, URL, DNS, przeglądarka, serwer, DevTools.
2. Funkcjonowanie stron i aplikacji internetowych. Podział na back-end (server-side) i front-end (client-side)
3. Język znaczników HTML - semantyczna struktura dokumentu, formularze, multimedia, metadane, walidacja.
4. Arkusze stylów kaskadowych CSS - selektory, kaskada, box model, Flexbox, Grid, media queries, responsive images.
5. JavaScript / ECMAScript - składnia, DOM, zdarzenia, formularze, walidacja, moduły, obsługa błędów.
6. API i dane: fetch, JSON, podstawy komunikacji z usługą, obsługa odpowiedzi i błędów.
7. Responsywność: mobile-first, breakpoints, testowanie na różnych wielkościach (viewport).
8. Frameworki CSS - siatka, komponenty, użycie i ograniczenia.
9. Dostępność: WCAG 2.2, semantyka, kontrast, klawiatura, tekst alternatywny, formularze.
10. Przygotowanie do dalszych przedmiotów: front-end jako warstwa widoku aplikacji internetowej.

Ćwiczenia (Ć)

1. Konfiguracja środowiska pracy: VS Code, przeglądarka, repozytorium.
2. Projektowanie struktury strony internetowej.
3. Semantyczny dokument HTML z nagłówkiem, nawigacją, treścią, formularzem i stopką.
4. Ustalanie wyglądu strony poprzez formatowanie przy pomocy stylów kaskadowych CSS - box model, typografia, kolory, układ elementów.
5. Formularze HTML i walidacja po stronie klienta.
6. Programowanie na stronach internetowych JavaScript - DOM, zdarzenia, interakcja, walidacja danych.
7. Budowa responsywnych stron internetowych w systemie siatkowym przy pomocy frameworka np. Bootstrap
8. Projekt i wykonanie responsywnej strony dla Parlamentu Studentów SWPW

Literatura podstawowa

1. J. Duckett, HTML i CSS. Zaprojektuj i zbuduj witrynę WWW. Helion 2014
2. J. Duckett, JAVASCRIPT i JQUERY. Interaktywne strony WWW dla każdego. Helion 2015
3. Materiały na platformie WLODEK

Literatura uzupełniająca

1. R. Grycan, BOOTSTRAP. Tworzenie własnych stylów graficznych. Helion 2017

Źródła dodatkowe

1. <https://www.w3.org/>
2. <https://html.spec.whatwg.org/multipage/>
3. <https://ecma-international.org/>
4. <https://www.w3schools.com/>
5. <https://getbootstrap.com/>
6. <https://developer.mozilla.org/en-US/docs/Web>
7. https://developer.mozilla.org/en-US/docs/Learn_web_development

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....45 godz.....24 godz.
- Realizacja kursu e-learningowego.....10 godz.....10 godz.
- Przygotowanie się do kolokwiów.....5 godz.....10 godz.
- Przygotowanie się do zaliczenia końcowego.....5 godz.....11 godz.
- Samodzielne wykonanie zadań.....10 godz.....20 godz.
- Suma obciążeń:.....75 godz.....75 godz.

Zaliczenie zajęć:

W trakcie semestru studenci przerabiają dodatkowo kurs e-learningowy. W każdej lekcji odpowiadają na pytania kontrolne i zdobywają punkty. Na zajęciach praktycznych studenci rozpoczynają pracę z witrynami, które kończą w domu zgodnie z opisem. Na zajęciach praktycznych studenci przystępują też do czterech kolokwii, które nie tylko weryfikują ich wiedzę teoretyczną, ale także sprawdzają umiejętność wykorzystania tej wiedzy w praktycznych przykładach. Na ostatnich zajęciach ćwiczeniowych studenci przystępują do zaliczenia praktycznego, które polega na przygotowaniu responsywnej strony internetowej zgodnie z podanym wzorem. Oceniane są umiejętności praktyczne oraz zaangażowanie w rozwiązanie zadania i zastosowanie zdobytej wiedzy w realnej sytuacji. Ocena końcowa wyliczana jest z czterech składowych: punkty zdobyte podczas realizacji kursu e-learningowego (15%), wynik kolokwii śródsemestralnych (35%), wynik prac domowych polegających na dokończeniu witryn (15%) oraz ocena z zaliczenia praktycznego (35%). Ocena końcowa zależy od liczby zdobytych punktów i wyliczana jest następująco:

0% - 50% - niedostateczny

51%-60% - dostateczny

61%-70% - dostateczny plus

71%-80% - dobry

81%-90% - dobry plus

91%-100% - bardzo dobry.

Programowanie aplikacji internetowych

Nazwa zajęć po angielsku: Web application development

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 5

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....45 / 24 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot "Programowanie aplikacji internetowych" przeznaczony jest dla studentów kierunku Informatyka, którzy znają już technologie umożliwiające budowę stron internetowych (HTML, CSS, JavaScript), potrafią określić wymagania klienta i funkcjonalności aplikacji (UX), potrafią zaprojektować model aplikacji (UML), prototypować interfejs użytkownika (UI), znają bazy danych (SQL), potrafią programować (C#) oraz potrafią posługiwać się systemami kontroli wersji (GitHub). Celem przedmiotu jest zapoznanie studentów z koncepcją aplikacji wielowarstwowych oraz wykształcenie umiejętności projektowania architektury aplikacji, implementacji logiki biznesowej, warstwy interfejsu użytkownika, a także zapewnienia bezpieczeństwa i spójności danych. Studenci uczą się też współpracy zespołowej przy realizacji projektu. Część praktyczna przedmiotu nastawiona jest na opracowywanie aplikacji typu CMS, generującej strony internetowe na podstawie informacji zapisanych w bazach danych. Studenci wykorzystują model MVC, bazę danych SQL Server, język zapytań LINQ, ORM, składnię Razor i pracują w Visual Studio. Po ukończeniu przedmiotu studenci będą znali zasady opracowywania aplikacji pracujących w środowisku internetowym oraz będą potrafili zaprojektować i wykonać nowoczesną aplikację webową w oparciu o architekturę MVC oraz platformę ASP.NET.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
omawia wzorce architektoniczne wykorzystywane w aplikacjach internetowych	K_W02
wyjaśnia zastosowanie aplikacji wielowarstwowych, MVC, wzorca repozytorium, dependency injection i warstwowej organizacji kodu w systemach webowych	K_W03
charakteryzuje proces obsługi żądania HTTP w ASP.NET Core, w tym routing, middleware, kontroler, model, widok, walidację, sesję i odpowiedź aplikacji	K_W04
identyfikuje metody, techniki i narzędzia stosowane w tworzeniu aplikacji internetowych: ASP.NET Core, EF Core, LINQ, Razor, Identity oraz zna podstawowe mechanizmy walidacji danych (DataAnnotations) w formularzach internetowych po stronie klienta oraz serwera	K_W05
rozpoznaje standardy i dobre praktyki tworzenia aplikacji webowych, w tym HTTP, HTML/CSS/JS, REST, OWASP Top 10, walidację, uwierzytelnianie, autoryzację, logging i dokumentację	K_W06
Umiejętności (student potrafi)	
integruje wiedzę z technologii internetowych, baz danych, programowania obiektowego, projektowania UX/UI, kontroli wersji i pracy zespołowej przy realizacji aplikacji internetowej	K_U02
dobiera środowisko programistyczne, biblioteki, pakiety, ORM, system kontroli wersji i narzędzia testowania do wymagań aplikacji webowej	K_U04

rozwiązuje praktyczny problem projektowy przez zaprojektowanie modelu domeny, bazy danych, funkcji CRUD, walidacji, autoryzacji i widoków aplikacji	K_U05
ocenia działanie aplikacji webowej pod kątem wymagań, przepływu danych, jakości kodu, zabezpieczeń, obsługi błędów i zgodności z przyjętą architekturą	K_U10
projektuje i implementuje aplikację internetową z użyciem ASP.NET Core, EF Core, warstwy prezentacji, warstwy danych, mechanizmów walidacji i uwierzytelniania	K_U11
stosuje standardy i dobre praktyki programowania aplikacji internetowych, w tym bezpieczne operacje na danych, walidację, autoryzację, dokumentację, testy i kontrolę wersji	K_U12
planuje i organizuje pracę własną oraz zespołu projektowego, wykorzystując repozytorium, zadania, harmonogram, przegląd kodu i dokumentację projektu	K_U17
współdziała z członkami zespołu przy implementacji, testowaniu, dokumentowaniu i prezentacji aplikacji internetowej	K_U18
Kompetencje społeczne (student jest gotów do)	
jest gotów do zasięgania opinii ekspertów, prowadzącego i członków zespołu przy rozwiązywaniu problemów architektonicznych, projektowych i bezpieczeństwa aplikacji	K_K02
jest gotów do przestrzegania zasad etyki zawodowej, ochrony danych, praw autorskich, licencji i odpowiedzialnego tworzenia oprogramowania webowego, w tym z wykorzystaniem AI	K_K05
jest gotów do samorozwoju w zakresie aktualnych wersji .NET, ASP.NET Core, EF Core, bezpieczeństwa, narzędzi DevOps i praktyk web developmentu	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Dyskusja
- Metoda sytuacyjna
- Metoda ćwiczeń
- Studium przypadku
- Metoda projektu

Metody oceniania:

- Zaliczenie testowe
- Zaliczenie praktyczne
- Rozliczenie projektu

Wykład konwersatoryjny (W)

1. Architektura aplikacji webowych: klient-serwer, HTTP/HTTPS, MVC, warstwy, monolit modułowy, API.
2. ASP.NET Core - pipeline, middleware, routing, dependency injection, konfiguracja.
3. MVC, Razor, Razor Pages, Blazor i Minimal APIs - zakres użycia, zalety i ograniczenia.
4. Model domeny, ViewModel, DTO, walidacja i mapowanie danych.
5. EF Core - DbContext, relacje, migracje, seed, transakcje, ładowanie danych, zapytania LINQ.
6. Formularze, model binding, walidacja po stronie serwera i klienta.
7. Uwierzytelnianie i autoryzacja - ASP.NET Core Identity, role, claims, polityki dostępu.
8. Bezpieczeństwo - OWASP Top 10, XSS, CSRF, SQLi, Broken Access Control, SSRF.
9. Testy - jednostkowe, integracyjne, testy kontrolerów/usług, testy walidacji i zabezpieczeń.
10. Logging, obsługa wyjątków, diagnostyka i podstawy observability.
11. Git, pull request, code review, konwencje kodu i dokumentacja projektu.
12. Publikacja i podstawy CI/CD; wersjonowanie; dokumentacja.

Ćwiczenia (Ć)

1. Analiza wymagań aplikacji CMS – specyfikacja funkcjonalna i нефункционална.
2. Tworzenie klas modelu – DTO i walidacja danych (Data Annotation).
3. EF Core - konfiguracja DbContext, relacje, migracja danych (metoda Seed) i aktualizacja schematu bazy.
4. Dodawanie kontrolerów i akcji CRUD z walidacją, komunikatami błędów i obsługą wyjątków.
5. Projektowanie widoków – Razor i Bootstrap.

6. Formularze, model binding, GET/POST, walidacja klient–serwer.
7. Autoryzacja. Filtry akcji i instrukcje warunkowe w zabezpieczaniu logiki aplikacji.
8. Bezpieczeństwo - demonstracja i zabezpieczenie przed XSS, CSRF, SQLi oraz błędami kontroli dostępu.
9. Widoki częściowe i dzielenie interfejsu.
10. Zarządzanie sesją i użytkownikami – ASP.NET Identity.
11. Logging, konfiguracja środowisk i podstawy deploymentu. Code review i poprawa jakości kodu.

Projekt (P)

1. Zebranie wymagań funkcjonalnych i нефункциональных aplikacji.
2. Projekt aplikacji internetowej. Scenariusze. Diagramy przypadków użycia.
3. Projekt bazy danych (ERD) i definiowanie klas modelu (Model).
4. Architektura aplikacji i uzasadnienie decyzji technicznych.
5. Implementacja migracji danych i seed testowy.
6. Budowa kontrolerów z akcjami CRUD oraz widoków z uwzględnieniem UX/UI, responsywności i podstaw dostępności.
7. Integracja walidacji danych i obsługa błędów użytkownika; zastosowanie filtrów, ról i autoryzacji.
8. Testy i weryfikacja zabezpieczeń.
9. README, instrukcja uruchomienia, opis konfiguracji i ograniczeń.
10. Prezentacja projektu, rozliczenie indywidualnego wkładu i złożenie dokumentacji.

Literatura podstawowa

1. <https://learn.microsoft.com/en-us/aspnet/core/>
2. <https://dotnettutorials.net/lesson/introduction-asp-net-core-mvc/>
3. Materiały na platformie WLODEK

Literatura uzupełniająca

1. Mark J. Price, C# 11 i .NET 7 dla programistów aplikacji wieloplatformowych. Twórz aplikacje, witryny WWW oraz serwisy sieciowe za pomocą ASP.NET Core 7, Blazor i EF Core 7. Wydanie VII, Helion 2023
2. Jason De Oliveira, Michel Bruchet, ASP.NET Core 2.0. Wprowadzenie, Helion 2019
3. Dawid Borycki, Maciej Pakulski, Maciej Grabek, Jacek Matulewski, ASP.NET MVC. Kompletny przewodnik dla programistów interaktywnych aplikacji internetowych w Visual Studio, Helion 2014

Źródła dodatkowe

1. <https://abp.io/docs/1.1/Tutorials/AspNetCore-Mvc/Part-I>
2. <https://owasp.org/Top10/2024/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Zapoznanie się z literaturą przedmiotu.....	5 godz.....	15 godz.
• Samodzielne wykonanie zadań.....	10 godz.....	25 godz.
• Przygotowanie projektu.....	25 godz.....	33 godz.
• Przygotowanie się do zaliczenia końcowego.....	10 godz.....	12 godz.
Suma obciążeń:.....	125 godz.....	125 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest przyswojenie wiadomości na temat projektowania i zdobycie umiejętności programowania aplikacji internetowych opartych na wzorcu MVC. Ocena końcowa zależy od liczby punktów zdobytych z testowego zaliczenia wykładu (30%), wykonanego modułu w aplikacji w trakcie zaliczenia ćwiczeń (35%) oraz wykonanej pracy projektowej (35%). Ocena wkładu każdego studenta w realizację projektu jest ustalana indywidualnie na podstawie wykonanego zakresu, jego prezentacji i omówienia oraz odpowiedzi na pytania dotyczące podjętych decyzji i powiązania własnej części z całością rozwiązania. Warunkiem uzyskania pozytywnej oceny końcowej jest uzyskanie co najmniej połowy punktów z każdego obowiązkowego składnika weryfikacji efektów uczenia się. Ocena końcowa jest następnie obliczana jako średnia ważona wyników poszczególnych składników. Ocena końcowa zależy od liczby uzyskanych punktów:

do 50% - ocena niedostateczna,
od 50% do 60% - ocena dostateczna,

od 60% do 70% - ocena dostateczna plus,
od 70% do 80% - ocena dobra,
od 80% do 90% - ocena dobra plus,
od 90% - ocena bardzo dobra.

Sztuczna inteligencja

Nazwa zajęć po angielsku: Artificial intelligence

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VII

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Mariusz Sochacki

Opis zajęć:

Celem przedmiotu „Sztuczna Inteligencja” jest zapoznanie studentów z koncepcjami, metodami oraz technikami stosowanymi w dziedzinie sztucznej inteligencji (AI). Przedmiot obejmuje teoretyczne i praktyczne aspekty AI, w tym algorytmy uczenia maszynowego, sieci neuronowe, przetwarzanie języka naturalnego oraz zastosowania AI w różnych dziedzinach przemysłu i nauki.

Ponadto w trakcie zajęć nastąpi pokazanie zastosowania symulacji ludzkiej inteligencji w maszynach, które mają zaprogramowane myślenie w oparciu o myślenie ludzkie i naśladowanie ludzkich działań. Prezentacja wykorzystania takich technologii takich jak RPA, NLU/NLG, TTS, ASR.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Student wyjaśnia pojęcia, metody i zależności występujące w uczeniu maszynowym, sieciach neuronowych, NLP i generatywnej AI.	K_W02
Student charakteryzuje praktyczne zastosowania AI w systemach informatycznych, automatyzacji procesów, analizie danych, chatbotach i systemach wspomagania decyzji.	K_W03
Student identyfikuje metody, techniki i narzędzia stosowane do przygotowania danych, trenowania, walidacji, interpretacji i wdrażania modeli AI.	K_W05
Student rozpoznaje standardy, regulacje i dobre praktyki dotyczące AI, zasady przejrzystości, bezpieczeństwa i dokumentowania modeli.	K_W06
Umiejętności (student potrafi)	
Student przygotowuje dane, trenuje i ocenia model uczenia maszynowego lub sieci neuronowej z użyciem gotowych bibliotek.	K_U07
Student uwzględnia aspekty etyczne, prawne, społeczne i bezpieczeństwa podczas projektowania oraz oceny rozwiązania AI.	K_U08
Student ocenia opłacalność i ograniczenia automatyzacji procesu z wykorzystaniem AI, uwzględniając nakład pracy, dane, ryzyko, jakość i korzyści biznesowe.	K_U09
Student implementuje prototyp rozwiązania AI, np. klasyfikator, model predykcyjny, chatbot, komponent NLP lub system wspomagania decyzji.	K_U12
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny własnej wiedzy, wyników modeli AI, ograniczeń danych oraz treści generowanych przez systemy AI.	K_K01

Student jest gotów do przestrzegania zasad etyki zawodowej, ochrony danych, praw autorskich i odpowiedzialnego korzystania z narzędzi AI.	K_K05
Student jest gotów do samorozwoju w zakresie dynamicznie zmieniających się technologii AI, modeli, narzędzi i regulacji.	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Wykład problemowy
- Pogadanka
- Dyskusja
- Metoda symulacyjna
- Metoda ćwiczeń
- Film dydaktyczny

Metody oceniania:

- Zaliczenie testowe
- Sprawozdanie z realizacji zadania

Wykład konwersatoryjny (W)

1. Wprowadzenie do sztucznej inteligencji. Historia sztucznej inteligencji.
2. Przetwarzanie języka naturalnego oraz wykorzystanie botów.
3. Uczenie maszynowe w biznesie.
4. Robotyzacja procesów
5. Możliwość zastosowania sztucznej inteligencji.
6. Inteligentne systemy wspomagające podejmowanie decyzji.
7. Przyszłość sztucznej inteligencji.

Ćwiczenia (Ć)

1. Podstawy Pythona i biblioteki AI
2. Implementacja algorytmów uczenia nadzorowanego
3. Algorytmy nienadzorowane
4. Przetwarzanie języka naturalnego
5. Wykorzystanie platform do tworzenia chatbotów.
6. Wykorzystanie NLP w chatbotach -Dialog Flow
7. Integracja agenta z rozwiązaniem NLU/NLG
8. Sieci neuronowe w Keras/TensorFlow

Literatura podstawowa

1. Ryszard Tadeusiewicz, Archipelag sztucznej inteligencji, Akademicka Oficyna Wydawnicza Exit, 2021
2. Rishal Hurbans, Algorytmy sztucznej inteligencji. Ilustrowany przewodnik, Helion 2021
3. Hadelin de Ponteves, Sztuczna inteligencja. Błyskawiczne wprowadzenie do uczenia maszynowego, uczenia ze wzmocnieniem i uczenia głębokiego, Helion 2021

Literatura uzupełniająca

1. Rojek Izabela, Wspomaganie procesów podejmowania decyzji i sterowania w systemach o różnej skali złożoności z udziałem metod sztucznej inteligencji, Wydawnictwo Uniwersytetu Kazimierza Wielkiego, 2010
2. Géron, A. Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow. Helion 2020
3. David Foster, Generatywne głębokie uczenie, Promise 2024
4. Daniel Vaughan, Umiejętności analityczne w pracy z danymi i sztuczną inteligencją. Wykorzystywanie najnowszych technologii w rozwijaniu przedsiębiorstwa, Helion 2021

Źródła dodatkowe

1. AI Sweigart, Automatyzacja nudnych zadań z Pythonem. Nauka programowania.
2. Laurence Moroney, Sztuczna inteligencja i uczenie maszynowe dla programistów, Helion 2021
3. Stuart Russell, Peter Norvig. Sztuczna inteligencja. Nowe spojrzenie. Wydanie IV, Helion 2023

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....60 godz.....32 godz.

- Zapoznanie się z literaturą przedmiotu..... 20 godz.....30 godz.
- Przygotowanie się do zaliczenia końcowego..... 10 godz.....18 godz.
- Wypełnienie dokumentacji zajęć..... 10 godz.....20 godz.
- Suma obciążeń:.....100 godz.....100 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen ze wszystkich obowiązkowych form weryfikacji efektów uczenia się określonych w sylabusie, Ocena końcowa wyliczana jest jako średnia ważona z wyniku zaliczenia testowego (40%) oraz złożonej dokumentacji ćwiczeniowej (60%).

Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi:

do 50% - ocena niedostateczna,

51% - 60% - ocena dostateczna,

61% - 70% - ocena dostateczna plus,

71% - 80% - ocena dobra

81% - 90% - ocena dobra plus,

powyżej 90% - ocena bardzo dobra.

Systemy wbudowane

Nazwa zajęć po angielsku: Embedded Systems

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Laboratorium.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Tomasz Brząkała

Opis zajęć:

Przedmiot „Systemy wbudowane” zapoznaje studentów z architekturą, projektowaniem, programowaniem i integracją współczesnych systemów wbudowanych wykorzystywanych w automatyce, Internecie Rzeczy (IoT), robotyce, elektronice użytkowej oraz systemach przemysłowych. Omawiane są zagadnienia związane z architekturą mikrokontrolerów i mikrokomputerów, komunikacją z urządzeniami zewnętrznymi, obsługą czujników i elementów wykonawczych, interfejsami komunikacyjnymi, systemami czasu rzeczywistego, energooszczędnością oraz bezpieczeństwem systemów wbudowanych.

Laboratorium ma charakter praktyczny i realizowane jest z wykorzystaniem trzech klas platform sprzętowych: mikrokontrolerów edukacyjnych, mikrokomputerów jednopłytkowych oraz układów przeznaczonych do zastosowań Internetu Rzeczy z komunikacją bezprzewodową. Dzięki temu studenci poznają różne klasy systemów wbudowanych bez uzależniania programu od konkretnego producenta lub modelu sprzętu.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia zastosowania systemów wbudowanych w informatyce, Internecie Rzeczy (IoT), automatyce domowej, elektronice użytkowej oraz inteligentnych urządzeniach.	K_W01
charakteryzuje architekturę systemów wbudowanych, budowę mikrokontrolerów i mikrokomputerów, organizację pamięci, wejść/wyjść, interfejsów komunikacyjnych oraz podstawowych elementów sprzętowych	K_W04
identyfikuje metody, techniki i narzędzia wykorzystywane do projektowania, programowania, testowania i diagnostyki systemów wbudowanych oraz aplikacji Internetu Rzeczy	K_W05
rozpoznaje zasady projektowania bezpiecznych, energooszczędnych i niezawodnych systemów wbudowanych oraz kierunki ich rozwoju, w tym Edge AI i TinyML	K_W06
Umiejętności (student potrafi)	
pozyskuje i wykorzystuje dokumentację techniczną układów, środowisk programistycznych, bibliotek oraz przykładowych projektów do realizacji zadań laboratoryjnych	K_U01
dobiera platformę sprzętową, elementy wejścia/wyjścia, czujniki, elementy wykonawcze oraz sposób komunikacji odpowiednio do wymagań projektowanego systemu wbudowanego	K_U04
implementuje, uruchamia i konfiguruje program systemu wbudowanego realizujący obsługę wejść i wyjść, komunikację z urządzeniami oraz współpracę z czujnikami i elementami wykonawczymi	K_U05
testuje, analizuje oraz interpretuje wyniki działania systemu wbudowanego, wykorzystując pomiary, komunikaty diagnostyczne oraz obserwację pracy urządzenia	K_U07

projektuje, realizuje i dokumentuje system wbudowany lub aplikację Internetu Rzeczy integrującą czujniki, elementy wykonawcze i komunikację z usługami sieciowymi	K_U11
stosuje dobre praktyki programowania, testowania, dokumentowania oraz zabezpieczania systemów wbudowanych z uwzględnieniem niezawodności, bezpieczeństwa i efektywności energetycznej	K_U12
Kompetencje społeczne (student jest gotów do)	
jest gotów do zasięgania opinii ekspertów podczas rozwiązywania problemów związanych z projektowaniem, programowaniem i integracją systemów wbudowanych	K_K01
jest gotów do myślenia i działania w sposób przedsiębiorczy podczas projektowania rozwiązań wykorzystujących systemy wbudowane oraz Internet Rzeczy	K_K04
jest gotów do przestrzegania zasad etyki zawodowej, bezpieczeństwa oraz ochrony danych podczas projektowania i eksploatacji systemów wbudowanych	K_K05
jest gotów do samodzielnego rozwijania wiedzy i umiejętności w zakresie nowych platform sprzętowych, technologii Internetu Rzeczy oraz narzędzi wykorzystywanych w projektowaniu systemów wbudowanych	U_K07

Metody i formy kształcenia:

- Metoda laboratoryjna
- Wykład problemowy
- Praca ze źródłem elektronicznym
- Studium przypadku
- Metoda pokazu

Metody oceniania:

- Przygotowanie do zajęć - wejściówki
- Zaliczenie testowe
- Sprawozdanie z realizacji zadania

Wykład konwersatoryjny (W)

1. Wprowadzenie do systemów wbudowanych. Definicja, klasyfikacja, obszary zastosowań.
2. Architektura systemów wbudowanych: mikrokontrolery, mikroprocesory i mikrokomputery.
3. Budowa i organizacja mikrokontrolera: CPU, pamięci, magistrale, porty wejścia/wyjścia.
4. Programowanie systemów wbudowanych. Kompilacja, debugowanie, środowiska programistyczne.
5. Obsługa wejść i wyjść cyfrowych oraz analogowych.
6. Czujniki, elementy wykonawcze i podstawowe układy elektroniczne.
7. Interfejsy komunikacyjne: UART, I²C, SPI, CAN oraz komunikacja bezprzewodowa.
8. Systemy czasu rzeczywistego i podstawowe zagadnienia RTOS.
9. Internet Rzeczy (IoT) i komunikacja urządzeń.
10. Bezpieczeństwo systemów wbudowanych.
11. Zarządzanie energią i energooszczędność.
12. Testowanie, diagnostyka i niezawodność systemów wbudowanych.
13. Kierunki rozwoju systemów wbudowanych: Edge AI, TinyML, inteligentne urządzenia.

Laboratorium (L)

1. Konfiguracja środowiska programistycznego oraz zapoznanie z platformami laboratoryjnymi.
2. Programowanie wejść i wyjść cyfrowych.
3. Obsługa sygnałów analogowych.
4. Sterowanie elementami wykonawczymi.
5. Współpraca z czujnikami.
6. Obsługa wyświetlaczy i elementów sygnalizacyjnych.
7. Komunikacja szeregową pomiędzy urządzeniami.
8. Komunikacja z wykorzystaniem magistral sprzętowych.
9. Komunikacja bezprzewodowa urządzeń.
10. Wymiana danych z usługami sieciowymi.

11. Projekt systemu Internetu Rzeczy.
12. Testowanie i diagnostyka opracowanego rozwiązania.
13. Integracja modułów systemu.

Literatura podstawowa

1. M. Banzi, M. Shiloh, Wprowadzenie do Arduino, wyd. II. APN Promise 2022
2. M. Richardson, S. Wallace, W. Donat, Wprowadzenie do Raspberry Pi, wyd. III, APN Promise 2022

Literatura uzupełniająca

1. R. Santos, S. Santos, 20 prostych projektów Raspberry Pi, Wydawnictwo Naukowe PWN 2019
2. I. Culic, A. Radovici, C. Rusu, Komercyjne i przemysłowe aplikacje Internetu rzeczy na Raspberry Pi, APN Promise 2021
3. J. Krawiec, Internet Rzeczy (IoT). Problemy cyberbezpieczeństwa, Oficyna Wydawnicza Politechniki Warszawskiej 2020

Źródła dodatkowe

1. <https://learn.arm.com/learning-paths/embedded-and-microcontrollers/arduino-pico/>
2. <https://www.raspberrypi.com/documentation/>
3. <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>
4. <https://learn.arm.com/learning-paths/embedded-and-microcontrollers/>
5. <https://embed.org.in/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Zapoznanie się z literaturą przedmiotu.....	11 godz.....	18 godz.
• Przygotowanie się do zajęć.....	8 godz.....	14 godz.
• Przygotowanie się do zaliczenia końcowego.....	5 godz.....	9 godz.
• Wypełnienie dokumentacji zajęć.....	6 godz.....	10 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen z obowiązkowych form weryfikacji efektów uczenia się określonych w sylabusie, potwierdzających osiągnięcie zakładanych efektów uczenia się. Na początku zajęć laboratoryjnych studenci piszą krótki sprawdzian (wejściówka) sprawdzający ich przygotowanie do realizacji zajęć na stanowisku laboratoryjnym. Po zajęciach przygotowują dokumentację realizacji zadania praktycznego. Na koniec semestru studenci przystępują do zaliczenia testowego, sprawdzającego opanowanie materiału z wykładu. Ocena końcowa wyliczana jest jako średnia ważona z wyniku zaliczenia testowego (30%) oraz wyników wejściówek i sprawozdań (70%).

Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi:

do 50% - ocena niedostateczna,
 51% - 60% - ocena dostateczna,
 61% - 70% - ocena dostateczna plus,
 71% - 80% - ocena dobra
 81% - 90% - ocena dobra plus,
 powyżej 90% - ocena bardzo dobra.

Problemy społeczne i zawodowe informatyki

Nazwa zajęć po angielsku: Social and professional problems of computer science

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VII

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 1

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

prof. SWPW dr hab. Władysław Błasiak

Opis zajęć:

Przedmiot przygotowuje studentów Informatyki do odpowiedzialnego funkcjonowania w zawodzie informatyka poprzez analizę społecznych, prawnych, etycznych i ekonomicznych uwarunkowań rozwoju technologii informacyjnych. Omawiane są zastosowania informatyki w gospodarce i życiu społecznym, ochrona danych osobowych, prawo autorskie i ochrona oprogramowania, etyka zawodowa, przedsiębiorczość informatyczna, sztuczna inteligencja, cyberuzależnienia oraz kierunki rozwoju technologii. Student analizuje przykłady problemów zawodowych, formułuje stanowisko w dyskusji i przygotowuje pracę pisemną opartą na źródłach.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
charakteryzuje korzyści wynikające z zastosowań informatyki w różnych obszarach działalności człowieka	K_W03
wyjaśnia społeczne skutki rozwoju informatyki, w tym cyberuzależnienia oraz sposoby profilaktyki	K_W07
opisuje prawne, ekonomiczne i etyczne uwarunkowania działalności informatyka, w tym ochronę danych osobowych	K_W08
wyjaśnia podstawowe zasady prawa autorskiego, ochrony oprogramowania i korzystania z utworów w kontekście narzędzi AI	K_W09
opisuje podstawowe etapy zakładania i rozwijania działalności gospodarczej w obszarze usług informatycznych	K_W10
Umiejętności (student potrafi)	
dyskutuje na temat etycznych i pozatechnicznych aspektów prawa autorskiego, ochrony danych i wykorzystania sztucznej inteligencji	K_U08
przedstawia i ocenia stanowiska z przeczytanej pozycji popularnonaukowej lub naukowej dotyczącej najnowszych osiągnięć informatyki	K_U15
Kompetencje społeczne (student jest gotów do)	
jest gotów oceniać wiarygodność informacji o nowych technologiach, AI i skutkach społecznych informatyki	K_K01
jest gotów respektować zasady etyki zawodowej informatyka i odpowiedzialnego korzystania z technologii	K_K05
jest gotów do odpowiedzialnego wykonywania zawodu informatyka z poszanowaniem jego dorobku, dobrych praktyk i zasad odpowiedzialności zawodowej	K_K06

Metody i formy kształcenia:

- Wykład problemowy
- Wykład informacyjny
- Film dydaktyczny

Metody oceniania:

- Praca pisemna

Wykład konwersatoryjny (W)

1. Rola informatyki w rozwoju społecznym i gospodarczym.
2. Zawód informatyka: odpowiedzialność, kompetencje, etyka zawodowa.
3. Podstawy prawne ochrony danych osobowych.
4. Zagrożenia związane z tworzeniem i rozwojem przedsięwzięć informatycznych.
5. Etyka w informatyce
6. Cyberuzależnienia i higiena cyfrowa.
7. Ochrona praw własności intelektualnej, Ochrona oprogramowania.
8. Sztuczna inteligencja. Szanse i zagrożenia. Czy chat AI może być mądrzejszy od ciebie?
9. Wizje przyszłości dla informatyki.

Literatura podstawowa

1. Cieciora M., Wybrane problemy społeczne i zawodowe informatyki, Otwarte Zasoby Edukacyjne p OEZ, Warszawa 2021.

Literatura uzupełniająca

1. Strona internetowa prof. Ryszarda Tadeusiewicza: <https://www.cri.agh.edu.pl/uczelnia/tad/>
2. Ryzyko i skutki uboczne technologii cyfrowej. Wystąpienie prof. Manfreda Spitzera (<https://www.youtube.com/watch?v=9a3YzWBSHvo>)

Źródła dodatkowe

1. Michio Kaku, Wizje, czyli jak nauka zmienia świat w XXI wieku, Prószyński i S-ka, Warszawa 2010

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- | | | |
|---|--------------|----------|
| • Zajęcia z udziałem nauczyciela akademickiego..... | 15 godz..... | 8 godz. |
| • Zapoznanie się z literaturą przedmiotu..... | 6 godz..... | 13 godz. |
| • Przygotowanie się do zajęć..... | 2 godz..... | 2 godz. |
| • Przygotowanie się do zaliczenia końcowego..... | 2 godz..... | 2 godz. |
| Suma obciążeń:..... | 25 godz..... | 25 godz. |

Zaliczenie zajęć:

Zaliczenie końcowe odbędzie się na podstawie pracy pisemnej.

Przykłady pytań:

1. Jakie kroki prawne podejmiesz przy zakładaniu firmy świadczącej usługi internetowe?
2. Jakie są Twoje propozycje zwalczania cyberuzależnień?
3. Podaj kilka przykładów korzyści korzystania chat.gpt.

Praca będzie oceniana w skali ocen szkolnych.

Podstawy telekomunikacji

Nazwa zajęć po angielsku: Fundamentals of telecommunications

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Wojciech Korzybski

Opis zajęć:

W ramach zajęć z podstaw telekomunikacji zdefiniowane i wyjaśnione są podstawowe pojęcia i zagadnienia z dziedziny telekomunikacji. Wykłady stanowią przekaz informacji związanej z przetwarzaniem i przesyłaniem sygnałów oraz ich modulacją. W ramach ćwiczeń studenci są zaznajamiani z narzędziami komputerowymi do wykorzystywania algorytmów obliczeniowych, pozwalających na znajdowanie rozwiązań przykładowych zagadnień telekomunikacyjnych i interpretację uzyskiwanych wyników.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia matematyczny opis sygnału analogowego i cyfrowego oraz widma częstotliwościowego w typowych zadaniach informatycznych	K_W01
charakteryzuje proces transmisji w mediach przewodowych, bezprzewodowych oraz systemach sieciowych i teleinformatycznych	K_W04
identyfikuje metody, techniki i narzędzia stosowane do analizy, symulacji i interpretacji prostych problemów transmisyjnych	K_W05
rozpoznaje podstawowe standardy i normy techniczne dotyczące transmisji danych, protokołów oraz sieci bezprzewodowych	K_W06
<i>Umiejętności (student potrafi)</i>	
stosuje narzędzia informacyjno-komunikacyjne do reprezentacji, wizualizacji i analizy sygnałów	K_U04
formułuje i rozwiązuje praktyczny problem doboru medium transmisyjnego, modulacji lub parametrów sygnału dla prostego scenariusza teleinformatycznego	K_U05
wykorzystuje metody analityczne, symulacyjne oraz eksperymentalne do badania modulacji, demodulacji, widma i wpływu zakłóceń	K_U07
<i>Kompetencje społeczne (student jest gotów do)</i>	
jest gotów oceniać wiarygodność źródeł oraz własnych wyników obliczeń i symulacji w obszarze telekomunikacji	K_K01
jest gotów w przedsiębiorczy sposób wykorzystać poznane narzędzia do zastosowania praktycznego w telekomunikacji	K_K04

Metody i formy kształcenia:

- Praca ze źródłem elektronicznym
- Metoda ćwiczeń
- Metoda pokazu
- Wykład problemowy

Metody oceniania:

- Zaliczenie testowe
- Aktywność w trakcie zajęć
- Zaliczenie praktyczne

Wykład konwersatoryjny (W)

1. Wprowadzenie do telekomunikacji i etapy rozwoju telekomunikacji
2. Model systemu telekomunikacyjnego oraz podstawowe pojęcia z zakresu telekomunikacji
3. Klasyfikacja i właściwości sygnałów telekomunikacyjnych
4. Przetwarzanie sygnałów i widma częstotliwościowe
5. Podstawy matematyczne z zakresu teorii sygnałów i ich przetwarzania
6. Zakłócenia i zniekształcenia sygnałów oraz metody redukcji szumów
7. Media transmisyjne
8. Rodzaje modulacji w telekomunikacji

Ćwiczenia (Ć)

1. Środowisko obliczeniowe Scilab - podstawowe komendy i interfejs programu
2. Operacje na wektorach i macierzach
3. Graficzna prezentacja wyników w formie wykresów 2D i 3D
4. Podstawy programowania i zapis skryptów w Scilabie
5. Podstawowe funkcje i ich wykorzystanie w telekomunikacji
6. Sposoby zapisu sygnałów analogowych i cyfrowych
7. Przykłady modulacji i demodulacji w telekomunikacji
8. Widma częstotliwościowe i ich zapis

Literatura podstawowa

1. Richard Read. Telekomunikacja. – WKŁ, Warszawa, 2000.

Literatura uzupełniająca

1. Simon Haykin. Systemy telekomunikacyjne 1,2. - WKŁ, Warszawa, 2004.
2. Bogumiła Mrozek, Zbigniew Mrozek. MATLAB i Simulink, poradnik użytkownika. HELION, Gliwice, 2014.
3. Bogdan Galwas, Podstawy telekomunikacji optofalowej, Warszawska Wyższa Szkoła Informatyki, 2018

Źródła dodatkowe

1. SciLab Tutorials - <https://www.scilab.org/tutorials>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Zapoznanie się z literaturą przedmiotu.....	15 godz.....	15 godz.
• Przygotowanie się do zajęć.....	10 godz.....	15 godz.
• Przygotowanie się do zaliczenia końcowego.....	5 godz.....	21 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych wyników ze wszystkich obowiązkowych form weryfikacji efektów uczenia się. Ocena końcowa wyliczana jest jako średnia z wykonanych zadań praktycznych na zajęciach ćwiczeniowych (50%) oraz z wyniku zaliczenia testowego (50%).

Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi:

do 50% - ocena niedostateczna,

51% - 60% - ocena dostateczna,

61% - 70% - ocena dostateczna plus,

71% - 80% - ocena dobra

81% - 90% - ocena dobra plus,
powyżej 90% - ocena bardzo dobra.
Na ocenę końcową może mieć wpływ aktywność w trakcie zajęć.

Język Python

Nazwa zajęć po angielsku: Programming language Python

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Tomasz Brząkała

Opis zajęć:

Przedmiot obejmuje praktyczne zastosowanie języka Python do rozwiązywania zadań informatycznych, automatyzacji pracy, przetwarzania danych, pracy z plikami, korzystania z bibliotek, budowy prostych programów narzędziowych oraz testowania kodu. Student poznaje składnię i idiomy Pythona, środowiska wirtualne, strukturę projektu, obsługę wyjątków, moduły, pakiety, podstawy typowania, testy jednostkowe oraz dobre praktyki formatowania i dokumentowania kodu. Zajęcia przygotowują do wykorzystania Pythona w dalszych obszarach kształcenia, w tym analizie danych, sztucznej inteligencji, automatyzacji, administracji systemami i prototypowaniu aplikacji.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia praktyczne zastosowania języka Python w automatyzacji, analizie danych, przetwarzaniu plików, prototypowaniu i rozwiązywaniu zadań informatycznych	K_W03
identyfikuje metody, techniki i narzędzia programowania w Pythonie, w tym interpreter, środowisko IDE, moduły, pakiety, venv, pip, testy, typowanie i dokumentację	K_W05
rozpoznaje standardy i dobre praktyki programowania w Pythonie, formatowanie kodu, strukturę projektu, środowiska wirtualne, testowanie i zarządzanie zależnościami	K_W06
<i>Umiejętności (student potrafi)</i>	
pozyskuje i wykorzystuje dokumentację Pythona, dokumentację bibliotek oraz źródła elektroniczne do rozwiązania problemu programistycznego	K_U01
dobiera struktury danych, biblioteki, środowisko pracy i narzędzia pomocnicze do wymagań zadania programistycznego w Pythonie	K_U04
rozwiązuje praktyczne problemy programistyczne z użyciem funkcji, struktur danych, modułów, plików, wyjątków, bibliotek i testów	K_U05
projektuje i realizuje program lub skrypt narzędziowy w Pythonie, dokumentując sposób uruchomienia, zależności i wyniki działania	K_U11
stosuje dobre praktyki programowania, testowania, formatowania, obsługi błędów i pracy z zależnościami w projekcie Pythonowym	K_U12
<i>Kompetencje społeczne (student jest gotów do)</i>	
jest gotów do krytycznej oceny jakości własnego kodu, poprawności rozwiązania, czytelności, testowalności i doboru bibliotek	K_K01
jest gotów do zasięgania opinii ekspertów i korzystania z dokumentacji w przypadku	K_K02

trudności z rozwiązaniem problemu programistycznego	
jest gotów do samorozwoju w zakresie nowych wersji Pythona, bibliotek, narzędzi, standardów kodowania i zastosowań języka w praktyce informatycznej	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Metoda ćwiczeń
- Metoda sytuacyjna
- Dyskusja

Metody oceniania:

- Zaliczenie testowe
- Zaliczenie praktyczne

Wykład konwersatoryjny (W)

1. Python we współczesnej informatyce: automatyzacja, data science, AI, web, testowanie, DevOps.
2. Instalacja, wersje Pythona, interpreter, REPL, VS Code/PyCharm, uruchamianie skryptów.
3. Składnia, typy danych, operatory, instrukcje warunkowe, pętle.
4. Funkcje, argumenty, zakresy nazw, dokumentacja funkcji, docstringi.
5. Struktury danych: listy, krotki, słowniki, zbiory, comprehensions.
6. Łańcuchy tekstowe, f-stringi, formatowanie, wyrażenia regularne — podstawy.
7. Obsługa plików: tekst, CSV, JSON, ścieżki, pathlib.
8. Moduły, pakiety, biblioteka standardowa, importy i struktura projektu.
9. Wyjątki i obsługa błędów.
10. Podstawy programowania obiektowego w Pythonie.
11. Środowiska wirtualne, pip, pyproject.toml, zależności.
12. Styl kodu: PEP 8, formatowanie, linting, czytelność.
13. Testowanie: unittest lub pytest, testy funkcji, przypadki brzegowe.
14. Podstawy typowania: adnotacje typów, typing, czytelność i utrzymanie kodu.

Ćwiczenia (Ć)

1. Konfiguracja środowiska, repozytorium i pierwsze skrypty.
2. Zadania z instrukcji warunkowych i iteracyjnych.
3. Funkcje i testowanie prostych funkcji.
4. Listy, słowniki, zbiory i comprehensions.
5. Przetwarzanie tekstu i walidacja danych wejściowych.
6. Pliki tekstowe, CSV i JSON.
7. Obsługa wyjątków i przypadków brzegowych.
8. Moduły, pakiety i organizacja kodu.
9. Klasy i modele obiektowe.
10. Środowisko venv, instalacja pakietów i praca z zależnościami.
11. Testy jednostkowe w pytest lub unittest.

Literatura podstawowa

1. Mertz D., Lepszy kod w Pythonie. Przewodnik dla aspirujących. APN Promise 2024
2. Linux Magazine. Python Nauka programowania dla każdego. Wiedza i Praktyka 2022

Literatura uzupełniająca

1. Martelli A, Martelli Ravenscroft A., Ho Mcguire S., Python w pigułce. APN Promise 2023
2. Sarbicki G., Python. Kurs dla nauczycieli i studentów. Helion 2019
3. Kierzkowski A., Gawryszewski M., Python. Ćwiczenia praktyczne. Helion 2017

Źródła dodatkowe

1. <https://docs.python.org/>
2. <https://packaging.python.org/en/latest/>
3. <https://docs.pytest.org/en/stable/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	16 godz.
• Samodzielne wykonanie zadań.....	10 godz.....	19 godz.
• Przygotowanie się do zaliczenia końcowego.....	10 godz.....	16 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen ze wszystkich obowiązkowych form weryfikacji efektów uczenia się określonych w sylabusie, potwierdzających osiągnięcie zakładanych efektów uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych.

Ocena końcowa wyliczana jest jako średnia ważona z wyniku zaliczenia testowego (40%) oraz zaliczenia ćwiczeń praktycznych (60%).

Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi:

do 50% - ocena niedostateczna,

51% - 60% - ocena dostateczna,

61% - 70% - ocena dostateczna plus,

71% - 80% - ocena dobra

81% - 90% - ocena dobra plus,

powyżej 90% - ocena bardzo dobra.

Zarządzanie procesami i systemami IT

Nazwa zajęć po angielsku: IT Process and System Management

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Mariusz Sochacki

Opis zajęć:

Przedmiot „Zarządzanie procesami i systemami IT” obejmuje zagadnienia związane z planowaniem, wdrażaniem i zarządzaniem systemami informatycznymi w organizacjach. Celem przedmiotu jest przekazanie studentom wiedzy o zarządzaniu usługami IT, procesach biznesowych, optymalizacji infrastruktury IT, a także o narzędziach i frameworkach wspierających te procesy, takich jak ITIL, COBIT czy DevOps.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Student wyjaśnia zastosowania zarządzania procesami, usługami i systemami IT w organizacji oraz relację między usługą IT, procesem biznesowym, infrastrukturą i wartością dla użytkownika.	K_W03
Student charakteryzuje metody, techniki i narzędzia ITSM/ITOM/DevOps, w tym ITIL, COBIT, BPMN, CMDB, monitoring, CI/CD i narzędzia zarządzania zgłoszeniami.	K_W05
Student rozpoznaje standardy, ramy i dobre praktyki zarządzania usługami i procesami IT, w tym ITIL 4, COBIT 2019, ISO/IEC 20000, BPMN 2.0.2, DORA metrics i praktyki DevOps.	K_W06
Umiejętności (student potrafi)	
Student pozyskuje i wykorzystuje dokumentację frameworków, narzędzi i standardów do rozwiązania problemu zarządzania usługą lub procesem IT.	K_U01
Student dobiera narzędzia do modelowania procesów, monitorowania infrastruktury, obsługi zgłoszeń, automatyzacji wdrożeń i raportowania usług IT.	K_U04
Student modeluje proces biznesowy lub proces IT w BPMN oraz proponuje usprawnienia na podstawie wybranych kryteriów jakości, kosztu, czasu lub ryzyka.	K_U05
Student ocenia opłacalność i ograniczenia usprawnienia procesu IT, automatyzacji wdrożenia lub wdrożenia narzędzia monitorowania.	K_U09
Student planuje i organizuje pracę własną lub zespołową przy przygotowaniu modelu procesu, karty usługi, raportu monitoringu lub prostego pipeline CI/CD.	K_U17
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do zasięgania opinii ekspertów przy złożonych problemach związanych z procesami, narzędziami, jakością usług lub bezpieczeństwem systemów IT.	K_K02
Student jest gotów do przestrzegania zasad etyki zawodowej, jakości, bezpieczeństwa,	K_K05

Metody i formy kształcenia:

- Wykład informacyjny
- Praca ze źródłem elektronicznym
- Studium przypadku
- Metoda ćwiczeń
- Wykład problemowy

Metody oceniania:

- Zaliczenie testowe
- Sprawozdanie z realizacji zadania

Wykład konwersatoryjny (W)

1. Wprowadzenie do zarządzania systemami IT
2. Zarządzanie usługami IT (IT Service Management)
3. Procesy biznesowe i ich integracja z systemami IT
4. Zarządzanie infrastrukturą IT
5. Frameworki zarządzania IT: COBIT i DevOps
6. Zarządzanie bezpieczeństwem systemów IT
7. Monitoring i raportowanie w systemach IT

Ćwiczenia (Ć)

1. Narzędzia do zarządzania infrastrukturą IT
2. Zarządzanie procesami IT przy użyciu ITIL
3. Modelowanie procesów biznesowych z BPMN
4. Modelowanie procesów biznesowych z BPMN - wykorzystanie narzędzi do BPMN
5. Modelowanie procesów biznesowych z BPMN - studium przypadku
6. Wirtualizacja i zarządzanie chmurą obliczeniową
7. DevOps: Continuous Integration i Continuous Deployment (CI/CD)

Literatura podstawowa

1. Jan van Bon, ITIL® 4 – A Pocket Guide, 2019
2. Winkler T. J., Wulf J., Effectiveness of IT Service Management Capability Value Co-Creation and Value Facilitation Mechanisms, 2019
3. Zrozumieć BPMN. Modelowanie procesów biznesowych. Wydanie 2 rozszerzone, Onepress, 2017

Literatura uzupełniająca

1. COBIT 2019 Framework: Governance and Management Objectives, Isaca, 2018
2. DevOps: A Software Architect's Perspective, Len Bass, Ingo Weber, Liming Zhu, Addison-Wesley Professional 2015

Źródła dodatkowe

1. OMG Standards Development Organization - <https://www.omg.org/bpmn/>
2. People CERT - peoplecert.org
3. ISACA - <https://www.isaca.org/resources/cobit>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Zapoznanie się z literaturą przedmiotu.....	15 godz.....	20 godz.
• Przygotowanie się do zajęć.....	5 godz.....	10 godz.
• Przygotowanie się do zaliczenia końcowego.....	5 godz.....	10 godz.
• Wypełnienie dokumentacji zajęć.....	5 godz.....	11 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen ze wszystkich obowiązkowych form weryfikacji efektów uczenia się określonych w sylabusie, Ocena końcowa wyliczana jest jako średnia ważona z wyniku zaliczenia testowego (40%) oraz złożonego sprawozdania z realizacji zadania ćwiczeniowego (60%).
Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi:

do 50% - ocena niedostateczna,
51% - 60% - ocena dostateczna,
61% - 70% - ocena dostateczna plus,
71% - 80% - ocena dobra
81% - 90% - ocena dobra plus,
powyżej 90% - ocena bardzo dobra.

Wirtualizacja systemów informatycznych

Nazwa zajęć po angielsku: Virtualization of IT systems

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Andrzej Ulicki

Opis zajęć:

Przedmiot obejmuje praktyczne podstawy projektowania, uruchamiania, zabezpieczania i utrzymywania środowisk wirtualnych oraz kontenerowych. Studenci poznają różnice między wirtualizacją maszyn, konteneryzacją i orkiestracją, konfiguruje maszyny wirtualne, sieci, wolumeny, kontenery, podstawowe mechanizmy wysokiej dostępności oraz narzędzia automatyzacji. Celem zajęć jest przygotowanie studentów do pracy z infrastrukturą serwerową, środowiskami laboratoryjnymi, chmurą prywatną/publiczną i narzędziami DevOps, z uwzględnieniem bezpieczeństwa, monitorowania, dokumentowania i optymalizacji zasobów.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia zastosowania wirtualizacji, konteneryzacji i orkiestracji w systemach informatycznych, środowiskach serwerowych, chmurowych i DevOps	K_W03
charakteryzuje procesy działania hypervisora, maszyny wirtualnej, obrazu kontenera, wolumenu, sieci wirtualnej i klastra kontenerowego	K_W04
identyfikuje metody, techniki i narzędzia wirtualizacji, konteneryzacji, automatyzacji, monitorowania i zabezpieczania środowisk infrastrukturalnych	K_W05
rozpoznaje standardy i dobre praktyki dotyczące obrazów, rejestrów, sieci, pamięci masowej, bezpieczeństwa, backupu i utrzymania środowisk wirtualnych oraz kontenerowych	K_W06
Umiejętności (student potrafi)	
pozyskuje i wykorzystuje dokumentację platform wirtualizacji, Docker, Kubernetes i narzędzi automatyzacji do rozwiązania problemu infrastrukturalnego	K_U01
dobiera narzędzia do tworzenia, uruchamiania, monitorowania, automatyzowania i zabezpieczania maszyn wirtualnych oraz kontenerów	K_U04
tworzy i konfiguruje maszyny wirtualne, sieci wirtualne, kontenery, wolumeny oraz podstawowe mechanizmy wysokiej dostępności w środowisku laboratoryjnym	K_U05
projektuje i realizuje środowisko uruchomieniowe aplikacji z użyciem wirtualizacji lub konteneryzacji oraz dokumentuje jego konfigurację	K_U11
stosuje dobre praktyki bezpieczeństwa, automatyzacji, monitorowania, backupu i utrzymania środowisk wirtualnych oraz kontenerowych	K_U12
stosuje rozwiązania i dobre praktyki wykorzystywane w środowisku zawodowym do utrzymania i administrowania zwirtualizowanymi systemami komputerowymi	K_U13

<i>Kompetencje społeczne (student jest gotów do)</i>	
jest gotów do krytycznej oceny własnej konfiguracji, użytych obrazów, poziomu bezpieczeństwa, kosztu zasobów i ograniczeń środowiska wirtualnego	K_K01
jest gotów do zasięgania opinii ekspertów w przypadku złożonych problemów infrastrukturalnych, bezpieczeństwa lub automatyzacji	K_K02
jest gotów do samorozwoju w zakresie dynamicznie zmieniających się technologii wirtualizacji, konteneryzacji, chmury i automatyzacji infrastruktury	U_K07

Metody i formy kształcenia:

- Wykład problemowy
- Dyskusja
- Studium przypadku
- Metoda ćwiczeń
- Metoda laboratoryjna

Metody oceniania:

- Zaliczenie testowe
- Sprawozdanie z realizacji zadania

Wykład konwersatoryjny (W)

1. Wprowadzenie do wirtualizacji
2. Wirtualizacja zasobów obliczeniowych
3. Rozproszone systemy pamięci masowej
4. Wprowadzenie do konteneryzacji
5. Systemy zarządzania kontenerami
6. Automatyzacja systemów wirtualizacji i konteneryzacji

Ćwiczenia (Ć)

1. Wirtualizacja (instalacja i konfiguracja platformy wirtualizacyjnej w środowisku laboratoryjnym)
2. Zarządzanie mechanizmami wysokiej dostępności (konfiguracja mechanizmów zapewniających ciągłość działania, usług wolumenów oraz podstaw backupu i odtwarzania)
3. Zarządzanie siecią (konfiguracja sieci w środowisku kontenerowym oraz testowanie komunikacji pomiędzy maszynami wirtualnymi i kontenerami)
4. Rozproszone systemy pamięci masowej (konfiguracja wolumenów wykorzystywanych przez maszyny wirtualne oraz trwałego przechowywania danych w środowisku kontenerowym)
5. Konteneryzacja (Przygotowanie obrazu kontenera i pliku Dockerfile, wykorzystaniem Compose, monitoring i analiza logów środowiska kontenerowego, podstawy orkiestracji kontenerów)

Literatura podstawowa

1. Sean Kane, Karl Matthias, Docker. Niezawodne kontenery produkcyjne. Praktyczne zastosowania. Wydanie III, Helion 2014
2. Marek Serafin, Wirtualizacja w praktyce, Helion 2011
3. Gigi Sayfan, Tajniki Kubernetes, Promise 2021

Literatura uzupełniająca

1. He Kun Yuan, Linux KVM virtualization architecture practical guide, 2017
2. Alan Hohn, Kubernetes. Przewodnik po orkiestracji kontenerów i tworzeniu niezawodnych aplikacji, Helion 2024

Źródła dodatkowe

1. James Turnbull, The Docker Book: Containerization is the new virtualization (<https://dockerbook.com/>)
2. Nigel Poulton, The Kubernetes Book (<https://www.nigelpoulton.com/books>)
3. Exploring KVM Virtualization with Real World Scenarios (<https://linuxgd.medium.com/exploring-kvm-virtualization-with-real-world-scenarios-070abbd14f76>)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....45 godz.....24 godz.
- Zapoznanie się z literaturą przedmiotu.....10 godz.....15 godz.
- Przygotowanie się do zajęć.....5 godz.....10 godz.
- Wypełnienie dokumentacji zajęć.....10 godz.....15 godz.

- Przygotowanie się do zaliczenia końcowego..... 5 godz.....11 godz.
Suma obciążeń:.....75 godz.....75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskania ponad 50% poprawnych odpowiedzi na teście obejmującym informacje teoretyczne oraz uzyskanie co najmniej 50% punktów z realizacji ćwiczeń praktycznych.

Podstawy programowania robotów

Nazwa zajęć po angielsku: Basics of robot programming

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Krzysztof Obst

Opis zajęć:

Przedmiot przeznaczony jest dla studentów kierunku Informatyka, którzy ukończyli już zajęcia z programowania, elektrotechniki oraz automatyki. W ramach wykładu studenci poznają podstawowe zagadnienia związane z robotyką, w tym budowę, działanie oraz zastosowania robotów w różnych dziedzinach przemysłu. Na zajęciach ćwiczeniowych zyskują podstawowe umiejętności programowania układów sterujących oraz algorytmy wykorzystywane w robotyce. Szczegółowo studenci poznają robota przemysłowego KUKA.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
charakteryzuje budowę, zasadę działania i zastosowania robotów przemysłowych oraz elementów stanowiska zrobotyzowanego: manipulatora, układu sterowania, sensorów, elementów wykonawczych i układów bezpieczeństwa	K_W03
wyjaśnia zasady wykorzystania układów współrzędnych, definiowania narzędzia i bazy oraz wyznaczania punktów i toru ruchu manipulatora przemysłowego	K_W04
identyfikuje metody, techniki i narzędzia wykorzystywane do programowania, konfiguracji, uruchamiania, testowania i diagnostyki robotów przemysłowych	K_W05
rozpoznaje zasady i wymagania dotyczące bezpiecznej eksploatacji stanowiska zrobotyzowanego oraz dobre praktyki programowania i uruchamiania robotów przemysłowych	K_W06
Umiejętności (student potrafi)	
pozyskuje i wykorzystuje informacje z dokumentacji technicznej robota, układu sterowania oraz elementów stanowiska zrobotyzowanego do realizacji zadania praktycznego	K_U01
dobiera tryb pracy, układ współrzędnych, parametry ruchu oraz narzędzia programistyczne odpowiednio do realizowanego zadania robotycznego	K_U04
konfiguruje układ współrzędnych narzędzia i bazy oraz wyznacza punkty niezbędne do realizacji zadanej trajektorii ruchu robota	K_U05
testuje, analizuje i diagnozuje działanie programu robota na podstawie obserwacji przebiegu ruchu, komunikatów układu sterowania i zachowania elementów stanowiska	K_U07
tworzy, uruchamia i modyfikuje program sterujący robotem przemysłowym w celu wykonania określonego zadania technologicznego	K_U11
projektuje sposób realizacji zadania przez robota przemysłowego, uwzględniając kolejność	K_U12

operacji, trajektorię ruchu, parametry pracy, elementy stanowiska i wymagania bezpieczeństwa	
Kompetencje społeczne (student jest gotów do)	
jest gotów do krytycznej oceny własnej wiedzy i przygotowanego rozwiązania oraz do konsultowania problemów związanych z programowaniem, konfiguracją i bezpieczeństwem robota, gdy wymagają wiedzy wykraczającej poza jego kompetencje	K_K01
jest gotów do przestrzegania zasad odpowiedzialnego i bezpiecznego programowania oraz eksploatacji systemów robotycznych, z uwzględnieniem konsekwencji działania urządzeń dla człowieka i otoczenia	K_K05

Metody i formy kształcenia:

- Wykład informacyjny
- Dyskusja
- Studium przypadku
- Metoda ćwiczeń
- Metoda pokazu

Metody oceniania:

- Zaliczenie testowe
- Zaliczenie praktyczne

Wykład konwersatoryjny (W)

1. Wprowadzenie do robotyki (historia, klasyfikacja, zastosowania)
2. Budowa i działanie robotów (układy sterujące, sensory, siłowniki, kinematyka i dynamika ruchu),
3. Podstawy programowania robotów (struktura programów sterujących)
4. Algorytmy sterowania robotów (PID i jego zastosowania, algorytmy nawigacji i omijania przeszkód)
5. Robotyka przemysłowa (ramię robotyczne, programowanie i kontrola robotów w liniach produkcyjnych)
6. Bezpieczeństwo i etyka w robotyce (wpływ robotów na społeczeństwo i rynek pracy)

Ćwiczenia (Ć)

1. Wprowadzenie do narzędzi programowania robotów przemysłowych
2. Budowa i funkcjonowanie układów sterowania robotem
3. Układy współrzędnych i metody przemieszczeń ramienia robota
4. Tryby pracy robota i układy zabezpieczeń
5. Uruchamianie programu i tworzenie projektów
6. Projektowanie stanowiska zrobotyzowanego
7. Uruchamianie i edycja programów. Obsługa programatora ręcznego

Literatura podstawowa

1. P. Husbands, Roboty. Co każdy powinien wiedzieć. Wydawnictwo Naukowe PWN 2023
2. P. Dutkiewicz, W. Wróblewski, K. Kozłowski, Modelowanie i sterowanie robotów. PWN 2005
3. C. Hughes, T. Hughes, Programowanie robotów. Sterowanie pracą robotów autonomicznych. Helion 2017

Literatura uzupełniająca

1. J. Kacprzyk (red.), Innowacyjne rozwiązania w obszarze automatyki, robotyki i pomiarów. PIAP 2013
2. P. Ciesielski, J. Sawoniewicz, A. Szmigielski, Elementy robotyki mobilnej. Wydawnictwo PJATK 2004
3. D. Cook, Budowa robotów dla początkujących. Apres 2012

Źródła dodatkowe

1. Robot (V)KR 150, 180, 210, 240, 270-2. Instrukcja montażu, KUKA
2. Systemy programowania robotów przemysłowych - obsługa i podstawy programowania robotów KUKA
3. FanucRobotics - kurs podstawowy obsługi i programowania sterownika
4. Kawasaki - instrukcja obsługi i programowania robota

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....45 godz.....24 godz.
- Zapoznanie się z literaturą przedmiotu.....10 godz.....10 godz.

- Samodzielne wykonanie zadań.....10 godz.....25 godz.
- Przygotowanie się do kolokwium.....10 godz.....16 godz.
- Suma obciążeń:.....75 godz.....75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia zajęć jest uzyskanie pozytywnego wyniku z zaliczenia testowego obejmującego zakres treści wykładowych oraz z zaliczenia praktycznego obejmującego zakres ćwiczeń praktycznych. Ocena końcowa wyliczana jest jako średnia ważona w 40% z wyniku kursu oraz w 60% z wykonania ćwiczeń praktycznych i wynosi:

0% - 50% - niedostateczna
 51%-60% - dostateczna
 61%-70% - dostateczna plus
 71%-80% - dobra
 81%-90% - dobra plus
 91%-100% - bardzo dobra.

Grywalizacja, czyli zastosowanie mechanizmów gier w informatyce

Nazwa zajęć po angielsku: Gamification: Applying Game Mechanics in Computer Science

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot „Grywalizacja, czyli zastosowanie mechanizmów gier w informatyce” poświęcony jest interdyscyplinarnemu zastosowaniu mechanizmów znanych z gier komputerowych w projektowaniu systemów informatycznych. Studenci zapoznają się z teorią grywalizacji, typologią graczy, elementami gier (mechanika, dynamika, komponenty), a także z praktycznym wykorzystaniem tych narzędzi w projektowaniu aplikacji. Szczególny nacisk położony jest na analizę potrzeb użytkownika, projektowanie rozwiązań sprzyjających zaangażowaniu oraz na tworzenie i prezentację koncepcji aplikacji grywalizacyjnych.

Zajęcia rozwijają umiejętności praktyczne i kreatywność, a także wspierają kształtowanie postaw ukierunkowanych na innowacyjność i etyczne wykorzystanie technologii. Studenci zyskują umiejętności samodzielnego projektowania aplikacji i usług IT, które skutecznie angażują użytkowników i wywołują długofalowe zmiany ich zachowań. Nabyte umiejętności zwiększają atrakcyjność absolwenta na rynku pracy.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
identyfikuje mechanizmy rządzące grami i potrafi wykorzystać stosowane tam zasady, metody i techniki podczas projektowania i budowania aplikacji informatycznych	K_W05
wyjaśnia społeczne konsekwencje rozwoju technologii ze szczególnym uwzględnieniem wykorzystania informatycznych aplikacji grywalizacyjnych w zmianie złych nawyków użytkowników	K_W07
charakteryzuje ekonomiczne i etyczne uwarunkowania wstawiania elementów grywalizacji do aplikacji użytkowych	K_W08
łączy wiedzę z zakresu grywalizacji z informatyczną wiedzą o UX/HCI, projektowania UI do projektowania zmiany zachowań użytkowników	U_W11
<i>Umiejętności (student potrafi)</i>	
integruje umiejętności informatyczną z wiedzą o motywacji, typach użytkowników i mechanizmach gier w projektowaniu i budowaniu aplikacji	K_U02
dobiera elementy grywalizacji z zakresu mechaniki, dynamiki oraz właściwe komponenty i wzorce interakcji do realizacji zadania informatycznego	K_U04
uwzględnia pozatechniczne aspekty grywalizacji (autonomia, prywatność, możliwość manipulacji i wykluczenia)	K_U08
planuje pracę nad zaplanowaniem wykorzystania grywalizacji w aplikacjach użytkowych	K_U17
<i>Kompetencje społeczne (student jest gotów do)</i>	

jest gotów do inicjowania działań mających na celu zmianę zachowania użytkowników na korzystniejszą dla nich samych oraz ich środowiska lokalnego, wykorzystując mechanizmy grywalizacji	K_K03
jest gotów do przestrzegania zasad etycznego projektowania systemów angażujących użytkowników	K_K05
jest gotów do samorozwoju w zakresie nowych metod grywalizacji, HCI, UX, badań użytkowników, UI i odpowiedzialnego projektowania zgrywalizowanych aplikacji	U_K07

Metody i formy kształcenia:

- Dyskusja
- Praca ze źródłem elektronicznym
- Studium przypadku
- Metoda ćwiczeń
- Wykład informacyjny

Metody oceniania:

- Ocena aktywności w kursie
- Sprawozdanie z realizacji zadania
- Zaliczenie testowe

Wykład konwersatoryjny (W)

1. Wprowadzenie do grywalizacji: pojęcia i konteksty
2. Geneza i psychologia gier: motywacja, flow, zaangażowanie
3. Typologie graczy: Richard Bartle i Andrzej Marczewski
4. Elementy grywalizacji: mechanika, dynamika, komponenty
5. Pętle zaangażowania i postępu: analiza mechanizmów
6. Zastosowania grywalizacji w informatyce
7. Projektowanie systemów informatycznych z elementami grywalizacji

Ćwiczenia (Ć)

1. Analiza przykładowej aplikacji grywalizacyjnej: siłownia miejska „Pod chmurką”
2. Określanie komponentów gier: punkty, odznaki, poziomy, rankingi
3. Projektowanie mechaniki gry: wyzwania, nagrody, współpraca
4. Budowanie dynamiki gry: emocje, narracja, relacje, ograniczenia
5. Dobór elementów gry do typów graczy
6. Projektowanie scenariusza interakcji użytkownika z aplikacją
7. Prototypowanie rozwiązania
8. Testowanie koncepcji

Literatura podstawowa

1. kurs e-learningowy na platformie WLODEK
2. J. Siadkowski, Grywalizacja. Zrób to sam. Poradnik. Fundacja Orange, Fundacja Highlight/Inaczej. 2014 (<https://www.sbc.org.pl/en/dlibra/publication/122525>)
3. J. Swacha: Grywalizacja w nauce programowania – platforma FGPE. UNIWERSYTET SZCZECIŃSKI 2023 (<https://wnus.usz.edu.pl/ris/pl/issue/1385/>)

Literatura uzupełniająca

1. P. Tkaczyk: Grywalizacja. Jak zastosować mechanizmy gier w działaniach marketingowych. One Press 2012
2. G. Zichermann, C. Cynningham: Grywalizacja. Mechanika gry na stronach WWW i w aplikacjach mobilnych. Helion 2012
3. A. Para: Możliwości wykorzystania grywalizacji w zdalnej edukacji. e-mentor, nr 1 (88)/2021
4. J. Korzeniewska: Ograć nawyki. Jak zastosować mechanizmy gier w procesie zmiany osobistej. One Press 2016
5. M. Reda, A. Cybulski, K. Przybylski: Osiągnij przewagę w biznesie dzięki grywalizacji. Gamfi 2015

Źródła dodatkowe

1. <https://www.gamified.uk>
2. <https://fundacja.orange.pl/strefa-wiedzy/poradniki>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Realizacja kursu e-learningowego.....	10 godz.....	13 godz.
• Zapoznanie się z literaturą przedmiotu.....	2 godz.....	8 godz.
• Wypełnienie dokumentacji zajęć.....	15 godz.....	25 godz.
• Przygotowanie się do zaliczenia końcowego.....	3 godz.....	5 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen ze wszystkich obowiązkowych form weryfikacji efektów uczenia się. W trakcie semestru studenci realizują materiały udostępniane na platformie WŁODEK. Każda lekcja zawiera materiały dydaktyczne oraz pytania kontrolne sprawdzające stopień opanowania omawianych zagadnień. W ramach zajęć studenci wykonują również zadania projektowe i ćwiczeniowe, których rozwiązania przekazują za pośrednictwem platformy lub podczas zajęć. Na zakończenie semestru odbywa się zaliczenie testowe obejmujące treści teoretyczne oraz ocena kompletnej dokumentacji ćwiczeniowej przygotowanej w trakcie realizacji przedmiotu.

Ocena końcowa wyliczana jest jako średnia ważona z pytań kontrolnych umieszczonych w lekcjach (20%), z wyniku zaliczenia testowego (30%), dokumentacji ćwiczeniowej (50%).

Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi:

do 50% - ocena niedostateczna,

51% - 60% - ocena dostateczna,

61% - 70% - ocena dostateczna plus,

71% - 80% - ocena dobra

81% - 90% - ocena dobra plus,

powyżej 90% - ocena bardzo dobra.

Administracja serwerami internetowymi

Nazwa zajęć po angielsku: Web server administration

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: kurs e-learningowy

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Maciej Krawczyk

Opis zajęć:

W ramach przedmiotu student uzyskuje wiedzę z zakresu samodzielnej administracji serwerami internetowymi. Poznaje techniki zdalnego dostępu do serwerów, wirtualizacji środowiska wykonawczego, określania ilości potrzebnych zasobów i ich dynamicznej alokacji. Ponadto poznaje podstawowe protokoły sieciowe wykorzystywane przez serwery internetowe (DHCP, DNS, HTTP(s), inne), oprogramowanie obsługujące w/w protokoły i metody konfiguracji tego oprogramowania. Podczas ćwiczeń laboratoryjnych nabywa umiejętność prowadzenia dokumentacji technicznej.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia cykl życia serwera internetowego: wybór systemu, instalację, konfigurację usług, zabezpieczenie, monitorowanie, aktualizację, backup i utrzymanie	K_W04
charakteryzuje metody, techniki i narzędzia administrowania systemem Linux, usługami WWW, DNS, HTTPS, kontenerami, logami i usługami systemowymi	K_W05
rozpoznaje standardy, protokoły i normy techniczne związane z usługami internetowymi, w tym DNS, HTTP/HTTPS, TLS, SSH i podstawowe standardy bezpieczeństwa serwerowego	K_W06
<i>Umiejętności (student potrafi)</i>	
dobiera narzędzia administracyjne do instalacji, konfiguracji, monitorowania i zabezpieczenia serwera internetowego	K_U04
instaluje, konfiguruje i zarządza podstawowymi usługami serwera internetowego, w tym DNS, WWW i HTTPS	K_U05
planuje i przeprowadza test działania usługi serwerowej, interpretuje wyniki oraz formułuje wnioski diagnostyczne	K_U06
rozwiązuje praktyczne zadania administracyjne, stosując właściwe technologie, dokumentację techniczną, standardy i dobre praktyki bezpieczeństwa	K_U12
wykorzystuje procedury utrzymania systemów informatycznych do diagnozowania awarii, analizy logów i przywracania działania usług serwerowych	K_U13
dokumentuje czynności administracyjne i posługuje się specjalistyczną terminologią w komunikacji technicznej	K_U14
<i>Kompetencje społeczne (student jest gotów do)</i>	
jest gotów krytycznie oceniać własną wiedzę, konfiguracje serwera i informacje pozyskane z	K_K01

dokumentacji oraz Internetu	
jest gotów zasięgać opinii ekspertów oraz korzystać z dokumentacji producenta w przypadku trudności z samodzielnym rozwiązaniem problemu administracyjnego	K_K02
jest gotów rozwijać kompetencje administratora serwerów z wykorzystaniem dokumentacji, kursów, źródeł elektronicznych i nowoczesnych technologii	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Praca ze źródłem elektronicznym
- Wykład problemowy
- Metoda pokazu
- Metoda laboratoryjna

Metody oceniania:

- Zaliczenie testowe
- Sprawozdanie z realizacji zadania

Wykład konwersatoryjny (W)

1. Cele zarządzania serwerami Internetowymi
2. Dobór i przygotowanie systemu operacyjnego dla serwera internetowego
3. Organizacja pamięci masowej serwera: systemy plików, macierze, kopie zapasowe i niezawodność
4. Emulacja, wirtualizacja, hipernadzorcy
5. Sieć: interfejsy, protokoły, usługi, domeny

Ćwiczenia (Ć)

1. Instalacja systemu operacyjnego
2. Instalacja i konfiguracja serwera DNS
3. Instalacja i konfiguracja serwera WWW
4. Włączenie i konfiguracja protokołu HTTPS

Literatura podstawowa

1. Dokumentacja techniczna online systemu operacyjnego (UBUNTU/FreeBSD/Windows/inny...)
2. Dokumentacja techniczna online usług internetowych (Apache/Nginx/DNS/itd.)
3. Dokumentacja techniczna online Certbot/Let's Encrypt
4. OWASP Top 10:2025

Literatura uzupełniająca

1. Aileen Frisch, Unix Administracja systemu, Oficyna Wydawnicza READ ME
2. W. Stallings, Organizacja i architektura systemu komputerowego Tom 2 - Wydanie XI, WNT

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....45 godz.....24 godz.
- Zapoznanie się z literaturą przedmiotu.....10 godz.....10 godz.
- Przygotowanie się do zajęć.....3 godz.....8 godz.
- Wypełnienie dokumentacji zajęć.....6 godz.....12 godz.
- Przygotowanie się do zaliczenia końcowego.....3 godz.....6 godz.
- Samodzielne wykonanie zadań.....8 godz.....15 godz.
- Suma obciążeń:.....75 godz.....75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia wykładu jest rozwiązywanie testów sprawdzających na platformie "Wlodek". Testy są udostępniane po każdym wykładzie. Testy są dostępne przez okres 7 dni. Odpowiedzi są oceniane w skali punktowej zależnej od wagi pytania. Obowiązuje następująca skala ocen dotycząca procentowej sumy punktów ze wszystkich testów:

- 51-60% - ocena dostateczna
- 61-70% - ocena dostateczna plus
- 71-80% - ocena dobra
- 81-90% - ocena dobra plus
- 91-100% - ocena bardzo dobra.

Warunkiem zaliczenia ćwiczeń laboratoryjnych jest przesłanie na platformę "Wlodek" sprawozdania z czynności

wykonywanych podczas zajęć. Sprawozdanie musi zawierać uzasadnienie dla wykonywanych czynności i podejmowanych decyzji. Sprawozdania są oceniane w skali punktowej od 0 do 5. Punkty są przeliczane na ocenę w sposób analogiczny jak w przypadku wykładów.

Warunkiem zaliczenia przedmiotu jest uzyskanie oceny pozytywnej z wykładu i z ćwiczeń laboratoryjnych. Na ocenę końcową składają się w równym stopniu ocena z wykładu i ocena z ćwiczeń laboratoryjnych.

Komputerowa grafika przestrzenna

Nazwa zajęć po angielsku: 3D computer graphics

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Wojciech Korzybski

Opis zajęć:

W ramach zajęć studenci są zapoznani z metodami zapisu obiektów geometrycznych na płaszczyźnie i w przestrzeni. Podczas zajęć komputerowych wykorzystywane są różne techniki modelowania cyfrowego oraz sposoby modyfikacji obiektów. Po zrealizowaniu części teoretycznej i praktycznej przedmiotu studenci nabędą wiedzę i umiejętności niezbędne do komputerowego przedstawienia obiektów rzeczywistych i wirtualnych.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
charakteryzuje zależności matematyczne wykorzystywane do opisu punktów, wektorów, przekształceń i obiektów geometrycznych w grafice 3D	K_W01
wyjaśnia zagadnienia z zakresu geometrii analitycznej oraz omawia metody i zasady graficznego zapisu obiektów płaskich i przestrzennych	K_W02
identyfikuje metody, techniki i narzędzia stosowane w modelowaniu, renderingu i wymianie danych 3D	K_W05
<i>Umiejętności (student potrafi)</i>	
wyszukuje i wykorzystuje informacje pozyskane z różnych źródeł do realizacji zadań z zakresu grafiki przestrzennej	K_U01
dobiera źródła, dokumentację i materiały techniczne potrzebne do rozwiązania zadania z zakresu grafiki 3D	K_U03
stosuje narzędzia grafiki 3D do tworzenia, modyfikacji i eksportu prostych modeli przestrzennych	K_U04
wykorzystuje poznane metody i techniki do tworzenia modeli obiektów geometrycznych	K_U07
rozwiązuje praktyczne zadanie inżynierskie związane z przygotowaniem modelu 3D do wybranego zastosowania informatycznego	K_U12
<i>Kompetencje społeczne (student jest gotów do)</i>	
weryfikuje aktualność własnej wiedzy o narzędziach, formatach i technologiach grafiki 3D	K_K01
ma potrzebę samodzielnego i dodatkowego uczenia się w celu aktualizacji swojej wiedzy i umiejętności oraz dzielenia się pozyskiwaną informacją z innymi	U_K07

Metody i formy kształcenia:

- Praca ze źródłem elektronicznym

Metody oceniania:

- Zaliczenie testowe

- Metoda ćwiczeń
- Wykład problemowy
- Zaliczenie praktyczne
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

1. Metody odwzorowywania obiektów
2. Graficzny zapis obiektów geometrycznych
3. Podstawowe przekształcenia geometryczne
4. Podstawy modelowania komputerowego
5. Techniki modelowania obiektów
6. Sposoby przekształcania rysunków płaskich 2D w modele przestrzenne 3D
7. Formaty zapisu i wymiany danych graficznych

Ćwiczenia (Ć)

1. Praktyczne sposoby reprezentacji obiektów graficznych
2. Podstawowe obiekty geometryczne i ich zapis w systemach komputerowych
3. Metody modyfikacji obiektów
4. Tworzenie obiektów symetrycznych i powtarzalnych
5. Przekształcenia geometryczne
6. Modelowanie przestrzenne 3D
7. Tworzenie obiektów przestrzennych z rysunków 2D
8. Modelowanie bryłowe

Literatura podstawowa

1. P. Kiciak, Podstawy modelowania krzywych i powierzchni, Wydawnictwo Naukowe PWN, 2019
2. Jankowski M., Elementy grafiki komputerowej, WNT, Warszawa 2006
3. Von Glitschka, Grafika wektorowa. Szkolenie podstawowe. Wydanie II, Helion 2012

Literatura uzupełniająca

1. Kiciak P., Grafika komputerowa - Portal z materiałami dydaktycznymi Wydziału Matematyki, Informatyki i Mechaniki UW (MIMUW), 2011

Źródła dodatkowe

1. K. Elam, Wprowadzenie do projektowania przestrzennego. Zasady, procesy i projekty, d2d 2021

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- | | | |
|---|--------------|----------|
| • Zajęcia z udziałem nauczyciela akademickiego..... | 45 godz..... | 24 godz. |
| • Zapoznanie się z literaturą przedmiotu..... | 15 godz..... | 25 godz. |
| • Samodzielne wykonanie zadań..... | 10 godz..... | 18 godz. |
| • Przygotowanie się do zaliczenia końcowego..... | 5 godz..... | 8 godz. |
| Suma obciążeń:..... | 75 godz..... | 75 godz. |

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnej oceny z testu z zakresu teorii prezentowanej na wykładzie oraz poprawne wykonanie zadania zaliczeniowego z części praktycznej zajęć. Ocena końcowa wyliczana jest jako średnia arytmetyczna z obu ocen. Na podwyższenie oceny końcowej może mieć wpływ aktywność na zajęciach.

Internet rzeczy

Nazwa zajęć po angielsku: Internet of things, Io T

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia kierunkowe

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Tomasz Brząkała

Opis zajęć:

Przedmiot "Internet rzeczy" przeznaczony jest dla studentów kierunku Informatyka, którzy potrafią programować w językach obiektowych Java, Object Pascal, C++ lub Python. Celem przedmiotu jest zapoznanie studentów z metodą programowania komputerów jednopłytkowych w oparciu o dystrybucję systemu linuksowego - Raspberry Pi OS na platformie komputera jednopłytkowego Raspberry Pi, wyjaśnienie koncepcji zastosowań komputerów jednopłytkowych jako identyfikowalnych przedmiotów mogących pośrednio albo bezpośrednio gromadzić, przetwarzać lub wymieniać dane za pomocą sieci komputerowej oraz wykształcenie umiejętności opracowywania aplikacji wykorzystujących komputery jednopłytkowe w oparciu o język programowania Python. Część praktyczna przedmiotu nastawiona jest na opracowywanie aplikacji wykorzystujących możliwości i zalety komputerów jednopłytkowych (np. odczyt danych z czujników zewnętrznych, rejestracja danych, integracja z systemem inteligentnego domu), obsługę podstawowych magistrali komunikacyjnych z urządzeniami zewnętrznymi oraz współpracę ze źródłami danych (pliki, bazy danych). Po ukończeniu kursu studenci będą znali podstawy programowania komputerów jednopłytkowych Raspberry Pi w systemie Raspberry Pi OS, będą potrafili zaprojektować i wykonać aplikację przeznaczoną na platformę Raspberry Pi z zastosowaniem języka programowania Python.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia zastosowania Internetu rzeczy w systemach informatycznych, automatyce, inteligentnym domu, monitoringu środowiskowym, usługach cyfrowych i rozwiązaniach edge/cloud	K_W03
charakteryzuje procesy zachodzące w systemie IoT: pozyskanie danych z czujnika, przetwarzanie na urządzeniu brzegowym, transmisję, magazynowanie, wizualizację i reakcję systemu	K_W04
identyfikuje metody, techniki i narzędzia IoT, w tym Raspberry Pi, mikrokontrolery, Python, GPIO, I2C, SPI, OneWire, MQTT, Node-RED, bazy danych i dashboardy	K_W05
rozpoznaje standardy i dobre praktyki IoT, w tym MQTT, HTTP/REST, podstawy Thread/Matter, ETSI EN 303 645, NISTIR 8259A, bezpieczną konfigurację urządzeń i ochronę danych	K_W06
<i>Umiejętności (student potrafi)</i>	
pozyskuje i wykorzystuje dokumentację Raspberry Pi, Pythona, bibliotek sprzętowych, MQTT, czujników i baz danych do rozwiązania problemu IoT	K_U01
dobiera urządzenie, czujnik, magistralę komunikacyjną, protokół, bazę danych i narzędzie wizualizacji do wymagań prostego systemu IoT	K_U04
projektuje i realizuje system IoT obejmujący czujnik, urządzenie brzegowe, program w	K_U11

Pythonie, transmisję danych, zapis i wizualizację	
stosuje właściwe technologie i dobre praktyki przy rozwiązywaniu zadania IoT, w tym obsługę GPIO, I2C/SPI/OneWire, MQTT, bazę danych, logowanie, dokumentację i podstawy bezpieczeństwa	K_U12
komunikuje konfigurację, architekturę, wyniki pomiarów i ograniczenia systemu IoT z użyciem specjalistycznej terminologii technicznej	K_U14
planuje i organizuje pracę własną lub zespołową nad prototypem IoT, dokumentując zadania, konfigurację, kod, testy i wyniki	K_U17
Kompetencje społeczne (student jest gotów do)	
jest gotów do krytycznej oceny jakości, wiarygodności, bezpieczeństwa i ograniczeń danych oraz rozwiązań IoT	K_K01
jest gotów do zasięgania opinii ekspertów w przypadku problemów sprzętowych, sieciowych, bezpieczeństwa lub integracji systemu IoT	K_K02
jest gotów do przestrzegania zasad etyki, prywatności, bezpieczeństwa i odpowiedzialnego przetwarzania danych zbieranych przez urządzenia IoT	K_K05
jest gotów do samorozwoju w zakresie nowych platform, protokołów, usług chmurowych, standardów bezpieczeństwa i narzędzi IoT	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Studium przypadku
- Metoda ćwiczeń
- Metoda pokazu
- Metoda sytuacyjna
- Dyskusja

Metody oceniania:

- Zaliczenie testowe
- Przygotowanie do zajęć - wejściówki
- Karta wykonania zadania

Wykład konwersatoryjny (W)

1. IoT jako architektura: urządzenie, czujnik, edge, sieć, broker, baza danych, dashboard, chmura.
2. Raspberry Pi i alternatywy: Raspberry Pi Zero 2 W, Raspberry Pi Pico/RP2040, ESP32 — dobór platformy.
3. Raspberry Pi OS, konfiguracja headless, SSH, aktualizacje, użytkownicy, podstawy hardeningu.
4. Python w IoT: środowisko, biblioteki, obsługa błędów, logowanie, usługi systemowe.
5. Interfejsy sprzętowe: GPIO, PWM, I2C, SPI, UART/RS232, OneWire.
6. Sensory i aktyatory: BME280, DS18B20, OLED/TFT, przekaźniki, silniki, RTC.
7. Komunikacja IoT: MQTT 5, HTTP/REST, WebSocket, podstawy BLE, LoRaWAN, Thread/Matter — przegląd.
8. MQTT w praktyce: broker, topic, QoS, retain, last will, TLS, uwierzytelnianie.
9. Dane IoT: pliki, SQLite, bazy szeregów czasowych, retencja danych, znaczniki czasu, jakość pomiarów.
10. Wizualizacja: Node-RED, Grafana albo dashboard WWW
11. Integracja smart home: Home Assistant, MQTT, Matter/Thread — poziom przeglądowy.
12. Bezpieczeństwo IoT: ETSI EN 303 645, NISTIR 8259A, aktualizacje, sekrety, certyfikaty, prywatność, segmentacja sieci.
13. Testowanie i utrzymanie: diagnostyka, logi, watchdog, monitoring, backup konfiguracji, dokumentacja.

Ćwiczenia (Ć)

1. Uruchomienie Raspberry Pi OS, konfiguracja sieci i SSH.
2. Konfiguracja środowiska Python, repozytorium i struktury projektu.
3. GPIO: przycisk, LED, przekaźnik, debouncing.
4. I2C: OLED albo BME280.
5. SPI: wyświetlacz TFT albo inny moduł.
6. OneWire: DS18B20 i pomiar temperatury.
7. PWM: sterowanie silnikiem DC lub jasnością LED.

8. UART/RS232: komunikacja z PC lub modulem zewnętrznym.
9. MQTT: publikacja i subskrypcja danych, broker Mosquitto, QoS, struktura topiców.
10. Zapis danych: SQLite albo baza time-series.
11. Dashboard: Node-RED/Grafana/prosta strona WWW
12. Bezpieczeństwo: zmiana haseł, klucze SSH, TLS dla MQTT, ochrona sekretów, minimalizacja uprawnień.
13. Mini-projekt: kolektor danych z czujnika + publikacja MQTT + zapis + wizualizacja + dokumentacja + test awarii.

Literatura podstawowa

1. D. Staple, Robotyka w domu przy użyciu Raspberry Pi Pico, APN Promise, 2023
2. M. Sikorski, Internet Rzeczy, Wydawnictwo Naukowe PWN, 2019

Literatura uzupełniająca

1. I. Culic, A. Radovici, C. Rusu, Komercyjne i przemysłowe aplikacje Internetu rzeczy na Raspberry Pi, APN Promise 2021
2. M. Richardson, S. Wallace, W. Donat, Wprowadzenie do Raspberry Pi, wyd. III, APN Promise 2022
3. R. Santos, S. Santos, 20 prostych projektów Raspberry Pi, Wydawnictwo Naukowe PWN 2019

Źródła dodatkowe

1. <https://www.raspberrypi.org/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Zapoznanie się z literaturą przedmiotu.....	9 godz.....	12 godz.
• Wypełnienie dokumentacji zajęć.....	14 godz.....	25 godz.
• Przygotowanie się do zaliczenia końcowego.....	7 godz.....	14 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen ze wszystkich obowiązkowych form weryfikacji efektów uczenia się określonych w sylabusie, potwierdzających osiągnięcie zakładanych efektów uczenia się. W ramach procesu weryfikacji oceniane są efekty uczenia się w zakresie wiedzy na temat systemów opartych na komputerach jednopłytkowych, zdobycie umiejętności projektowania i wykonania aplikacji funkcjonującej na urządzeniach platformy Raspberry Pi oraz nabycie kompetencji polegających na grupowym rozwiązywaniu problemów zawodowych. Na koniec semestru studenci przystępują do zaliczenia testowego, sprawdzającego opanowanie materiału z wykładu. Zaliczenie ćwiczeń polega na przygotowaniu dokumentacji oraz zaprezentowaniu wykonanego zespołowego zadania.

Praktyka zawodowa I

Nazwa zajęć po angielsku: Professional practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: III

Forma zaliczenia: Zaliczenie bez oceny

Punkty ECTS: 6

Materiały: brak

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Dodatkowe zajęcia wynikające z organizacji studiów

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Praktyka.....160 / 160 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Praktyka zawodowa na kierunku Informatyka ma na celu połączenie wiedzy teoretycznej zdobytej podczas studiów z praktycznym doświadczeniem zawodowym. Dzięki praktykom studenci rozwijają umiejętności inżynierskie i kompetencje społeczne, poznają specyfikę środowiska zawodowego, zasady organizacji pracy oraz realia rynku pracy. W trakcie praktyk mają możliwość doskonalenia umiejętności rozwiązywania rzeczywistych zadań inżynierskich, organizacji pracy własnej i zespołowej, a także zapoznania się z technologicznym i informatycznym wyposażeniem podmiotu. Celem nadrzędnym jest przygotowanie studentów do efektywnego wykonywania zawodu informatyka poprzez praktyczne zastosowanie wiedzy i rozwój kompetencji wymaganych przez pracodawców.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
zna wybrane metody, technologie i narzędzia stosowane w praktyce zawodowej informatyków oraz rozumie ich praktyczne zastosowanie	K_W05
zna standardy i dobre praktyki stosowane w środowiskach produkcyjnych w branży IT	K_W06
ma wiedzę o podmiocie, w którym odbywa praktykę, w tym o stosowanych procedurach, metodach organizacji pracy, zasadach bezpieczeństwa i etyki zawodowej	K_W08
<i>Umiejętności (student potrafi)</i>	
potrafi samodzielnie uczyć się i rozwijać swoje kompetencje zawodowe, korzystając z dostępnych źródeł wiedzy, technologii oraz wskazówek opiekuna praktyk (przełożonego)	K_U01
potrafi identyfikować i rozwiązywać praktyczne problemy techniczne i organizacyjne występujące w środowisku pracy informatyka, wykorzystując zdobytą wiedzę i doświadczenie	K_U02
potrafi stosować odpowiednie metody, technologie, narzędzia i standardy branżowe do rozwiązywania i realizacji zadań zleconych przez opiekuna praktyk (przełożonego) w miejscu pracy	K_U12
potrafi skutecznie komunikować się w środowisku zawodowym z wykorzystaniem specjalistycznej terminologii i odpowiednich narzędzi komunikacyjnych	K_U14
potrafi planować i organizować swoją pracę, uwzględniając cele, terminy oraz zasady obowiązujące w środowisku pracy	K_U17
<i>Kompetencje społeczne (student jest gotów do)</i>	
jest gotów do krytycznej oceny własnych działań w środowisku pracy oraz przyjmowania konstruktywnej krytyki od współpracowników i przełożonych	K_K01
jest gotów do rozwiązywania problemów zawodowych w sposób odpowiedzialny i	K_K02

samodzielny, a w przypadku trudności potrafi skutecznie współpracować z zespołem lub zasięgać opinii ekspertów	
jest gotów do ciągłego uczenia się i rozwijania swoich kompetencji społecznych i zawodowych, w celu lepszego dostosowania się do wymagań dynamicznie zmieniającego się środowiska IT	U_K07

Metody i formy kształcenia:

- Pogadanka
- Metoda sytuacyjna
- Studium przypadku
- Metoda pokazu
- Metoda ćwiczeń

Metody oceniania:

- Rozliczenie praktyki

Praktyka (Pr)

1. Zapoznanie z przepisami i zasadami BHP, ochrony danych osobowych (RODO), tajemnicy służbowej, cyberbezpieczeństwa, archiwizowania danych firmowych i innymi regulacjami określającymi funkcjonowanie podmiotu.
2. Poznanie struktury organizacyjnej podmiotu, zakresu działalności, sposobu zarządzania i środowiska zawodowego. Zapoznanie się z przepływem informacji oraz rodzajami i obiegiem dokumentów w instytucji.
3. Obserwacja pracy podmiotu / działu IT. Poznanie procesów i wykorzystywanych metodologii pracy, zapoznanie z używanymi w podmiocie standardami, technologiami i narzędziami.
4. Realizacja, pod nadzorem opiekuna praktyki, zadań praktycznych wybranych z katalogu rekomendowanych zagadnień.

Literatura podstawowa

1. Zarządzenie nr 11/2023 Rektora SWPW z dnia 5 lipca 2023 r. w sprawie Uczelnianego Regulaminu Studenckich Praktyk Zawodowych obowiązującego studentów Szkoły Wyższej im. Pawła Włodkowica w Płocku
2. Uchwała 15/2019 Senatu SWPW w Płocku z dnia 30 września 2019 r. w sprawie ustalenia Zasad, warunków i trybu zaliczania w SWPW praktyki zawodowej poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego
3. Katalog rekomendowanych zagadnień do realizacji na praktyce zawodowej na kierunku Informatyka

Literatura uzupełniająca

1. Zarządzenie nr 06/2023 Rektora SWPW w Płocku z dnia 13 kwietnia 2023 r. w sprawie realizacji zadań Uczelnianego Pełnomocnika ds. praktyk w Szkole Wyższej im. Pawła Włodkowica w Płocku
2. Zarządzenie nr 13/2024 Rektora SWPW z dnia 22 maja 2024 r. w sprawie wprowadzenia "Zasad oceny realizacji praktyk przez studentów SWPW w Płocku"

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....160 godz.....160 godz.
Suma obciążeń:.....160 godz.....160 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia praktyki jest właściwe zrealizowanie zadań praktycznych w wybranym podmiocie oraz uzyskanie efektów uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych. Potwierdzeniem tego jest wypełniony dziennik praktyk z oceną uzyskania efektów uczenia się przez przedstawiciela podmiotu, w którym realizowano praktykę oraz jego opinią o studencie. Ważnym elementem procesu zaliczenia są wnioski studenta z realizacji praktyki, samoocena działań, realizacji zadań i postawy społecznej w trakcie praktyki oraz uwagi dotyczące realizacji praktyki zawodowej.

W przypadku wniosku o zaliczenie praktyki poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego, wpływ na decyzję ma przedstawiona dokumentacja potwierdzająca uzyskanie efektów uczenia się w pracy zawodowej, argumentacja studenta zawarta we wniosku oraz opinia przełożonego.

Praktyka zawodowa II

Nazwa zajęć po angielsku: Professional practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: IV

Forma zaliczenia: Zaliczenie bez oceny

Punkty ECTS: 6

Materiały: brak

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Dodatkowe zajęcia wynikające z organizacji studiów

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Praktyka.....160 / 160 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Praktyka zawodowa na kierunku Informatyka ma na celu połączenie wiedzy teoretycznej zdobytej podczas studiów z praktycznym doświadczeniem zawodowym. Dzięki praktykom studenci rozwijają umiejętności inżynierskie i kompetencje społeczne, poznają specyfikę środowiska zawodowego, zasady organizacji pracy oraz realia rynku pracy. W trakcie praktyk mają możliwość doskonalenia umiejętności rozwiązywania rzeczywistych zadań inżynierskich, organizacji pracy własnej i zespołowej, a także zapoznania się z technologicznym i informatycznym wyposażeniem podmiotu. Celem nadrzędnym jest przygotowanie studentów do efektywnego wykonywania zawodu informatyka poprzez praktyczne zastosowanie wiedzy i rozwój kompetencji wymaganych przez pracodawców.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
zna wybrane metody, technologie i narzędzia stosowane w praktyce zawodowej informatyków oraz rozumie ich praktyczne zastosowanie	K_W05
zna standardy i dobre praktyki stosowane w środowiskach produkcyjnych w branży IT	K_W06
ma wiedzę o podmiocie, w którym odbywa praktykę, w tym o stosowanych procedurach, metodach organizacji pracy, zasadach bezpieczeństwa i etyki zawodowej	K_W08
Umiejętności (student potrafi)	
potrafi samodzielnie uczyć się i rozwijać swoje kompetencje zawodowe, korzystając z dostępnych źródeł wiedzy, technologii oraz wskazówek opiekuna praktyk (przełożonego)	K_U01
potrafi identyfikować i rozwiązywać praktyczne problemy techniczne i organizacyjne występujące w środowisku pracy informatyka, wykorzystując zdobytą wiedzę i doświadczenie	K_U02
potrafi stosować odpowiednie metody, technologie, narzędzia i standardy branżowe do rozwiązywania i realizacji zadań zleconych przez opiekuna praktyk (przełożonego) w miejscu pracy	K_U12
potrafi skutecznie komunikować się w środowisku zawodowym z wykorzystaniem specjalistycznej terminologii i odpowiednich narzędzi komunikacyjnych	K_U14
potrafi planować i organizować swoją pracę, uwzględniając cele, terminy oraz zasady obowiązujące w środowisku pracy	K_U17
Kompetencje społeczne (student jest gotów do)	
jest gotów do krytycznej oceny własnych działań w środowisku pracy oraz przyjmowania konstruktywnej krytyki od współpracowników i przełożonych	K_K01
jest gotów do rozwiązywania problemów zawodowych w sposób odpowiedzialny i	K_K02

samodzielny, a w przypadku trudności potrafi skutecznie współpracować z zespołem lub zasięgać opinii ekspertów	
jest gotów do ciągłego uczenia się i rozwijania swoich kompetencji społecznych i zawodowych, w celu lepszego dostosowania się do wymagań dynamicznie zmieniającego się środowiska IT	U_K07

Metody i formy kształcenia:

- Pogadanka
- Metoda sytuacyjna
- Studium przypadku
- Metoda pokazu
- Metoda ćwiczeń

Metody oceniania:

- Rozliczenie praktyki

Praktyka (Pr)

1. Zapoznanie z przepisami i zasadami BHP, ochrony danych osobowych (RODO), tajemnicy służbowej, cyberbezpieczeństwa, archiwizowania danych firmowych i innymi regulacjami określającymi funkcjonowanie podmiotu.
2. Poznanie struktury organizacyjnej podmiotu, zakresu działalności, sposobu zarządzania i środowiska zawodowego. Zapoznanie się z przepływem informacji oraz rodzajami i obiegiem dokumentów w instytucji.
3. Obserwacja pracy podmiotu / działu IT. Poznanie procesów i wykorzystywanych metodologii pracy, zapoznanie z używanymi w podmiocie standardami, technologiami i narzędziami.
4. Realizacja, pod nadzorem opiekuna praktyki, zadań praktycznych wybranych z katalogu rekomendowanych zagadnień.

Literatura podstawowa

1. Zarządzenie nr 11/2023 Rektora SWPW z dnia 5 lipca 2023 r. w sprawie Uczelnianego Regulaminu Studenckich Praktyk Zawodowych obowiązującego studentów Szkoły Wyższej im. Pawła Włodkowica w Płocku
2. Uchwała 15/2019 Senatu SWPW w Płocku z dnia 30 września 2019 r. w sprawie ustalenia Zasad, warunków i trybu zaliczania w SWPW praktyki zawodowej poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego
3. Katalog rekomendowanych zagadnień do realizacji na praktyce zawodowej na kierunku Informatyka

Literatura uzupełniająca

1. Zarządzenie nr 06/2023 Rektora SWPW w Płocku z dnia 13 kwietnia 2023 r. w sprawie realizacji zadań Uczelnianego Pełnomocnika ds. praktyk w Szkole Wyższej im. Pawła Włodkowica w Płocku
2. Zarządzenie nr 13/2024 Rektora SWPW z dnia 22 maja 2024 r. w sprawie wprowadzenia "Zasad oceny realizacji praktyk przez studentów SWPW w Płocku"

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....160 godz.....160 godz.
Suma obciążeń:.....160 godz.....160 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia praktyki jest właściwe zrealizowanie zadań praktycznych w wybranym podmiocie oraz uzyskanie efektów uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych. Potwierdzeniem tego jest wypełniony dziennik praktyk z oceną uzyskania efektów uczenia się przez przedstawiciela podmiotu, w którym realizowano praktykę oraz jego opinią o studencie. Ważnym elementem procesu zaliczenia są wnioski studenta z realizacji praktyki, samoocena działań, realizacji zadań i postawy społecznej w trakcie praktyki oraz uwagi dotyczące realizacji praktyki zawodowej.

W przypadku wniosku o zaliczenie praktyki poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego, wpływ na decyzję ma przedstawiona dokumentacja potwierdzająca uzyskanie efektów uczenia się w pracy zawodowej, argumentacja studenta zawarta we wniosku oraz opinia przełożonego.

Praktyka zawodowa III

Nazwa zajęć po angielsku: Professional practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie bez oceny

Punkty ECTS: 6

Materiały: brak

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Dodatkowe zajęcia wynikające z organizacji studiów

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Praktyka.....160 / 160 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Praktyka zawodowa na kierunku Informatyka ma na celu połączenie wiedzy teoretycznej zdobytej podczas studiów z praktycznym doświadczeniem zawodowym. Dzięki praktykom studenci rozwijają umiejętności inżynierskie i kompetencje społeczne, poznają specyfikę środowiska zawodowego, zasady organizacji pracy oraz realia rynku pracy. W trakcie praktyk mają możliwość doskonalenia umiejętności rozwiązywania rzeczywistych zadań inżynierskich, organizacji pracy własnej i zespołowej, a także zapoznania się z technologicznym i informatycznym wyposażeniem podmiotu. Celem nadrzędnym jest przygotowanie studentów do efektywnego wykonywania zawodu informatyka poprzez praktyczne zastosowanie wiedzy i rozwój kompetencji wymaganych przez pracodawców.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
zna wybrane metody, technologie i narzędzia stosowane w praktyce zawodowej informatyków oraz rozumie ich praktyczne zastosowanie	K_W05
zna standardy i dobre praktyki stosowane w środowiskach produkcyjnych w branży IT	K_W06
ma wiedzę o podmiocie, w którym odbywa praktykę, w tym o stosowanych procedurach, metodach organizacji pracy, zasadach bezpieczeństwa i etyki zawodowej	K_W08
Umiejętności (student potrafi)	
potrafi samodzielnie uczyć się i rozwijać swoje kompetencje zawodowe, korzystając z dostępnych źródeł wiedzy, technologii oraz wskazówek opiekuna praktyk (przełożonego)	K_U01
potrafi identyfikować i rozwiązywać praktyczne problemy techniczne i organizacyjne występujące w środowisku pracy informatyka, wykorzystując zdobytą wiedzę i doświadczenie	K_U02
potrafi stosować odpowiednie metody, technologie, narzędzia i standardy branżowe do rozwiązywania i realizacji zadań zleconych przez opiekuna praktyk (przełożonego) w miejscu pracy	K_U12
potrafi skutecznie komunikować się w środowisku zawodowym z wykorzystaniem specjalistycznej terminologii i odpowiednich narzędzi komunikacyjnych	K_U14
potrafi planować i organizować swoją pracę, uwzględniając cele, terminy oraz zasady obowiązujące w środowisku pracy	K_U17
Kompetencje społeczne (student jest gotów do)	
jest gotów do krytycznej oceny własnych działań w środowisku pracy oraz przyjmowania konstruktywnej krytyki od współpracowników i przełożonych	K_K01
jest gotów do rozwiązywania problemów zawodowych w sposób odpowiedzialny i	K_K02

samodzielny, a w przypadku trudności potrafi skutecznie współpracować z zespołem lub zasięgać opinii ekspertów	
jest gotów do ciągłego uczenia się i rozwijania swoich kompetencji społecznych i zawodowych, w celu lepszego dostosowania się do wymagań dynamicznie zmieniającego się środowiska IT	U_K07

Metody i formy kształcenia:

- Pogadanka
- Metoda sytuacyjna
- Studium przypadku
- Metoda pokazu
- Metoda ćwiczeń

Metody oceniania:

- Rozliczenie praktyki

Praktyka (Pr)

1. Zapoznanie z przepisami i zasadami BHP, ochrony danych osobowych (RODO), tajemnicy służbowej, cyberbezpieczeństwa, archiwizowania danych firmowych i innymi regulacjami określającymi funkcjonowanie podmiotu.
2. Poznanie struktury organizacyjnej podmiotu, zakresu działalności, sposobu zarządzania i środowiska zawodowego. Zapoznanie się z przepływem informacji oraz rodzajami i obiegiem dokumentów w instytucji.
3. Obserwacja pracy podmiotu / działu IT. Poznanie procesów i wykorzystywanych metodologii pracy, zapoznanie z używanymi w podmiocie standardami, technologiami i narzędziami.
4. Realizacja, pod nadzorem opiekuna praktyki, zadań praktycznych wybranych z katalogu rekomendowanych zagadnień.

Literatura podstawowa

1. Zarządzenie nr 11/2023 Rektora SWPW z dnia 5 lipca 2023 r. w sprawie Uczelnianego Regulaminu Studenckich Praktyk Zawodowych obowiązującego studentów Szkoły Wyższej im. Pawła Włodkowica w Płocku
2. Uchwała 15/2019 Senatu SWPW w Płocku z dnia 30 września 2019 r. w sprawie ustalenia Zasad, warunków i trybu zaliczania w SWPW praktyki zawodowej poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego
3. Katalog rekomendowanych zagadnień do realizacji na praktyce zawodowej na kierunku Informatyka

Literatura uzupełniająca

1. Zarządzenie nr 06/2023 Rektora SWPW w Płocku z dnia 13 kwietnia 2023 r. w sprawie realizacji zadań Uczelnianego Pełnomocnika ds. praktyk w Szkole Wyższej im. Pawła Włodkowica w Płocku
2. Zarządzenie nr 13/2024 Rektora SWPW z dnia 22 maja 2024 r. w sprawie wprowadzenia "Zasad oceny realizacji praktyk przez studentów SWPW w Płocku"

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....160 godz.....160 godz.
Suma obciążeń:.....160 godz.....160 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia praktyki jest właściwe zrealizowanie zadań praktycznych w wybranym podmiocie oraz uzyskanie efektów uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych. Potwierdzeniem tego jest wypełniony dziennik praktyk z oceną uzyskania efektów uczenia się przez przedstawiciela podmiotu, w którym realizowano praktykę oraz jego opinią o studencie. Ważnym elementem procesu zaliczenia są wnioski studenta z realizacji praktyki, samoocena działań, realizacji zadań i postawy społecznej w trakcie praktyki oraz uwagi dotyczące realizacji praktyki zawodowej.

W przypadku wniosku o zaliczenie praktyki poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego, wpływ na decyzję ma przedstawiona dokumentacja potwierdzająca uzyskanie efektów uczenia się w pracy zawodowej, argumentacja studenta zawarta we wniosku oraz opinia przełożonego.

Praktyka zawodowa IV

Nazwa zajęć po angielsku: Professional practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie bez oceny

Punkty ECTS: 6

Materiały: brak

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Dodatkowe zajęcia wynikające z organizacji studiów

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Praktyka.....160 / 160 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Praktyka zawodowa na wyższych semestrach służy potwierdzeniu i pogłębieniu wiedzy oraz kształtowaniu umiejętności i kompetencji społecznych w rzeczywistym środowisku pracy informatyka. Student realizuje zadania powierzone przez podmiot przyjmujący na praktykę (zgodnie z katalogiem rekomendowanych obszarów), ale – o ile to możliwe – powiązane z tematyką przygotowywanego projektu inżynierskiego. Praktyka umożliwia bezpośredni kontakt z procesami IT (wytworzenie, utrzymanie, sieci, bezpieczeństwo, bazy danych, UX/UI, DevOps), uczestnictwo w pracy zespołowej, poznanie standardów branżowych, narzędzi oraz stosowanie zasad etyki i odpowiedzialności zawodowej. Celem nadrzędnym jest zapewnienie transferu i utrwalenie efektów kierunkowych w warunkach pracy, z priorytetem na działanie praktyczne, feedback i refleksję.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
rozpoznaje praktyczne zastosowania wiedzy kierunkowej w zadaniach realizowanych podczas praktyki	K_W03
wyjaśnia podstawowe procesy działania urządzeń, systemów oraz sieci w kontekście realizowanych czynności	K_W04
zna wybrane metody, technologie i narzędzia stosowane w praktyce zawodowej informatyków oraz rozumie ich praktyczne zastosowanie	K_W05
analizuje wpływ standardów i norm na jakość i bezpieczeństwo wykonania zadania; uzasadnia wybory	K_W06
zna, rozumie i analizuje dylematy społeczeństwa informacyjnego związane z wykonywaną praktyką	K_W07
Umiejętności (student potrafi)	
samodzielnie uczy się i rozwija swoje kompetencje zawodowe, korzystając z doświadczenia praktycznego oraz wskazówek opiekuna praktyk (przełożonego), w tym pozyskuje wiedzę i zastosowuje ją do rozwiązania problemu praktycznego w ramach projektu inżynierskiego	K_U01
identyfikuje i rozwiązuje praktyczne problemy techniczne i organizacyjne występujące w środowisku pracy informatyka, wykorzystując zdobytą wiedzę i doświadczenie m.in. w realizowanym projekcie inżynierskim	K_U02
stosuje odpowiednie metody, technologie, narzędzia i standardy branżowe do rozwiązywania i realizacji zadań zleconych przez opiekuna praktyk (przełożonego) w miejscu pracy	K_U12
wykorzystuje doświadczenie zdobyte w środowisku zawodowym do diagnozowania i rozwiązywania problemów związanych z eksploatacją i utrzymaniem systemów	K_U13

informatycznych	
skutecznie komunikuje się w środowisku zawodowym z wykorzystaniem specjalistycznej terminologii i odpowiednich narzędzi komunikacyjnych	K_U14
planuje własną pracę informatyka w kontekście zdobywania doświadczenia praktycznego do wykorzystania w przygotowaniu projektu inżynierskiego	K_U17
Kompetencje społeczne (student jest gotów do)	
jest gotów do krytycznej oceny własnych działań w środowisku pracy oraz przyjmowania konstruktywnej krytyki od współpracowników i przełożonych	K_K01
jest gotów do rozwiązywania problemów zawodowych w sposób odpowiedzialny i samodzielny, a w przypadku trudności potrafi skutecznie współpracować z zespołem lub zasięgać opinii ekspertów	K_K02

Metody i formy kształcenia:

- Pogadanka
- Metoda sytuacyjna
- Studium przypadku
- Metoda pokazu
- Metoda ćwiczeń

Metody oceniania:

- Rozliczenie praktyki

Praktyka (Pr)

1. Zapoznanie z przepisami i zasadami BHP, ochrony danych osobowych (RODO), tajemnicy służbowej, cyberbezpieczeństwa, archiwizowania danych firmowych i innymi regulacjami określającymi funkcjonowanie podmiotu.
2. Poznanie struktury organizacyjnej podmiotu, zakresu działalności, sposobu zarządzania i środowiska zawodowego. Zapoznanie się z przepływem informacji oraz rodzajami i obiegiem dokumentów w instytucji.
3. Obserwacja pracy podmiotu / działu IT. Poznanie procesów i wykorzystywanych metodologii pracy, zapoznanie z używanymi w podmiocie standardami, technologiami i narzędziami.
4. Realizacja, pod nadzorem opiekuna praktyki, zadań praktycznych wybranych z katalogu rekomendowanych zagadnień.
5. Konsultacja tematyki projektu inżynierskiego z praktykami.

Literatura podstawowa

1. Zarządzenie nr 11/2023 Rektora SWPW z dnia 5 lipca 2023 r. w sprawie Uczelnianego Regulaminu Studenckich Praktyk Zawodowych obowiązującego studentów Szkoły Wyższej im. Pawła Włodkowica w Płocku
2. Uchwała 15/2019 Senatu SWPW w Płocku z dnia 30 września 2019 r. w sprawie ustalenia Zasad, warunków i trybu zaliczania w SWPW praktyki zawodowej poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego
3. Katalog rekomendowanych zagadnień do realizacji na praktyce zawodowej na kierunku Informatyka

Literatura uzupełniająca

1. Zarządzenie nr 06/2023 Rektora SWPW w Płocku z dnia 13 kwietnia 2023 r. w sprawie realizacji zadań Uczelnianego Pełnomocnika ds. praktyk w Szkole Wyższej im. Pawła Włodkowica w Płocku
2. Zarządzenie nr 13/2024 Rektora SWPW z dnia 22 maja 2024 r. w sprawie wprowadzenia "Zasad oceny realizacji praktyk przez studentów SWPW w Płocku"

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....160 godz.....160 godz.
- Suma obciążeń:160 godz.....160 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia praktyki jest właściwe zrealizowanie zadań praktycznych w wybranym podmiocie oraz uzyskanie efektów uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych. Potwierdzeniem tego jest wypełniony dziennik praktyk z oceną uzyskania efektów uczenia się przez przedstawiciela podmiotu, w którym realizowano praktykę oraz jego opinią o studencie. Ważnym elementem procesu zaliczenia są wnioski studenta z realizacji praktyki, samoocena działań, realizacji zadań i postawy społecznej w trakcie praktyki oraz uwagi dotyczące realizacji praktyki zawodowej.

W przypadku wniosku o zaliczenie praktyki poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego, wpływ na decyzję ma przedstawiona dokumentacja potwierdzająca uzyskanie efektów uczenia się w pracy zawodowej, argumentacja studenta zawarta we wniosku oraz opinia przełożonego.

Praktyka zawodowa V

Nazwa zajęć po angielsku: Professional practice

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VII

Forma zaliczenia: Zaliczenie bez oceny

Punkty ECTS: 12

Materiały: brak

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Dodatkowe zajęcia wynikające z organizacji studiów

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Praktyka.....320 / 320 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Praktyka zawodowa na wyższych semestrach służy potwierdzeniu i pogłębieniu wiedzy oraz kształtowaniu umiejętności i kompetencji społecznych w rzeczywistym środowisku pracy informatyka. Student realizuje zadania powierzone przez podmiot przyjmujący na praktykę (zgodnie z katalogiem rekomendowanych obszarów), ale – o ile to możliwe – powiązane z tematyką przygotowywanego projektu inżynierskiego. Praktyka umożliwia bezpośredni kontakt z procesami IT (wytworzenie, utrzymanie, sieci, bezpieczeństwo, bazy danych, UX/UI, DevOps), uczestnictwo w pracy zespołowej, poznanie standardów branżowych, narzędzi oraz stosowanie zasad etyki i odpowiedzialności zawodowej. Celem nadrzędnym jest zapewnienie transferu i utrwalenie efektów kierunkowych w warunkach pracy, z priorytetem na działanie praktyczne, feedback i refleksję.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
rozpoznaje praktyczne zastosowania wiedzy kierunkowej w zadaniach realizowanych podczas praktyki	K_W03
wyjaśnia podstawowe procesy działania urządzeń, systemów oraz sieci w kontekście realizowanych czynności	K_W04
zna wybrane metody, technologie i narzędzia stosowane w praktyce zawodowej informatyków oraz rozumie ich praktyczne zastosowanie	K_W05
analizuje wpływ standardów i norm na jakość i bezpieczeństwo wykonania zadania; uzasadnia wybory	K_W06
zna, rozumie i analizuje dylematy społeczeństwa informacyjnego związane z wykonywaną praktyką	K_W07
Umiejętności (student potrafi)	
potrafi samodzielnie uczyć się i rozwijać swoje kompetencje zawodowe, korzystając z doświadczenia praktycznego oraz wskazówek opiekuna praktyk (przełożonego), w tym pozyskuje wiedzę i zastosowuje ją do rozwiązywania problemu praktycznego w ramach projektu inżynierskiego	K_U01
potrafi identyfikować i rozwiązywać praktyczne problemy techniczne i organizacyjne występujące w środowisku pracy informatyka, wykorzystując zdobytą wiedzę i doświadczenie m.in. w realizowanym projekcie inżynierskim	K_U02
potrafi stosować odpowiednie metody, technologie, narzędzia i standardy branżowe do rozwiązywania i realizacji zadań zleconych przez opiekuna praktyk (przełożonego) w miejscu pracy	K_U12

wykorzystuje doświadczenie zdobyte w środowisku zawodowym do diagnozowania i rozwiązywania problemów związanych z eksploatacją i utrzymaniem systemów informatycznych	K_U13
potrafi skutecznie komunikować się w środowisku zawodowym z wykorzystaniem specjalistycznej terminologii i odpowiednich narzędzi komunikacyjnych	K_U14
planuje własną pracę informatyka w kontekście zdobywania doświadczenia praktycznego do wykorzystania w przygotowaniu projektu inżynierskiego	K_U17
Kompetencje społeczne (student jest gotów do)	
jest gotów do krytycznej oceny własnych działań w środowisku pracy oraz przyjmowania konstruktywnej krytyki od współpracowników i przełożonych	K_K01
jest gotów do rozwiązywania problemów zawodowych w sposób odpowiedzialny i samodzielny, a w przypadku trudności potrafi skutecznie współpracować z zespołem lub zasięgać opinii ekspertów	K_K02

Metody i formy kształcenia:

- Pogadanka
- Metoda sytuacyjna
- Studium przypadku
- Metoda pokazu
- Metoda ćwiczeń

Metody oceniania:

- Rozliczenie praktyki

Praktyka (Pr)

1. Zapoznanie z przepisami i zasadami BHP, ochrony danych osobowych (RODO), tajemnicy służbowej, cyberbezpieczeństwa, archiwizowania danych firmowych i innymi regulacjami określającymi funkcjonowanie podmiotu.
2. Poznanie struktury organizacyjnej podmiotu, zakresu działalności, sposobu zarządzania i środowiska zawodowego. Zapoznanie się z przepływem informacji oraz rodzajami i obiegiem dokumentów w instytucji.
3. Obserwacja pracy podmiotu / działu IT. Poznanie procesów i wykorzystywanych metodologii pracy, zapoznanie z używanymi w podmiocie standardami, technologiami i narzędziami.
4. Realizacja, pod nadzorem opiekuna praktyki, zadań praktycznych wybranych z katalogu rekomendowanych zagadnień.
5. Konsultacja tematyki projektu inżynierskiego z praktykami.

Literatura podstawowa

1. Zarządzenie nr 11/2023 Rektora SWPW z dnia 5 lipca 2023 r. w sprawie Uczelnianego Regulaminu Studenckich Praktyk Zawodowych obowiązującego studentów Szkoły Wyższej im. Pawła Włodkowica w Płocku
2. Uchwała 15/2019 Senatu SWPW w Płocku z dnia 30 września 2019 r. w sprawie ustalenia Zasad, warunków i trybu zaliczania w SWPW praktyki zawodowej poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego
3. Katalog rekomendowanych zagadnień do realizacji na praktyce zawodowej na kierunku Informatyka

Literatura uzupełniająca

1. Zarządzenie nr 06/2023 Rektora SWPW w Płocku z dnia 13 kwietnia 2023 r. w sprawie realizacji zadań Uczelnianego Pełnomocnika ds. praktyk w Szkole Wyższej im. Pawła Włodkowica w Płocku
2. Zarządzenie nr 13/2024 Rektora SWPW z dnia 22 maja 2024 r. w sprawie wprowadzenia "Zasad oceny realizacji praktyk przez studentów SWPW w Płocku"

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....160 godz.....160 godz.
- Suma obciążeń:160 godz.....160 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia praktyki jest właściwe zrealizowanie zadań praktycznych w wybranym podmiocie oraz uzyskanie efektów uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych. Potwierdzeniem tego jest wypełniony dziennik praktyk z oceną uzyskania efektów uczenia się przez przedstawiciela podmiotu, w którym realizowano praktykę oraz jego opinią o studencie. Ważnym elementem procesu zaliczenia są wnioski studenta z

realizacji praktyki, samoocena działań, realizacji zadań i postawy społecznej w trakcie praktyki oraz uwagi dotyczące realizacji praktyki zawodowej.

W przypadku wniosku o zaliczenie praktyki poprzez uznanie efektów uczenia się na podstawie doświadczenia zawodowego, wpływ na decyzję ma przedstawiona dokumentacja potwierdzająca uzyskanie efektów uczenia się w pracy zawodowej, argumentacja studenta zawarta we wniosku oraz opinia przełożonego.

Praca w zespole

Nazwa zajęć po angielsku: Teamwork

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: IV

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 3

Materiały: kurs e-learningowy + materiały

Język zajęć: polski

Zajęcia do wyboru: Nie

Grupa zajęć: Dodatkowe zajęcia wynikające z organizacji studiów

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny 15 / 8 godz 15 / 8 godz.
- Ćwiczenia 30 / 16 godz 0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Umiejętność pracy zespołowej to jedna z najbardziej poszukiwanych kompetencji absolwentów szkół wyższych, szczególnie kierunków technicznych. Przedmiot przeznaczony jest dla studentów kierunku Informatyka, którzy chcą zdobyć podstawową wiedzę niezbędną do sprawnego i efektywnego funkcjonowania w zespole oraz zdobyć umiejętności wspólnej pracy z wykorzystaniem narzędzi informatycznych. Celem kształcenia jest prezentacja wiedzy oraz zdobycie umiejętności, które pozwolą studentowi rozwinąć kompetencji interpersonalne, niezbędne do efektywnej pracy w zespole. Zaprezentowane zostaną narzędzia i techniki pomocne w usprawnianiu współpracy, a także przyczynią się do większej integracji i spójności zespołu. Wiadomości teoretyczne poparte są testem Belbina diagnozującym role zespołowe poszczególnych członków grup oraz samoocena własnych zdolności. Ponadto w trakcie zajęć ćwiczeniowych studenci pracują w grupach w internetowym systemie kontroli wersji.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia procesy zachodzące w systemach kontroli wersji podczas zespołowej realizacji projektu informatycznego	K_W04
identyfikuje narzędzia zarządzania pracą zespołu informatycznego, w tym GitHub, Issues, Projects/Kanban, README, wiki, pull request i code review stosowane przy rozwiązywaniu praktycznych zadań inżynierskich z zakresu informatyki	K_W05
charakteryzuje znaczenie pracy zespołowej, komunikacji, odpowiedzialności i dzielenia się wiedzą w społeczeństwie informacyjnym oraz w projektach informatycznych	K_W07
charakteryzuje zespół, zasady efektywnej współpracy oraz czynniki wpływające na integrację i spójność zespołu	U_W11
Umiejętności (student potrafi)	
dobiera metody i narzędzia pracy zespołowej do realizacji zadania informatycznego, uwzględniając repozytorium, tablicę zadań i sposób komunikacji	K_U04
organizuje i realizuje zespołowe zadanie informatyczne z wykorzystaniem Git, GitHub, tablicy zadań, dokumentacji i uzgodnionego podziału odpowiedzialności	K_U05
uwzględnia społeczne, komunikacyjne i etyczne aspekty pracy informatyka w zespole, w tym feedback, konflikty, odpowiedzialność za kod i kulturę współpracy	K_U08
stosuje technologie i dobre praktyki zespołowego wytwarzania oprogramowania, w tym Kanban, Git, pull request, code review, README i dokumentowanie decyzji	K_U12
stosuje podstawowe techniki komunikacji, konstruktywnego rozwiązywania konfliktów,	K_U14

udzielania informacji zwrotnej i wspierania członków zespołu w sytuacji zmiany	
przedstawia swoje stanowisko w zespole współpracujących ze sobą ludzi oraz podejmować wspólne decyzje	K_U15
planuje i organizuje pracę własną oraz zespołu, określając zadania, priorytety, role, terminy, odpowiedzialności i sposób raportowania postępu	K_U17
współdziała z innymi osobami w zespole projektowym, wykonując zadania, zgłaszając zmiany, uczestnicząc w code review i rozwiązując konflikty współpracy	K_U18
Kompetencje społeczne (student jest gotów do)	
jest gotów do krytycznej oceny własnych umiejętności pracy zespołowej, komunikacji, jakości wkładu i sposobu reagowania na feedback	K_K01
jest gotów do zasięgania opinii ekspertów, prowadzącego, recenzentów kodu i członków zespołu w przypadku trudności projektowych lub konfliktu decyzyjnego	K_K02
jest gotów do przestrzegania zasad etyki zawodowej w pracy zespołowej, w tym uczciwego przypisywania autorstwa, odpowiedzialności za kod i szacunku w komunikacji	K_K03

Metody i formy kształcenia:

- Praca ze źródłem elektronicznym
- Metoda sytuacyjna
- Metoda ćwiczeń
- Studium przypadku

Metody oceniania:

- Zaliczenie testowe
- Zaliczenie praktyczne
- Ocena aktywności w kursie

Wykład konwersatoryjny (W)

1. Praca zespołowa w IT i jej uwarunkowania - role, odpowiedzialności, komunikacja, zaufanie i jakość współpracy.
2. Modele zespołu: Belbin jako narzędzie refleksji, ograniczenia typologii ról, samoocena i feedback
3. Proces budowania zespołu.
4. Komunikacja w zespole.
5. Podejmowanie decyzji zespołowych.
6. Pobudzanie kreatywności zespołu.
7. Konflikty i ich rozwiązywanie w zespole.
8. Zespół w sytuacji wdrażania zmiany.

Ćwiczenia (Ć)

1. Test M. Belbina – identyfikacja własnych predyspozycji i ról zespołowych. Ustalenie mocnych stron, ryzyk, preferencji współpracy
2. Budowa zespołu - dobór członków na podstawie uzdolnień i ról w zespole.
3. Wprowadzenie do systemu Git – podstawowe polecenia, zasady pracy, struktura projektu.
4. Praca w serwisie GitHub - licencja, zespół.
5. Organizacja pracy zespołowej – projekty, tablica zadań (Kanban), harmonogramy, komunikacja, priorytety, kryteria akceptacji.
6. Praca z gałęziami: branch, commit, push, merge, rozwiązywanie konfliktów.
7. Zgłoszenia akceptacji zmian (pull request) - opis zmiany, powiązanie z issue, review, poprawki, merge.
8. Przegląd kodu (code review) - komentarze, checklisty, prośba o zmiany, akceptacja.
9. Dokumentacja pracy zespołu – README, Issues, wiki.
10. Zadanie zespołowe - mały projekt grupowy z repozytorium, pull requestami, code review i prezentacją wkładu.
11. Podsumowanie pracy zespołowej – prezentacja, refleksja i wnioski końcowe.

Literatura podstawowa

1. kurs e-learningowy i dodatkowe materiały na platformie WLODEK

Literatura uzupełniająca

1. A. Hardigan, Praca w zespole. Petit 1999

2. R. M. Belbin, Twoja rola w zespole. GWP 2008
3. M. Tsitoara, Git i GitHub. Kontrola wersji, zarządzanie projektami i zasady pracy zespołowej

Źródła dodatkowe

1. <https://git-scm.com/book/en/v2>
2. <https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-project>
3. <https://docs.github.com/en/organizations/collaborating-with-groups-in-organizations/best-practices-for-organizations>
4. <https://github.blog/enterprise-software/devops/best-practices-for-organizations-and-teams-using-github-enterprise-cloud/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	30 godz.....	16 godz.
• Realizacja kursu e-learningowego.....	25 godz.....	30 godz.
• Zapoznanie się z literaturą przedmiotu.....	5 godz.....	10 godz.
• Przygotowanie się do zaliczenia końcowego.....	15 godz.....	19 godz.
Suma obciążeń:.....	75 godz.....	75 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest, co najmniej w stopniu podstawowym, przyswojenie wiadomości, zdobycie umiejętności oraz kompetencji społecznych umożliwiających efektywną pracę w zespole z wykorzystaniem standardów i narzędzi informatycznych. Ocena końcowa zależy w 20% od realizacji kursu e-learningowego (liczba punktów zdobytych z odpowiedzi na pytania kontrolne zadawane w trakcie realizacji lekcji), w 30% od zaliczenia wykładu (liczba punktów zdobytych podczas rozwiązywania testu końcowego) oraz w 50% od wyniku uzyskanego w trakcie realizacji praktycznego zadania zespołowego kończącego zajęcia ćwiczeniowe. Zaliczenie otrzymują osoby, które uzyskają ponad 50% punktów, a ocena zależy od ich ilości (co 10% o pół oceny wyżej).

Projekt inżynierski I

Nazwa zajęć po angielsku: Engineering Project I

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie bez oceny

Punkty ECTS: 2

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Dodatkowe zajęcia wynikające z organizacji studiów

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot „Projekt inżynierski” jest trzysemestralnym procesem dydaktyczno-projektowym, którego celem jest przygotowanie studenta do samodzielnego zaplanowania, wykonania, udokumentowania i zaprezentowania praktycznego rozwiązania inżynierskiego z zakresu informatyki. W ramach zajęć student rozwija projekt odpowiadający rzeczywistym potrzebom użytkowników, organizacji lub środowiska zawodowego. Projekt może mieć charakter aplikacyjny, systemowy, infrastrukturalny, sieciowy albo cyberbezpieczeństwa. Istotą przedmiotu jest połączenie wiedzy technicznej z praktyką inżynierską, odpowiedzialnością za jakość rozwiązania, dokumentowaniem podjętych decyzji oraz umiejętnością uzasadnienia przyjętych założeń projektowych. Efektem końcowym przedmiotu jest wykonana samodzielnie lub zespołowo praktyczna praca inżynierska, rozumiana jako zrealizowane i udokumentowane rozwiązanie informatyczne. Praca praktyczna powinna być uzupełniona dokumentacją obejmującą analizę problemu, opis wymagań, uzasadnienie wyboru technologii, opis wykonania, testy, analizę wyników, wnioski oraz wskazanie możliwości dalszego rozwoju rozwiązania.

Semestr 5 służy przygotowaniu koncepcji projektu inżynierskiego. Student wybiera temat, określa problem inżynierski, opisuje genezę, cel, wizję systemu, miary sukcesu, grupę docelową, role użytkowników, wymagania funkcjonalne i нефункционалне oraz analizuje istniejące rozwiązania. Dla projektów programistycznych powstaje wstępna struktura aplikacji, model danych, założenia UI i plan repozytorium. Dla projektów infrastrukturalnych i cyberbezpieczeństwa powstaje wstępny projekt sieci/systemu, założenia bezpieczeństwa, normy, wymagania prawne i plan symulacji.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
charakteryzuje praktyczny kontekst problemu inżynierskiego oraz zastosowanie planowanego rozwiązania	K_W03
identyfikuje metody, techniki i narzędzia właściwe dla planowanego projektu informatycznego	K_W05
Umiejętności (student potrafi)	
analizuje problem projektowy oraz formułuje założenia rozwiązania	K_U01
dobiera źródła, selekcjonuje informacje i porównuje rozwiązania konkurencyjne lub alternatywne	K_U03
dokonyuje wstępnej analizy ekonomicznej projektowanego rozwiązania informatycznego, uwzględniając koszty jego wykonania, wdrożenia i eksploatacji	K_U09
planuje i organizuje prace projektowe indywidualnie lub w zespole	K_U17
Kompetencje społeczne (student jest gotów do)	
ocenia zakres własnych kompetencji i identyfikuje potrzebę konsultacji eksperckich	K_K02

Metody i formy kształcenia:

- Metoda projektu
- Dyskusja

Metody oceniania:

- Rozliczenie projektu

- Metoda symulacyjna

Projekt (P)

1. Rola projektu inżynierskiego w procesie dyplomowania.
2. Wybór tematu i określenie problemu inżynierskiego.
3. Geneza pracy, cel pracy, wizja systemu, określenie miar sukcesu.
4. Grupa docelowa, interesariusze, role użytkowników.
5. Wymagania funkcjonalne i нефункционалне.
6. Analiza rozwiązań konkurencyjnych lub alternatywnych.
7. Wymagania prawne, etyczne, techniczne i bezpieczeństwa.
8. Harmonogram, ryzyka, organizacja pracy i plan dokumentacji.

Literatura podstawowa

1. L. Bass, R. Kazman, P. Clements: Architektura oprogramowania w praktyce. Wydanie IV, Helion 2022
2. R. White, E. Banks: Sieci komputerowe. Najczęstsze problemy i ich rozwiązania, Helion 2019
3. W. Nowak (i inni): Cyberbezpieczeństwo. Zarys wykładu, Wydawnictwo: Wolters Kluwer Polska SA 2023

Literatura uzupełniająca

1. M. Keeling: Jak zostać architektem oprogramowania, WN PWN 2021
2. M. Rogalski: Bezpieczeństwo sieci i usług łączności elektronicznej w prawie Unii Europejskiej oraz w prawie polskim, Wydawnictwo: Uczelnia Łazarskiego, 2024
3. R. Anderson: Inżynieria zabezpieczeń Tom I, WN PWN 2024

Źródła dodatkowe

1. arc42 Template Overview - <https://arc42.org/overview>
2. S. Brown, The C4 Model, O'Reilly - <https://c4model.com>
3. IEEE 830 Template. Appendix C - <https://press.rebus.community/requirementsengineering/back-matter/appendix-c-ieee-830-template>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- | | | |
|---|--------------|----------|
| • Zajęcia z udziałem nauczyciela akademickiego..... | 15 godz..... | 8 godz. |
| • Zapoznanie się z literaturą przedmiotu..... | 8 godz..... | 10 godz. |
| • Przygotowanie projektu..... | 16 godz..... | 19 godz. |
| • Wypełnienie dokumentacji zajęć..... | 11 godz..... | 13 godz. |
| Suma obciążeń:..... | 50 godz..... | 50 godz. |

Zaliczenie zajęć:

Warunkiem zaliczenia jest złożenie częściowej dokumentacji obejmującej: analizę wymagań, analizę rozwiązań, harmonogram i wstępną dokumentację zgodną z wybraną ścieżką.

Projekt inżynierski II

Nazwa zajęć po angielsku: Engineering Project II

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie bez oceny

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Dodatkowe zajęcia wynikające z organizacji studiów

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Projekt..... 30 / 16 godz..... 0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

Opis zajęć:

Przedmiot „Projekt inżynierski” jest trzysemestralnym procesem dydaktyczno-projektowym, którego celem jest przygotowanie studenta do samodzielnego zaplanowania, wykonania, udokumentowania i zaprezentowania praktycznego rozwiązania inżynierskiego z zakresu informatyki. W ramach zajęć student rozwija projekt odpowiadający rzeczywistym potrzebom użytkowników, organizacji lub środowiska zawodowego. Projekt może mieć charakter aplikacyjny, systemowy, infrastrukturalny, sieciowy albo cyberbezpieczeństwa. Istotą przedmiotu jest połączenie wiedzy technicznej z praktyką inżynierską, odpowiedzialnością za jakość rozwiązania, dokumentowaniem podjętych decyzji oraz umiejętnością uzasadnienia przyjętych założeń projektowych. Efektem końcowym przedmiotu jest wykonana samodzielnie lub zespołowo praktyczna praca inżynierska, rozumiana jako zrealizowane i udokumentowane rozwiązanie informatyczne. Praca praktyczna powinna być uzupełniona dokumentacją obejmującą analizę problemu, opis wymagań, uzasadnienie wyboru technologii, opis wykonania, testy, analizę wyników, wnioski oraz wskazanie możliwości dalszego rozwoju rozwiązania.

Semestr 6 służy realizacji rozwiązania zgodnie z założeniami przyjętymi w semestrze 5. W ścieżce programistycznej student tworzy repozytorium, strukturę projektu, implementuje moduły, przygotowuje UI, stosuje wersjonowanie, opisuje technologie, zabezpieczenia, walidację danych i diagnostykę. W ścieżce infrastrukturalnej/cyberbezpieczeństwa student opracowuje schematy, dobiera technologie i urządzenia, przygotowuje konfigurację, symulację działania sieci lub mechanizmów bezpieczeństwa oraz dokumentuje wykonane prace.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia procesy zachodzące w projektowanym systemie informatycznym, sieciowym lub teleinformatycznym	K_W04
identyfikuje standardy i normy techniczne właściwe dla realizowanego projektu	K_W06
Umiejętności (student potrafi)	
przeprowadza eksperymenty, testy cząstkowe lub symulacje oraz interpretuje uzyskane wyniki	K_U06
projektuje oraz realizuje aplikację, system bezpieczeństwa lub strukturę sieci teleinformatycznej	K_U11
Kompetencje społeczne (student jest gotów do)	
ocenia decyzje wykonawcze z uwzględnieniem zasad etyki zawodowej informatyka	K_K05

Metody i formy kształcenia:

- Metoda projektu
- Dyskusja
- Metoda sytuacyjna
- Studium przypadku

Metody oceniania:

- Rozliczenie projektu

Projekt (P)

1. Przegląd założeń z semestru 5 i ewentualna korekta planu.
2. Organizacja środowiska pracy: repozytorium, struktura plików, dokumentacja.
3. Implementacja modułów aplikacji lub konfiguracja infrastruktury.
4. Kontrola wersji, gałęzie, commity, pull requesty, CI.
5. Walidacja danych, autoryzacja, logowanie, diagnostyka i bezpieczeństwo aplikacji.
6. Schematy infrastruktury, konfiguracja urządzeń, adresacja, polityki bezpieczeństwa.
7. Symulacje działania sieci lub bezpieczeństwa.
8. Konsultacje postępów i rozwiązywanie problemów projektowych.
9. Przygotowanie dokumentacji wykonawczej.

Literatura podstawowa

1. W zależności od wybranej tematyki i zakresu projektu inżynierskiego.

Literatura uzupełniająca

1. R. Anderson: Inżynieria zabezpieczeń Tom I, WN PWN 2024
2. R. Anderson: Inżynieria zabezpieczeń Tom II, WN PWN 2024

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	30 godz.....	16 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	12 godz.
• Przygotowanie projektu.....	44 godz.....	52 godz.
• Wypełnienie dokumentacji zajęć.....	16 godz.....	20 godz.
Suma obciążeń:.....	100 godz.....	100 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest przedstawienie działającego prototypu / aplikacji / konfiguracji / symulacji, repozytorium lub plików konfiguracyjnych, raportu postępu i częściowej dokumentacji technicznej.

Projekt inżynierski III

Nazwa zajęć po angielsku: Engineering Project III

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VII

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 5

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Dodatkowe zajęcia wynikające z organizacji studiów

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Projekt..... 45 / 24 godz..... 0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot „Projekt inżynierski” jest trysemestralnym procesem dydaktyczno-projektowym, którego celem jest przygotowanie studenta do samodzielnego zaplanowania, wykonania, udokumentowania i zaprezentowania praktycznego rozwiązania inżynierskiego z zakresu informatyki. W ramach zajęć student rozwija projekt odpowiadający rzeczywistym potrzebom użytkowników, organizacji lub środowiska zawodowego. Projekt może mieć charakter aplikacyjny, systemowy, infrastrukturalny, sieciowy albo cyberbezpieczeństwa. Istotą przedmiotu jest połączenie wiedzy technicznej z praktyką inżynierską, odpowiedzialnością za jakość rozwiązania, dokumentowaniem podjętych decyzji oraz umiejętnością uzasadnienia przyjętych założeń projektowych. Efektem końcowym przedmiotu jest wykonana samodzielnie lub zespołowo praktyczna praca inżynierska, rozumiana jako zrealizowane i udokumentowane rozwiązanie informatyczne. Praca praktyczna powinna być uzupełniona dokumentacją obejmującą analizę problemu, opis wymagań, uzasadnienie wyboru technologii, opis wykonania, testy, analizę wyników, wnioski oraz wskazanie możliwości dalszego rozwoju rozwiązania.

Semestr 7 służy domknięciu projektu inżynierskiego: wykonaniu testów, analizie rezultatów, przygotowaniu kompletnej dokumentacji, sformułowaniu wniosków i ewentualne przygotowaniu prezentacji do egzaminu końcowego. Student porządkuje dokumentację techniczną, podpisuje i numeruje rysunki, tabele, kody lub pliki konfiguracyjne, opracowuje bibliografię, netografię, spisy oraz załączniki.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
wyjaśnia ekonomiczne, prawne i etyczne uwarunkowania opisywanego rozwiązania	K_W08
identyfikuje zasady prawa autorskiego i ochrony własności przemysłowej stosowane w pracy inżynierskiej	K_W09
Umiejętności (student potrafi)	
formułuje i uzasadnia wnioski wynikające z testów, symulacji lub analizy działania rozwiązania	K_U02
analizuje funkcjonowanie wykonanego systemu, sieci, usługi lub procesu i ocenia istniejące rozwiązania techniczne	K_U10
przygotowuje dokumentację i porozumiewa się z użyciem specjalistycznej terminologii informatycznej	K_U14
przedstawia i ocenia opinie dotyczące własnego rozwiązania oraz dyskutuje o nich z użyciem terminologii specjalistycznej	K_U15
Kompetencje społeczne (student jest gotów do)	
jest gotowy do oceny własnej wiedzy, ograniczenia projektu i jakości wykorzystanych źródeł	K_K01
jest gotów odpowiedzialnie korzystać z dorobku i dobrych praktyk zawodu informatyka oraz dbać o ich zachowanie poprzez rzetelne dokumentowanie własnego rozwiązania i wiedzy	K_K06

Metody i formy kształcenia:

- Metoda projektu
- Dyskusja
- Metoda sytuacyjna
- Metoda symulacyjna
- Metoda pokazu

Metody oceniania:

- Rozliczenie projektu

Projekt (P)

1. Audyt stanu projektu po semestrze 6.
2. Plan testów końcowych i kryteria akceptacji.
3. Testy funkcjonalne, niefunkcjonalne, bezpieczeństwa lub symulacje.
4. Analiza wyników testów i korekta rozwiązania.
5. Redakcja rozdziałów pracy: geneza, cel, analiza, rozwiązanie, testy.
6. Bibliografia, netografia, przypisy, spisy i załączniki.
7. Podsumowanie z wnioskami, ograniczenia, kierunki rozwoju.

Literatura podstawowa

1. arc42 Template Overview - <https://arc42.org/overview>

Literatura uzupełniająca

1. P. Kutnyj: Sztuka autoprezentacji i wystąpień publicznych, WN PWN 2020
2. S. Brown, The C4 Model, O'Reilly - <https://c4model.com>

Źródła dodatkowe

1. IEEE 830 Template. Appendix C - <https://press.rebus.community/requirementsengineering/back-matter/appendix-c-ieee-830-template>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	45 godz.....	24 godz.
• Zapoznanie się z literaturą przedmiotu.....	12 godz.....	16 godz.
• Przygotowanie projektu.....	42 godz.....	58 godz.
• Wypełnienie dokumentacji zajęć.....	26 godz.....	27 godz.
Suma obciążeń:.....	125 godz.....	125 godz.

Zaliczenie zajęć:

Zaliczenie polega na przedstawieniu projektu inżynierskiego oraz złożeniu dokumentacji pisemnej. Na ocenę końcową wpływ ma:

- * kompletność pracy pisemnej – 40%
- * jakość wykonanego rozwiązania / prototypu / konfiguracji / symulacji – 30%
- * testy, analiza wyników i wnioski – 10%
- * odpowiedzi na pytania opiekuna – 20%

Zarządzanie bezpieczeństwem informacji

Nazwa zajęć po angielsku: Security Information Management

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 6

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Specjalność: Sieci komputerowe i cyberbezpieczeństwo

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Bogusław Kowalski

Opis zajęć:

Celem kształcenia na kierunku Informatyka – Zarządzanie Bezpieczeństwem Informacji jest dostarczenie studentom kompetencji wymaganych przez sektor ochrony systemów informatycznych i cyberprzestrzeni. Praca w tym obszarze wymaga połączenia wysokich umiejętności inżynierskich (programowanie, analiza systemów komputerowych, zarządzanie infrastrukturą) z szerokimi umiejętnościami społecznymi (komunikatywność, umiejętność działania w zespole, zdolność do działania w warunkach stresu, umiejętności społeczne). Absolwenci uzyskają specjalistyczną wiedzę w wielu obszarach mających znaczenie dla bezpieczeństwa. Do obszarów tych należą programowanie i budowa oprogramowania, konfigurowanie i zarządzanie infrastrukturą informatyczną, zbieranie i analizowanie informacji, testy penetracyjne i audyt, sztuczna inteligencja i jej zastosowania do detekcji zagrożeń, aspekty prawne bezpieczeństwa w cyberprzestrzeni, socjotechnika, psychologia. Dzięki efektom kształcenia zaplanowanym w ramach kierunku absolwent będzie w stanie stawić czoło wieloaspektowym i w ostatnim czasie znacząco rosnącym wyzwaniom w obszarze ochrony systemów informatycznych i cyberprzestrzeni.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Student wyjaśnia zastosowanie zarządzania bezpieczeństwem informacji w organizacjach, systemach informatycznych i usługach cyfrowych.	K_W03
Student analizuje przebieg procesów technicznych i organizacyjnych w systemach bezpieczeństwa informacyjnego	K_W04
Student przytacza standardy bezpieczeństwa informacji (np. ISO/IEC 27001, RODO) i normy techniczne związane z ochroną informacji i wyjaśnia ich znaczenie w systemach zarządzania bezpieczeństwem	K_W06
Student rozpoznaje problemy społeczne związane z nadzorem cyfrowym, monitoringiem, profilowaniem, prywatnością i zaufaniem do usług cyfrowych.	K_W07
Student wyjaśnia prawne, ekonomiczne i etyczne uwarunkowania ochrony danych, cyberbezpieczeństwa, nadzoru, prywatności i reagowania na incydenty	K_W08
Student rozróżnia prawa i obowiązki wynikające z ochrony własności intelektualnej i omawia sposoby zapobiegania naruszeniom praw autorskich w sieci	K_W09
Umiejętności (student potrafi)	
Student formułuje problem bezpieczeństwa informacji oraz proponuje środki organizacyjne i techniczne adekwatne do ryzyka	K_U01
Student integruje dane z audytu, logów, skanów podatności i dokumentacji organizacyjnej	K_U02

oraz uzasadnia wnioski z analizy ryzyka	
Student dobiera źródła, regulacje, dokumentację techniczną i raporty o zagrożeniach do przygotowania polityki bezpieczeństwa lub procedury incydentowej	K_U03
Student dobiera narzędzia informatyczne wspomagające zarządzanie bezpieczeństwem, analizę logów, wykrywanie podatności i obsługę incydentów	K_U04
Student opracowuje procedurę odpowiedzi na incydent oraz dokumentuje działania dla scenariuszy takich jak phishing, ransomware, wyciek danych lub DDoS	K_U05
Student planuje i przeprowadza kontrolowany eksperyment bezpieczeństwa, np. analizę logów, skan podatności lub test konfiguracji, oraz interpretuje wyniki	K_U06
Student uwzględnia aspekty systemowe, organizacyjne, prawne i etyczne podczas projektowania zabezpieczeń i procedur bezpieczeństwa	K_U08
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny aktualności własnej wiedzy, przyjętych zabezpieczeń, polityk i wyników audytu	K_K01
Student jest gotów do zasięgania opinii ekspertów w przypadku złożonych problemów prawnych, organizacyjnych lub technicznych	K_K02
Student jest gotów do inicjowania działań edukacyjnych i prewencyjnych na rzecz ochrony prywatności i bezpieczeństwa informacji	K_K03
Student jest gotów do działania przedsiębiorczego przy planowaniu usług i rozwiązań bezpieczeństwa informacji	K_K04
Student jest gotów do przestrzegania zasad etyki zawodowej, ochrony danych, poufności informacji i odpowiedzialnego używania narzędzi bezpieczeństwa	K_K05
Student jest gotów do samorozwoju w zakresie zmieniających się zagrożeń, standardów, przepisów, technologii bezpieczeństwa informacji oraz wykorzystania AI	U_K07

Metody i formy kształcenia:

- Wykład problemowy
- Dyskusja
- Studium przypadku
- Burza mózgów
- Metoda ćwiczeń
- Metoda projektu

Metody oceniania:

- Zaliczenie opisowe
- Rozliczenie projektu
- Przygotowanie do zajęć - wejściówki
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

1. Wprowadzenie do bezpieczeństwa informacji Pojęcie informacji jako zasobu Triada CIA: poufność, integralność, dostępność Prawne i etyczne podstawy ochrony informacji
2. Standardy i normy w zakresie bezpieczeństwa informacji ISO/IEC 27001, ISO/IEC 27002 – struktura, wymagania, wdrażanie RODO – ochrona danych osobowych w kontekście IT Krajowe ramy polityki bezpieczeństwa (np. ustawa o KSC)
3. Identyfikacja zagrożeń i analiza ryzyka Typowe zagrożenia dla informacji (malware, phishing, ransomware, insider threat) Metody oceny ryzyka (np. analiza jakościowa, STRIDE, DREAD) Klasyfikacja informacji i modelowanie zagrożeń
4. Zarządzanie incydentami bezpieczeństwa Definicja i typologia incydentów Cykl życia zarządzania incydentem Dokumentowanie i raportowanie incydentów
5. Polityki bezpieczeństwa informacji Struktura i funkcje polityki bezpieczeństwa Przykłady polityk: haseł, dostępu, szyfrowania, backupu Tworzenie polityki bezpieczeństwa w organizacji
6. Zarządzanie dostępem i autoryzacją Modele kontroli dostępu: DAC, MAC, RBAC, ABAC Autoryzacja i uwierzytelnianie użytkowników Systemy kontroli dostępu fizycznego i logicznego
7. Narzędzia i techniki zabezpieczania informacji Firewalle, IDS/IPS, VPN, szyfrowanie Oprogramowanie antywirusowe i antymalware Bezpieczna konfiguracja systemów operacyjnych i sieć
8. Audyt i monitoring bezpieczeństwa Cele i zakres audytów Rejestracja i analiza logów Monitoring w czasie

rzeczywistym – SIEM

9. Aspekty społeczne, etyczne i ekonomiczne bezpieczeństwa Etyka zawodowa i odpowiedzialność informatyka
Cyberbezpieczeństwo a zaufanie społeczne Koszty wdrażania systemów bezpieczeństwa vs ryzyko incydentu

Ćwiczenia (Ć)

1. Wprowadzenie do środowiska bezpieczeństwa informacyjnego Instalacja narzędzi testowych i diagnostycznych (np. Wireshark, Nmap, Metasploit)
2. Analiza zagrożeń i podatności systemów Skanowanie portów i usług sieciowych (Nmap, Zenmap)
3. Monitorowanie ruchu sieciowego i analiza pakietów Przechwytywanie pakietów z użyciem Wireshark
4. Bezpieczeństwo haseł i autoryzacji użytkowników Analiza logów logowania i prób włamań
5. Audyt bezpieczeństwa – studium przypadku Analiza przykładowego przypadku naruszenia bezpieczeństwa

Projekt (P)

Literatura podstawowa

1. Kowalewski J., Kowalewski M. - Zarządzanie ryzykiem w bezpieczeństwie informacji organizacji, Oficyna Wydawnicza Politechniki Warszawskiej, 2021.
2. Księżopolski B. – Bezpieczeństwo systemów komputerowych. Teoria i praktyka, PWN, Warszawa.
3. Siczek K. – Ochrona informacji w systemach i sieciach komputerowych, Helion, Gliwice.
4. Polska Norma PN-EN ISO/IEC 27001 Bezpieczeństwo informacji, cyberbezpieczeństwo i ochrona prywatności. Systemy zarządzania bezpieczeństwem informacji. Wymagania

Literatura uzupełniająca

1. Andress J. - Podstawy bezpieczeństwa informacji : praktyczne wprowadzenie, Helion
2. Pfleeger C., Pfleeger S. – Bezpieczeństwo w systemach komputerowych, Helion.
3. Grzelak A. – Audyt bezpieczeństwa systemów informatycznych, PWN.

Źródła dodatkowe

1. Serwis CERT Polska – raporty i zalecenia dotyczące cyberbezpieczeństwa (<https://cert.pl>).
2. ENISA – European Union Agency for Cybersecurity – raporty o stanie bezpieczeństwa informacji (<https://www.enisa.europa.eu>).
3. Portale branżowe: Zaufana Trzecia Strona, Niebezpiecznik.pl, Sekurak.pl.

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Przygotowanie projektu.....	25 godz.....	35 godz.
• Przygotowanie się do zajęć.....	20 godz.....	25 godz.
• Zapoznanie się z literaturą przedmiotu.....	20 godz.....	25 godz.
• Przygotowanie się do zaliczenia końcowego.....	10 godz.....	25 godz.
Suma obciążeń:.....	150 godz.....	150 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie zaliczenia z wszystkich trzech form prowadzenia zajęć: co najmniej 50% punktów z testu zaliczeniowego wykład, co najmniej 50% punktów ze sprawozdań z realizowanych ćwiczeń i co najmniej 50% punktów z zespołowego projektu ocenianego na podstawie złożonej i zaprezentowanej dokumentacji. Ocena końcowa wyliczana jest na podstawie znormalizowanych liczb punktów z poszczególnych form zajęć z odpowiadającymi im wagami: 40% wykład, 30% ćwiczenia, 30% projekt.

Monitoring zagrożeń cyberbezpieczeństwa

Nazwa zajęć po angielsku: Cybersecurity Threat Monitoring

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 6

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Specjalność: Sieci komputerowe i cyberbezpieczeństwo

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Andrzej Ulicki

Opis zajęć:

Przedmiot „Monitoring zagrożeń cyberbezpieczeństwa” przygotowuje studentów do praktycznego wykrywania, analizowania i kwalifikowania zdarzeń mogących wskazywać na naruszenie bezpieczeństwa systemów informatycznych, sieci, usług i tożsamości użytkowników. Stanowi rozwinięcie wiedzy zdobytej na wcześniejszych zajęciach z technologii sieciowych, administracji, bezpieczeństwa systemów informatycznych i kryptografii, koncentrując się na operacyjnym wykorzystaniu telemetrii bezpieczeństwa.

Studenci poznają źródła danych wykorzystywane w monitoringu bezpieczeństwa, zasady centralizacji i korelacji logów, architekturę systemów SIEM, IDS/IPS, NDR i EDR/XDR, metody tworzenia reguł detekcji oraz proces kwalifikowania alertów w środowisku Security Operations Center (SOC). Wprowadzane są techniki mapowania obserwowanych zdarzeń do modelu zachowań przeciwnika, wykorzystania informacji o zagrożeniach oraz prowadzenia threat huntingu.

Część praktyczna obejmuje zbieranie i analizę logów systemowych i sieciowych, analizę alertów, tworzenie i strojenie reguł detekcji, korelację zdarzeń, ocenę false positive/false negative oraz dokumentowanie wyników analizy. Studenci realizują projekt zespołowy polegający na zaprojektowaniu i uruchomieniu środowiska monitorowania bezpieczeństwa dla przykładowej infrastruktury oraz przygotowaniu scenariuszy detekcji wybranych zagrożeń.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
student wyjaśnia rolę monitoringu bezpieczeństwa, Security Operations Center oraz procesów detekcji i kwalifikowania zagrożeń w ochronie infrastruktury informatycznej	K_W03
student charakteryzuje źródła telemetrii bezpieczeństwa oraz przepływ informacji pomiędzy systemami końcowymi, urządzeniami sieciowymi, systemami IDS/IPS, EDR/NDR i platformą SIEM	K_W04
student porównuje metody, techniki i narzędzia wykorzystywane do gromadzenia, normalizacji, korelacji, wyszukiwania i analizy zdarzeń związanych z cyberbezpieczeństwem	K_W05
student wyjaśnia zastosowanie standardów, modeli i formatów wykorzystywanych w monitorowaniu oraz detekcji zagrożeń, w tym taksonomii technik przeciwnika i reguł detekcyjnych	K_W06
Umiejętności (student potrafi)	
student integruje, interpretuje i koreluje informacje pochodzące z logów systemowych, sieciowych, aplikacyjnych oraz systemów bezpieczeństwa w celu odtworzenia przebiegu podejrzanego zdarzenia	K_U02
student dobiera źródła telemetrii, reguły oraz narzędzia monitorowania odpowiednio do	K_U04

analizowanego scenariusza zagrożenia	
student identyfikuje, analizuje i kwalifikuje anomalie oraz alerty bezpieczeństwa, odróżniając zdarzenia wymagające reakcji od zdarzeń fałszywie dodatnich	K_U05
student projektuje, przeprowadza i interpretuje testy skuteczności reguł oraz mechanizmów wykrywania zagrożeń	K_U06
student ocenia skuteczność konfiguracji systemu monitorowania, zakres widoczności infrastruktury oraz jakość reguł detekcji, wskazując luki i proponując działania korygujące	K_U10
student projektuje i realizuje środowisko monitorowania bezpieczeństwa integrujące wybrane źródła telemetrii, reguły detekcji, korelację zdarzeń i sposób prezentacji alertów	K_U11
student opracowuje raport z analizy zdarzenia bezpieczeństwa i uzasadnia wnioski z wykorzystaniem specjalistycznej terminologii	K_U14
Kompetencje społeczne (student jest gotów do)	
student jest gotów do krytycznej oceny otrzymywanych alertów, informacji o zagrożeniach i wyników działania automatycznych systemów detekcji	K_K01
student jest gotów do zasięgania opinii ekspertów oraz eskalowania zdarzeń w sytuacji, w której zakres lub skutki incydentu przekraczają jego kompetencje	K_K02
student jest gotów do przestrzegania zasad etyki zawodowej podczas monitorowania aktywności użytkowników, analizowania danych telemetrycznych oraz przetwarzania informacji związanych z incydentami	K_K05

Metody i formy kształcenia:

- Dyskusja
- Praca ze źródłem elektronicznym
- Metoda symulacyjna
- Metoda ćwiczeń
- Metoda projektu
- Wykład problemowy

Metody oceniania:

- Zaliczenie testowe
- Zaliczenie praktyczne
- Rozliczenie projektu

Wykład konwersatoryjny (W)

1. Monitoring zagrożeń cyberbezpieczeństwa i organizacja procesu detekcji
2. Systemy i technologie monitorowania bezpieczeństwa
3. Wykrywanie i analiza zagrożeń
4. Analiza alertów, korelacja zdarzeń i threat hunting
5. Współczesne kierunki monitorowania cyberzagrożeń

Ćwiczenia (Ć)

1. Pozyskiwanie i przygotowanie danych do monitorowania bezpieczeństwa
2. Monitorowanie aktywności systemów, sieci i użytkowników
3. Wykrywanie zagrożeń i tworzenie reguł detekcji
4. Analiza i korelacja alertów bezpieczeństwa
5. Doskonalenie skuteczności monitorowania i detekcji
6. Analiza zagrożeń i threat hunting
7. Analiza złożonego scenariusza cyberzagrożenia

Projekt (P)

1. Analiza problemu i określenie założeń projektu systemu monitorowania cyberzagrożeń
2. Opracowanie koncepcji systemu monitorowania i detekcji zagrożeń
3. Realizacja i konfiguracja rozwiązania projektowego
4. Opracowanie i przeprowadzenie scenariuszy testowych
5. Ocena i doskonalenie opracowanego rozwiązania

6. Opracowanie dokumentacji i prezentacja projektu

Literatura podstawowa

1. praca zbiorowa - Cyberbezpieczeństwo. Zarys wykładu. Wolters Kluwer Polska SA 2023
2. Andress J. - Podstawy bezpieczeństwa informacji : praktyczne wprowadzenie, Helion 2022
3. Kowalewski J., Kowalewski M. - Zarządzanie ryzykiem w bezpieczeństwie informacji organizacji, Oficyna Wydawnicza Politechniki Warszawskiej, 2021.

Literatura uzupełniająca

1. V. Costa-Gazcon, Aktywne wykrywanie zagrożeń w systemach IT w praktyce. Wykorzystywanie analizy danych, frameworku ATT&CK oraz narzędzi open source. Helion 2022
2. Molendowska M., Miernik R. (red.) - Bezpieczeństwo w cyberprzestrzeni: wybrane zagadnienia. Wydawnictwo Adam Marszałek, 2020.

Źródła dodatkowe

1. MITRE | ATT&CK - <https://attack.mitre.org/>
2. wazuh - <https://documentation.wazuh.com/current/> (<https://documentation.wazuh.com/current/getting-started/use-cases/threat-hunting.html>)
3. zeek - <https://docs.zeek.org/en/current/>
4. Suricata User Guide - <https://docs.suricata.io/en/>
5. Sigma Detection Format - <https://sigmahq.io/>
6. Wyszukiwanie zagrożeń w usłudze Microsoft Sentinel - <https://learn.microsoft.com/pl-pl/training/paths/sc-200-perform-threat-hunting-azure-sentinel/>
7. NIST - <https://csrc.nist.gov/Projects/incident-response/publications>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Zapoznanie się z literaturą przedmiotu.....	15 godz.....	20 godz.
• Przygotowanie się do zajęć.....	15 godz.....	25 godz.
• Wypełnienie dokumentacji zajęć.....	15 godz.....	20 godz.
• Przygotowanie projektu.....	20 godz.....	35 godz.
• Przygotowanie się do zaliczenia końcowego.....	10 godz.....	10 godz.
Suma obciążeń:.....	150 godz.....	150 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie zaliczenia z wszystkich trzech form prowadzenia zajęć: co najmniej 50% punktów z testu zaliczeniowego wykład, co najmniej 50% punktów ze sprawozdań z realizowanych ćwiczeń i co najmniej 50% punktów z zespołowego projektu ocenianego na podstawie złożonej i zaprezentowanej dokumentacji. Ocena końcowa wyliczana jest na podstawie znormalizowanych liczb punktów z poszczególnych form zajęć z odpowiadającymi im wagami.

Informatyka śledcza

Nazwa zajęć po angielsku: Computer forensics

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 6

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Specjalność: Sieci komputerowe i cyberbezpieczeństwo

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Andrzej Ulicki

Opis zajęć:

Przedmiot "Informatyka śledcza" przeznaczony jest dla studentów kierunku Informatyka, którzy zainteresowani są obsługą incydentów bezpieczeństwa. Studenci poznają metody zabezpieczania, zbierania, weryfikacji, identyfikacji, analizy, interpretacji, dokumentowania oraz prezentowania cyfrowego materiału dowodowego pozyskanego z różnorodnych cyfrowych źródeł danych w celu ujawnienia i rekonstrukcji zdarzeń polegających na wykonaniu nieautoryzowanych czynności lub mających charakter kryminalny. W trakcie zajęć studenci nauczą się lokalizować i interpretować dowody elektroniczne znajdujące się w systemach komputerowych, m.in. na dyskach, w systemach plików, w pamięci RAM, w logach systemowych. Poznają też i wykorzystują praktycznie techniki i narzędzia stosowane do skutecznego wykrywania artefaktów, które jako istotne dla prowadzonego dochodzenia stają się dowodem cyfrowym.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
student wyjaśnia praktyczne zastosowania informatyki śledczej w obsłudze incydentów, rekonstrukcji zdarzeń, analizie powłamaniowej i dokumentowaniu dowodów cyfrowych	K_W03
student charakteryzuje procesy zachodzące w systemach operacyjnych, systemach plików, pamięci, logach, aplikacjach i sieciach, które generują artefakty śledcze	K_W04
student identyfikuje metody, techniki i narzędzia stosowane w identyfikacji, zabezpieczaniu, pozyskiwaniu, analizie i dokumentowaniu dowodów cyfrowych	K_W05
rozpoznaje standardy i dobre praktyki informatyki śledczej, w tym zasady integralności dowodu, chain of custody, ISO/IEC 27037, ISO/IEC 27042, NIST i SWGDE	K_W06
student wyjaśnia prawne, ekonomiczne i etyczne uwarunkowania pozyskiwania, przetwarzania, przechowywania i prezentowania cyfrowego materiału dowodowego	K_W08
Umiejętności (student potrafi)	
student integruje dane z obrazu nośnika, pamięci RAM, logów, artefaktów internetowych i źródeł zewnętrznych oraz uzasadnia wnioski śledcze w warunkach niepełnej informacji	K_U02
student dobiera źródła, dokumentację, bazy wiedzy, standardy i narzędzia do przeprowadzenia analizy śledczej określonego incydentu	K_U03
student dobiera metody, techniki i narzędzia do wyszukiwania artefaktów, pozyskiwania obrazu, analizy pamięci, logów i śladów aktywności użytkownika	K_U04
student rozwiązuje praktyczny problem śledczy przez zabezpieczenie materiału, analizę	K_U05

artefaktów, odtworzenie przebiegu zdarzeń i sformułowanie wniosków	
student planuje i przeprowadza badanie śledcze w środowisku laboratoryjnym, interpretuje wyniki i przygotowuje dokumentację zgodną z zasadami pracy z dowodem	K_U06
student wykorzystuje metody analityczne, symulacyjne i eksperymentalne do rekonstrukcji incydentu oraz weryfikacji hipotez śledczych	K_U07
student uwzględnia aspekty prawne, organizacyjne i etyczne podczas pozyskiwania, analizy i prezentowania materiału dowodowego	K_U08
student ocenia funkcjonowanie systemu informatycznego pod kątem śladów incydentu, luk bezpieczeństwa i wiarygodności pozyskanych artefaktów	K_U10
student współdziała w zespole śledczym, dokumentując własny wkład, rolę, decyzje i wyniki analizy	K_U18
Kompetencje społeczne (student jest gotów do)	
student jest gotów do zasięgania opinii ekspertów i korzystania z wiedzy doświadczonych praktyków przy złożonych problemach analizy śledczej	K_K02
student jest gotów do przestrzegania zasad etyki zawodowej, poufności, legalności działań i integralności materiału dowodowego	K_K05
student jest gotów do samorozwoju w zakresie nowych technik ataku, artefaktów systemowych, narzędzi forensic i procedur DFIR	U_K07

Metody i formy kształcenia:

- Wykład problemowy
- Dyskusja
- Studium przypadku
- Metoda ćwiczeń
- Metoda projektu
- Metoda laboratoryjna

Metody oceniania:

- Zaliczenie testowe
- Sprawozdanie z realizacji zadania
- Rozliczenie projektu

Wykład konwersatoryjny (W)

1. Proces dochodzeniowy i zasady pracy z dowodami.
2. Pozyskiwanie dowodów elektronicznych - metody, techniki i narzędzia kryminalistyczne.
3. Analiza artefaktów w systemach operacyjnych.
4. Analiza i pozyskiwanie dowodów z nośników pamięci.
5. Analiza systemów plików i logów systemowych.
6. Artefakty internetowe.
7. Skuteczne robienie notatek i opracowanie raportów.

Ćwiczenia (Ć)

1. Budowa stanowiska badawczego - dobór sprzętu i oprogramowania.
2. Mapowanie i emulowanie przeciwnika.
3. Cechy wspólne i różnice w badaniu różnych systemów operacyjnych.
4. Badanie sekwencji rozruchowej systemu operacyjnego.
5. Analiza i pozyskiwanie danych z systemu plików.
6. Analiza pamięci RAM.
7. Analiza zawartości nośników pamięci oraz systemu plików.
8. Analiza logów systemowych.
9. Wyszukiwanie i analiza artefaktów internetowych.
10. Automatyzacja procesów analizy śledczej.

Projekt (P)

1. Podział na zespoły oraz przydzielenie ról w zespołach.

2. Opracowanie scenariuszy incydentów bezpieczeństwa.
3. Przygotowanie procesu dochodzeniowego.
4. Pozyskiwanie dowodów z różnych źródeł.
5. Przygotowanie raportu końcowego.

Literatura podstawowa

1. W. Oettinger, Informatyka śledcza. Helion 2022
2. G. Johansen, Informatyka śledcza. Narzędzia i techniki skutecznego reagowania na incydenty bezpieczeństwa. Wydanie III, Helion 2024
3. S. V. N. Parasram, Informatyka śledcza i Kali Linux. Przeprowadź analizy nośników pamięci, ruchu sieciowego i zawartości RAM-u za pomocą narzędzi systemu Kali Linux 2022.x. Wydanie III, Helion 2024

Literatura uzupełniająca

1. P. Olber, Informatyka śledcza i cyberprzestępczość. Wybrane zagadnienia w ujęciu policyjnym. Akademia Policji w Szczytnie 2022
2. B. Nikkel, Systemy Linux w kryminalistyce. Praktyczny przewodnik dla analityków śledczych. Helion 2022
3. J. Luttgens, M. Pepe, K. Mandia, Incydenty bezpieczeństwa. Metody reagowania w informatyce śledczej. Helion 2016
4. C. Altheide, H. Carvey, Informatyka śledcza. Przewodnik po narzędziach open source, Helion 2014

Źródła dodatkowe

1. <https://resources.infosecinstitute.com/>
2. <https://www.esecurityplanet.com/>
3. <https://www.kali.org/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	20 godz.
• Przygotowanie się do zajęć.....	20 godz.....	25 godz.
• Wypełnienie dokumentacji zajęć.....	15 godz.....	25 godz.
• Przygotowanie projektu.....	20 godz.....	25 godz.
• Przygotowanie się do zaliczenia końcowego.....	10 godz.....	15 godz.
Suma obciążeń:.....	150 godz.....	150 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskanie zaliczenia z wszystkich trzech form prowadzenia zajęć: co najmniej 50% punktów z testu zaliczeniowego wykład, co najmniej 50% punktów ze sprawozdań z realizowanych ćwiczeń i co najmniej 50% punktów z zespołowego projektu ocenianego na podstawie złożonej i zaprezentowanej dokumentacji. Ocena końcowa wyliczana jest na podstawie znormalizowanych liczb punktów z poszczególnych form zajęć z odpowiadającymi im wagami.

Systemy zarządzania bezpieczeństwem sieci komputerowych

Nazwa zajęć po angielsku: Computer Network Management Systems

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VII

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Specjalność: Sieci komputerowe i cyberbezpieczeństwo

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....45 / 24 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Bogusław Kowalski

Opis zajęć:

Przedmiot „Systemy zarządzania bezpieczeństwem sieci komputerowych” integruje wiedzę i umiejętności zdobyte podczas wcześniejszych zajęć z zakresu technologii sieciowych, projektowania i administracji sieciami oraz cyberbezpieczeństwa. Koncentruje się na systemowym podejściu do zapewniania bezpieczeństwa infrastruktury sieciowej organizacji poprzez monitorowanie jej stanu, gromadzenie i korelację informacji o zdarzeniach, identyfikację zagrożeń, ocenę ryzyka oraz planowanie i podejmowanie działań ograniczających skutki incydentów. Studenci poznają architekturę i zasady działania systemów wspierających zarządzanie bezpieczeństwem sieci, wykorzystujących m.in. SNMP, Syslog, NetFlow/IPFIX, systemy NMS, IDS/IPS oraz SIEM. Analizują zależności pomiędzy zarządzaniem infrastrukturą, monitorowaniem zdarzeń, zarządzaniem podatnościami, kontrolą dostępu, segmentacją sieci, reagowaniem na incydenty oraz zapewnianiem ciągłości działania usług sieciowych. W części praktycznej studenci analizują dane pochodzące z różnych źródeł infrastruktury sieciowej, identyfikują symptomy awarii i incydentów bezpieczeństwa, oceniają konfigurację zabezpieczeń oraz opracowują i weryfikują rozwiązania służące monitorowaniu i zarządzaniu bezpieczeństwem sieci. Przedmiot uwzględnia również automatyzację zarządzania, wykorzystanie współczesnych systemów wspomaganie analizy zagrożeń oraz aspekty prawne, etyczne i organizacyjne związane z monitorowaniem infrastruktury i aktywności użytkowników.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Student porównuje metody, techniki i narzędzia służące do gromadzenia, korelacji i analizy informacji o stanie infrastruktury, zdarzeniach, podatnościach i zagrożeniach sieciowych.	K_W03
Student charakteryzuje procesy monitorowania, zabezpieczania, utrzymania i odtwarzania infrastruktury sieciowej w cyklu życia systemów informatycznych i teleinformatycznych.	K_W04
Student identyfikuje metody, techniki i narzędzia stosowane w zarządzaniu bezpieczeństwem sieci, w szczególności SNMPv3, Syslog, NetFlow, IDS/IPS, SIEM, NMS/NSM oraz narzędzia analizy ruchu.	K_W05
Student rozpoznaje standardy, normy i dobre praktyki wykorzystywane w zarządzaniu bezpieczeństwem sieci, w szczególności ISO/IEC 27001, ISO/IEC 27002, ISO/IEC 27005, NIST CSF oraz zalecenia dotyczące obsługi incydentów.	K_W06
Student wyjaśnia prawne, organizacyjne i etyczne uwarunkowania monitorowania sieci, rejestrowania logów, ochrony danych, reagowania na incydenty i egzekwowania polityk bezpieczeństwa.	K_W08
Umiejętności (student potrafi)	
Student analizuje logi, statystyki ruchu, alerty bezpieczeństwa i informacje o konfiguracji sieci oraz formułuje wnioski dotyczące stabilności, ryzyka i bezpieczeństwa infrastruktury	K_U02

Student dobiera źródła informacji technicznej, dokumentację narzędzi i standardów bezpieczeństwa sieci oraz ocenia ich przydatność do rozwiązania problemu administracyjnego lub incydentu.	K_U03
Student dobiera i stosuje narzędzia monitorowania, analizy ruchu, centralizacji logów i wykrywania incydentów do typowych zadań zarządzania bezpieczeństwem sieci.	K_U04
Student rozwiązuje praktyczne problemy związane z bezpieczeństwem sieci, w tym identyfikuje zagrożenia, proponuje środki zaradcze i uzasadnia dobór zabezpieczeń.	K_U05
Student planuje i przeprowadza test monitorowania lub analizy bezpieczeństwa sieci, interpretuje wyniki oraz formułuje wnioski dotyczące skuteczności zastosowanych rozwiązań.	K_U06
Student uwzględnia aspekty techniczne, organizacyjne, prawne i etyczne podczas projektowania monitorowania sieci, rejestrowania aktywności użytkowników i reagowania na incydenty.	K_U08
Student ocenia istniejące rozwiązania bezpieczeństwa sieciowego pod kątem podatności, skuteczności monitorowania, kompletności logowania, zgodności z polityką bezpieczeństwa i odporności na incydenty.	K_U10
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny odbieranych informacji o zagrożeniach, podatnościach, alertach i konfiguracjach bezpieczeństwa sieci.	K_K01
Student jest gotów do rozwiązywania problemów bezpieczeństwa sieci oraz do zasięgania opinii ekspertów, dokumentacji producenta lub zespołu w przypadku złożonych incydentów i decyzji konfiguracyjnych.	K_K02
Student jest gotów do przestrzegania zasad etyki zawodowej administratora bezpieczeństwa sieci, w szczególności w zakresie monitorowania użytkowników, ochrony danych, poufności logów i odpowiedzialnego reagowania na incydenty.	K_K05
Student jest gotów do rozwijania wiedzy z zakresu nowych zagrożeń, narzędzi monitorowania, systemów SIEM/IDS/IPS oraz standardów bezpieczeństwa sieci, z poszanowaniem praw własności intelektualnej.	U_K07

Metody i formy kształcenia:

- Wykład problemowy
- Dyskusja
- Metoda ćwiczeń
- Studium przypadku

Metody oceniania:

- Zaliczenie opisowe
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

1. Systemowe zarządzanie bezpieczeństwem sieci – bezpieczeństwo jako proces obejmujący infrastrukturę, usługi, użytkowników, dane i procedury.
2. Architektura systemu zarządzania bezpieczeństwem sieci – źródła danych, zbieranie, agregacja, korelacja, analiza, alarmowanie, reakcja i raportowanie
3. Widoczność infrastruktury jako podstawa bezpieczeństwa – inwentaryzacja zasobów, topologia, stan urządzeń i usług.
4. Źródła informacji o stanie sieci – SNMP/SNMPv3, Syslog, NetFlow/IPFIX, logi urządzeń i usług, telemetria.
5. Monitorowanie dostępności, wydajności i bezpieczeństwa – zależności pomiędzy awarią, przeciążeniem, błędną konfiguracją i incydemtem.
6. NMS, IDS/IPS, SIEM i inne klasy systemów – zakresy odpowiedzialności, możliwości i integracja.
7. Identyfikacja anomalii i zdarzeń bezpieczeństwa – korelacja danych z wielu źródeł i kontekst zdarzenia.
8. Zarządzanie podatnościami i konfiguracją infrastruktury sieciowej – wykrywanie, ocena, priorytetyzacja i działania korygujące.
9. Segmentacja, kontrola dostępu i ograniczanie skutków incydentów – podejście defense in depth i Zero Trust.
10. Zarządzanie incydemtem sieciowym – wykrycie, analiza, ograniczenie skutków, usunięcie przyczyn, odtworzenie

i wnioski.

11. Odporność i ciągłość działania infrastruktury sieciowej – redundancja, kopie konfiguracji, odtwarzanie usług i Disaster Recovery.
12. Automatyzacja zarządzania bezpieczeństwem – automatyczne zbieranie danych, reakcje, orkiestracja i SOAR.
13. Bezpieczeństwo środowisk chmurowych, hybrydowych, IoT i pracy zdalnej – rozszerzanie granic zarządzanej sieci.
14. AI i analiza behawioralna w zarządzaniu bezpieczeństwem sieci – możliwości, ograniczenia i ryzyka automatyzacji decyzji.
15. Aspekty prawne, etyczne i organizacyjne monitorowania sieci – prywatność, zakres monitoringu, odpowiedzialność administratora i dokumentowanie działań.

Ćwiczenia (Ć)

1. Analiza architektury przykładowej infrastruktury sieciowej i identyfikacja zasobów wymagających monitorowania.
2. Identyfikacja źródeł informacji o stanie i bezpieczeństwie infrastruktury
3. Konfiguracja gromadzenia informacji z wykorzystaniem SNMP/SNMPv3.
4. Centralizacja i analiza komunikatów Syslog.
5. Definiowanie zdarzeń, progów, alarmów i reguł korelacji.
6. Analiza scenariusza awarii infrastruktury i odróżnienie awarii od incydentu bezpieczeństwa.
7. Analiza scenariusza ataku na infrastrukturę sieciową.
8. Identyfikacja i priorytetyzacja podatności urządzeń i usług sieciowych.
9. Analiza zdarzeń z wykorzystaniem systemu klasy SIEM.
10. Opracowanie scenariusza reakcji na wybrany incydent sieciowy.
11. Projekt integracji systemów monitorowania i bezpieczeństwa dla przykładowej organizacji.
12. Testowanie, ocena skuteczności i dokumentowanie zaprojektowanego rozwiązania.

Literatura podstawowa

1. Stallings W. – Kryptografia i bezpieczeństwo sieci komputerowych. Zasady i praktyka, Helion
2. Grzelak A. – Audyt bezpieczeństwa systemów informatycznych, PWN
3. <https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf>

Literatura uzupełniająca

1. Nowicki A., Świątek J. – Inżynieria bezpieczeństwa systemów informatycznych, Wydawnictwo PJWSTK
2. Skorupski J. (red.) – Bezpieczeństwo informacyjne i ochrona danych osobowych w administracji publicznej, Difin.

Źródła dodatkowe

1. <https://csrc.nist.gov/library/alt-SP800-61rev2.pdf>
2. <https://www.iso.org/pl/aktualnosci-2022/iso-27001-2022/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	60 godz.....	32 godz.
• Przygotowanie się do zaliczenia końcowego.....	20 godz.....	28 godz.
• Przygotowanie się do zajęć.....	10 godz.....	20 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	20 godz.
Suma obciążeń:.....	100 godz.....	100 godz.

Zaliczenie zajęć:

Wykład:

Zaliczenie w formie pisemnej obejmującego pytania testowe i problemowe.

Warunkiem uzyskania zaliczenia jest zdobycie minimum 50% punktów .

Ćwiczenia:

Bieżąca ocena aktywności i zaangażowania podczas zajęć (udział w dyskusjach, realizacja krótkich zadań analitycznych).

Student przygotowuje opis infrastruktury organizacji oraz zestaw danych pochodzących z systemów i urządzeń sieciowych. Analizuje stan infrastruktury, identyfikuje zagrożenia i nieprawidłowości, dobiera mechanizmy monitorowania i zabezpieczenia, projektuje sposób integracji systemów zarządzania bezpieczeństwem oraz

przygotowuje rekomendacje dotyczące reagowania na wykryte zdarzenia.
Oddanie zadania w terminie oraz uzyskanie oceny co najmniej dostatecznej.

Bezpieczeństwo oprogramowania i testy penetracyjne

Nazwa zajęć po angielsku: Software security and penetration testing

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VII

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Grupa zajęć: Zajęcia międzyspecjalnościowe do wyboru

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....45 / 24 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr inż. Andrzej Ulicki

Opis zajęć:

Przedmiot "Bezpieczeństwo oprogramowania i testy penetracyjne" przeznaczony jest dla studentów kierunku Informatyka, którzy zainteresowani są zapewnieniem bezpieczeństwa informatycznego. Tematyka przedmiotu jest szeroka. Obejmuje zagadnienia projektowania i wytwarzania bezpiecznego oprogramowania, identyfikacji ważnych zasobów, wyszukiwania podatności oprogramowania na ataki, wykrywania luk w zabezpieczeniach sieciowych oraz przeprowadzania testów penetracyjnych. Dzięki poznanym metodologiom, narzędziom oraz technikom testów penetracyjnych, na zajęciach praktycznych studenci zbudują nowoczesne środowisko testowe umożliwiające realizację kontrolowanych prób przełamania zabezpieczeń. Nie zabrakło tematyki dotyczącej przygotowania raportów z badań, obejmujących nie tylko informację o wykrytych błędach, ale dających jak najwięcej użytecznych informacji odbiorcy.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
zna podstawowe zagadnienia związane z bezpieczeństwem aplikacji internetowych i rozumie procesy bezpiecznego projektowania aplikacji z uwzględnieniem aspektów bezpieczeństwa	K_W04
zna podstawowe metody hakerów oraz zna techniki i narzędzia wykorzystywane podczas realizacji testów penetracyjnych	K_W05
zna metodologię realizacji testów penetracyjnych	K_W06
zna i rozumie problemy związane z zapewnieniem bezpieczeństwa informacji we współczesnym świecie	K_W07
posiada wiedzę z zakresu socjotechniki oraz wie jakie metody inżynierii społecznej wykorzystują hakerzy	U_W11
Umiejętności (student potrafi)	
potrafi wykorzystać dotychczasową wiedzę na temat projektowania aplikacji internetowych i wyciągać wnioski dotyczące bezpieczeństwa danych w sytuacji stosowania coraz to wymyślniejszych metod ich pozyskiwania przez osoby niepowołane	K_U02
potrafi dobrać właściwe metody, techniki oraz narzędzia w celu przeprowadzenia testów bezpieczeństwa w zależności od metod ataku	K_U04
potrafi wykorzystać posiadaną wiedzę z zakresu projektowania aplikacji internetowych a oraz rozwiązywać praktyczne problemy z ich bezpieczeństwem w trakcie eksploatacji	K_U05
potrafi przygotować środowisko testowe oraz zaplanować, wykonać testy penetracyjne, zaprezentować uzyskane dane, wyciągnąć wnioski i przygotować raport przydatny dla klienta	K_U06

dostrzega etyczne problemy wynikające z prób hakowania systemów i aplikacji	K_U08
potrafi dokonać krytycznej analizy funkcjonowania aplikacji internetowych oraz ocenić faktyczne zagrożenia	K_U10
potrafi przygotować raport z badań i zaprezentować wyniki w sposób przystępny dla odbiorcy	K_U14
Kompetencje społeczne (student jest gotów do)	
gotów jest do zasięgania opinii ekspertów od zabezpieczeń sieciowych, będąc świadomym zmieniających się zagrożeń oraz szybkości ich rozpowszechniania się we współczesnym świecie	K_K02
jest gotów do przestrzegania zasad etyki w trakcie realizowania testów bezpieczeństwa i uzyskując dostęp do danych klienta	K_K05

Metody i formy kształcenia:

- Wykład informacyjny
- Dyskusja
- Studium przypadku
- Metoda ćwiczeń
- Metoda laboratoryjna
- Metoda symulacyjna

Metody oceniania:

- Zaliczenie testowe
- Sprawozdanie z realizacji zadania

Wykład konwersatoryjny (W)

1. Bezpieczeństwo aplikacji - identyfikacja zagrożeń oraz obszarów ataku.
2. Projektowanie i programowanie z uwzględnieniem aspektów bezpieczeństwa.
3. Typowe mechanizmy wykorzystywane w atakach i obrona przed nimi.
4. Etyka i hacking.
5. Metodologia testów penetracyjnych, wykorzystywane techniki i narzędzia.
6. Wykorzystanie inżynierii społecznej (socjotechnika).

Ćwiczenia (Ć)

1. Przygotowanie środowiska testowego.
2. Analiza oprogramowania oraz istniejących zabezpieczeń systemu.
3. Skanowanie sieci.
4. Ataki w systemie lokalnym.
5. Ataki w sieci - wykorzystywanie usług sieciowych.
6. Podatności i luki w zabezpieczeniach aplikacji internetowych.
7. Kryptografia i łamanie skrótów haseł.
8. Raportowanie wyników badań i prezentacja danych.

Literatura podstawowa

1. A. Hoffman, Bezpieczeństwo nowoczesnych aplikacji internetowych. Przewodnik po zabezpieczeniach. Helion 2021

Literatura uzupełniająca

1. P. Kim, Podręcznik pentestera. Bezpieczeństwo systemów informatycznych. Helion 2015
2. V. Li, Bug Bounty Bootcamp. Przewodnik po tropieniu i zgłaszaniu luk w zabezpieczeniach. Helion 2022
3. G. Khawaja, Kali Linux i testy penetracyjne. Biblia. Helion 2022
4. L. Kohnfelder, Po pierwsze: bezpieczeństwo. Przewodnik dla twórców oprogramowania. Helion 2022

Źródła dodatkowe

1. <https://zaufanatrzeciastrona.pl/>
2. <https://www.enisa.europa.eu>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

- Zajęcia z udziałem nauczyciela akademickiego.....60 godz.....32 godz.

- Zapoznanie się z literaturą przedmiotu..... 15 godz.....25 godz.
- Przygotowanie się do zajęć..... 15 godz.....25 godz.
- Wypełnienie dokumentacji zajęć..... 10 godz..... 18 godz.
- Suma obciążeń:.....100 godz.....100 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia przedmiotu jest uzyskania ponad 50% poprawnych odpowiedzi na teście obejmującym informacje teoretyczne oraz uzyskanie co najmniej 50% punktów z realizacji ćwiczeń praktycznych. Ocena końcowa wyliczana z jest na podstawie znormalizowanych liczb punktów z poszczególnych form zajęć z odpowiadającymi im wagami.

Front-end developer

Nazwa zajęć po angielsku: Front-end developer

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: V

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 6

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Specjalność: Aplikacje internetowe i mobilne

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Marcin Zawadzki

Opis zajęć:

Wprowadzenie do front-end developmentu:

Rola front-end developera

Podstawowe technologie: HTML, CSS, JavaScript

HTML i semantyka:

- Struktura dokumentu HTML
- Semantyczne znaczniki HTML5
- Formularze i walidacja

Stylizacja za pomocą CSS:

- Podstawy CSS
- Flexbox i Grid Layout
- Preprocesory CSS (Sass, LESS)

Zaawansowane techniki CSS:

- Animacje i przejścia
- Responsywność i media queries
- Frameworki CSS (Bootstrap, Tailwind CSS)

JavaScript:

- Podstawy JavaScript
- Manipulacja DOM
- Eventy i ich obsługa

Nowoczesny JavaScript (ES6+):

- Let, Const, Arrow Functions
- Moduły i klasy
- Asynchroniczność (Promises, Async/Await)

Frameworki i biblioteki JavaScript:

- React: komponenty, stan, propsy

Narzędzia i workflow:

- Systemy kontroli wersji (Git)

Optymalizacja i wydajność:

- Minifikacja i kompresja
- Lazy loading i code splitting

Praktyczne projekty:

- Tworzenie portfolio
- Praca z API i integracja z backendem

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Opisuje przypadki użycia technologii front-end w rzeczywistych projektach.	K_W03
Ocena: Odpowiedzi indywidualne na zajęciach, w których student opisuje przypadki użycia	

technologii front-end.	
Wyjaśnia zastosowanie bibliotek i frameworków front-end w tworzeniu aplikacji webowych. Ocena: Odpowiedzi indywidualne na zajęciach, w których student wyjaśnia zastosowanie konkretnych bibliotek i frameworków.	K_W05
Wymienia i opisuje standardy HTML, CSS i JavaScript. Ocena: Odpowiedzi indywidualne na zajęciach, w których student wymienia i opisuje standardy HTML, CSS i JavaScript.	K_W06
Umiejętności (student potrafi)	
Rozwiązuje złożone problemy związane z responsywnością i optymalizacją aplikacji webowych. Ocena: Projekt indywidualny, w którym student rozwiązuje problemy związane z responsywnością i optymalizacją.	K_U01
Dobiera odpowiednie narzędzia do testowania i debugowania aplikacji front-end. Ocena: Praktyczne zadanie laboratoryjne, w którym student dobiera i stosuje narzędzia do testowania i debugowania.	K_U04
Implementuje funkcjonalności interaktywne w aplikacjach webowych. Ocena: Projekt indywidualny, w którym student implementuje interaktywne funkcjonalności.	K_U05
Projektuje i tworzy aplikację webową z użyciem HTML, CSS i JavaScript. Ocena: Projekt indywidualny, w którym student projektuje i tworzy aplikację webową.	K_U11
Kompetencje społeczne (student jest gotów do)	
Analizuje i ocenia własne projekty pod kątem jakości kodu i zgodności z najlepszymi praktykami. Ocena: Odpowiedzi indywidualne na zajęciach, w których student krytycznie ocenia swoje projekty.	K_K01
Przestrzega zasad etyki zawodowej podczas realizacji projektów zespołowych. Ocena: Ocena pracy w grupie oraz obserwacja zachowań etycznych podczas realizacji projektu.	K_K05

Metody i formy kształcenia:

- Metoda ćwiczeń
- Metoda pokazu
- Praca ze źródłem elektronicznym
- Pogadanka
- Metoda projektu
- Wykład problemowy

Metody oceniania:

- Rozliczenie projektu
- Aktywność w trakcie zajęć

Wykład konwersatoryjny (W)

1. Wprowadzenie do Pełnego Stacku Web Developmentu
2. Podstawy React
3. Zaawansowane Koncepcje w React
4. Wprowadzenie do Spring Boot

5. Tworzenie RESTful API z Spring Boot
6. Integracja z Bazą Danych
7. Bezpieczeństwo i Walidacja
8. Wdrażanie i Testowanie Aplikacji

Ćwiczenia (Ć)

1. Konfiguracja Środowiska Pracy
2. Tworzenie Pierwszej Aplikacji w React
3. Zarządzanie Stanem w React
4. Routing w React
5. Tworzenie Pierwszego Projektu Spring Boot
6. Integracja z Bazą Danych
7. Bezpieczeństwo w Aplikacjach Webowych
8. Testowanie i Wdrażanie Aplikacji

Projekt (P)

1. Planowanie Projektu
2. Implementacja Front-endu
3. Implementacja Back-endu
4. Prezentacja i Ocena Projektu

Literatura podstawowa

1. "Learning React: Modern Patterns for Developing React Apps" - Alex Banks, Eve Porcello
2. "Spring Boot in Action" - Craig Walls
3. "Spring Boot: Up and Running" - Mark Heckler

Literatura uzupełniająca

1. "Pro React 16" - Adam Freeman
2. "Spring Microservices in Action" - John Carnell
3. "JavaScript: The Good Parts" - Douglas Crockford

Źródła dodatkowe

1. Oficjalna dokumentacja React: <https://react.dev/learn>
2. Oficjalna dokumentacja Spring Boot: <https://docs.spring.io/spring-boot/index.html>
3. Repozytoria GitHub: <https://github.com/enaqx/awesome-react> oraz <https://github.com/ityouknow/awesome-spring-boot>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	10 godz.
• Samodzielne wykonanie zadań.....	12 godz.....	26 godz.
• Przygotowanie projektu.....	46 godz.....	70 godz.
• Przygotowanie się do zajęć.....	6 godz.....	3 godz.
• Wypełnienie dokumentacji zajęć.....	1 godz.....	1 godz.
Suma obciążeń:.....	150 godz.....	150 godz.

Zaliczenie zajęć:

Stworzenie projektu praktycznego:

Temat projektu: Implementacja aplikacji webowej z wykorzystaniem technologii front-end (React) oraz back-end (Spring Boot).

Opis projektu:

Front-end: Aplikacja powinna być stworzona w oparciu o bibliotekę React. Powinna zawierać co najmniej trzy komponenty, interakcje z użytkownikiem oraz responsywny design.

Back-end: Aplikacja powinna korzystać z REST API stworzonego w technologii Spring Boot. API powinno obsługiwać co najmniej trzy różne zasoby (np. użytkownicy, produkty, zamówienia) oraz implementować podstawowe operacje

CRUD (Create, Read, Update, Delete).

Szczegółowe wymagania dotyczące projektu:

- Front-end:

- Użycie React do zarządzania stanem aplikacji (np. poprzez Context API lub Redux).
- Implementacja routingu z użyciem React Router.
- Formularze walidowane po stronie klienta.
- Stylizacja aplikacji z użyciem CSS (preferowane użycie CSS-in-JS lub preprocesorów CSS).

- Back-end:

- Stworzenie RESTful API w Spring Boot.
- Użycie bazy danych (np. H2, MySQL, PostgreSQL) do przechowywania danych.
- Implementacja warstwy serwisowej oraz kontrolerów.
- Obsługa błędów i walidacja danych po stronie serwera.

Kryteria oceny projektu:

- Poprawność działania: Aplikacja powinna działać zgodnie z założeniami i spełniać wszystkie wymagania funkcjonalne.
- Jakość kodu: Kod powinien być czytelny, dobrze zorganizowany i zgodny z najlepszymi praktykami programistycznymi.
- Interfejs użytkownika: Aplikacja powinna być intuicyjna i przyjazna dla użytkownika.
- Dokumentacja: Projekt powinien zawierać dokumentację opisującą sposób uruchomienia aplikacji oraz instrukcję użytkowania.

Prezentacja projektu:

- Każdy student jest zobowiązany do przedstawienia swojego projektu podczas zajęć.
- Prezentacja powinna obejmować:
 - Krótkie wprowadzenie do projektu.
 - Demonstrację działania aplikacji.
 - Omówienie kluczowych elementów implementacji.
 - Odpowiedzi na pytania prowadzącego oraz innych studentów.

Ostateczny termin oddania projektu:

- Projekt należy oddać najpóźniej do najpóźniej w dniu ostatnich zajęć zgodnie z rozkładem.
- Wszelkie opóźnienia w oddaniu projektu mogą skutkować obniżeniem oceny końcowej.

Sposób oddania projektu:

- Projekt należy przesłać w formie repozytorium Git (np. GitHub, GitLab, Bitbucket).
- Repozytorium powinno zawierać pełny kod źródłowy aplikacji oraz plik README z instrukcją uruchomienia.

Uwagi końcowe:

- Wszelkie pytania i wątpliwości dotyczące projektu należy kierować do prowadzącego zajęcia.
- Współpraca w grupach jest dozwolona, jednak każdy student musi przedstawić indywidualną część projektu oraz wykazać się samodzielną pracą.

Internetowe technologie informatyczne

Nazwa zajęć po angielsku: Internet information technologies

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 6

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Specjalność: Aplikacje internetowe i mobilne

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

mgr Marcin Zawadzki

Opis zajęć:

Przedmiot „Internetowe technologie informatyczne” ma na celu zapoznanie studentów z teorią i praktyką tworzenia nowoczesnych, interaktywnych aplikacji internetowych. W trakcie kursu uczestnicy zdobędą kompleksową wiedzę na temat architektury klient-serwer, projektowania i implementacji usług sieciowych (API REST) oraz technologii wykorzystywanych zarówno po stronie przeglądarki (front-end), jak i serwera (back-end). Szczególny nacisk położony jest na zdobycie praktycznych umiejętności poprzez realizację zadań programistycznych oraz budowę kompletnej aplikacji internetowej w ramach semestralnego projektu. Kurs obejmuje również zagadnienia bezpieczeństwa aplikacji webowych, dostępności cyfrowej (WCAG), testowania oraz wdrażania i utrzymania aplikacji. Celem jest przygotowanie studentów do pracy w dynamicznym środowisku IT poprzez naukę aktualnych i pożądanych na rynku technologii, w tym TypeScript, React, Node.js, Express, MongoDB oraz narzędzi współpracy zespołowej. Uzasadnienie doboru technologii: React, Node.js, Express i MongoDB zostały wybrane ze względu na ich szerokie zastosowanie w branży IT, spójny ekosystem JavaScript/TypeScript po stronie klienta i serwera oraz dostępność dokumentacji i materiałów dydaktycznych. TypeScript wprowadzono jako standard branżowy zwiększający bezpieczeństwo typów i czytelność kodu. Projekt semestralny integruje wszystkie technologie w realnym scenariuszu inżynierskim.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
Wyjaśnia architekturę aplikacji internetowej, model klient-serwer, działanie HTTP/HTTPS oraz przepływ danych między front-endem, back-endem i bazą danych.	K_W01
Charakteryzuje technologie i narzędzia stosowane do budowy aplikacji internetowych, w tym React, Node.js, Express, MongoDB, Git i narzędzia testowania.	K_W02
Wyjaśnia znaczenie standardów i dobrych praktyk projektowania REST API, dokumentacji OpenAPI, dostępności WCAG 2.2 oraz bezpieczeństwa aplikacji webowych zgodnie z OWASP Top 10.	K_W03
Wyjaśnia zasady legalnego wykorzystania bibliotek, frameworków i komponentów open-source w projekcie internetowym.	K_W04
Wyjaśnia proces wdrażania, konfiguracji środowisk i utrzymania aplikacji webowej w cyklu życia oprogramowania.	K_W04
Analizuje strategie testowania aplikacji internetowych (testy jednostkowe, integracyjne, API, end-to-end) oraz ich znaczenie dla jakości oprogramowania.	K_W05
Umiejętności (student potrafi)	
Projektuje REST API wraz z zasobami, metodami HTTP, kodami odpowiedzi, walidacją	K_U01

danych i dokumentacja OpenAPI.	
Implementuje logikę serwerową aplikacji z użyciem Node.js, Express i bazy danych MongoDB.	K_U02
Tworzy interaktywny interfejs użytkownika z użyciem React, TypeScript, routingu, formularzy i obsługi komunikacji z API.	K_U03
Integruje front-end, back-end i bazy danych, stosując testy, obsługę błędów, walidację i repozytorium Git.	K_U04
Wyszukuje, analizuje i wykorzystuje dokumentację technologiczną do implementacji funkcjonalności i rozwiązywania problemów występujących w aplikacjach internetowych.	K_U05
Kompetencje społeczne (student jest gotów do)	
Student jest gotów do krytycznej oceny jakości własnego kodu, bezpieczeństwa, dostępności i rozwiązań innych członków zespołu.	K_K01
Student jest gotów do działania przedsiębiorczego przy proponowaniu funkcjonalności odpowiadających na potrzeby użytkowników i ograniczenia techniczne projektu.	K_K02
Student jest gotów do przestrzegania zasad etyki zawodowej, licencji open-source, prywatności użytkowników i odpowiedzialnego korzystania z narzędzi AI.	K_K03
Student jest gotów do samorozwoju w zakresie dynamicznie zmieniających się technologii internetowych.	K_K04

Metody i formy kształcenia:

- Pogadanka
- Metoda ćwiczeń
- Metoda pokazu
- Metoda projektu
- Wykład informacyjny
- Dyskusja

Metody oceniania:

- Rozliczenie projektu
- Zaliczenie praktyczne
- Zaliczenie testowe

Wykład konwersatoryjny (W)

1. Architektura aplikacji internetowych: klient-serwer, SPA, REST, API, baza danych.
2. HTTP/HTTPS: metody, nagłówki, statusy, cookies, CORS, cache.
3. Projektowanie REST API: zasoby, endpointy, walidacja, błędy, dokumentacja OpenAPI, licencje.
4. Node.js LTS i Express: routing, middleware, obsługa błędów, konfiguracja.
5. MongoDB: model dokumentowy, schematy, walidacja, relacje, indeksy.
6. React i TypeScript: komponenty, stan, formularze, routing, integracja z API.
7. Bezpieczeństwo aplikacji: OWASP Top 10, XSS, CSRF, kontrola dostępu, sekrety, walidacja.
8. Dostępność i użyteczność: WCAG 2.2, semantyka HTML, obsługa klawiaturą, kontrast.
9. Testowanie: testy jednostkowe, integracyjne, API, end-to-end.
10. Wdrażanie i utrzymanie: środowiska, CI/CD, logi, monitoring, dokumentacja.

Ćwiczenia (Ć)

1. Konfiguracja środowiska: Node.js, npm/pnpm, Git, edytor, devtools.
2. Serwer Express: routing, middleware, walidacja, obsługa błędów.
3. MongoDB: model danych, operacje CRUD, walidacja, seed danych.
4. Dokumentowanie API w OpenAPI (Swagger).
5. React + TypeScript: komponenty, formularze, walidacja.
6. Integracja front-endu z API, obsługa błędów i stanów ładowania.
7. Testy API i komponentów (Vitest, Testing Library).
8. Samodzielna implementacja małej funkcjonalności full-stack z code review.
9. Bezpieczeństwo: walidacja, CORS, ochrona sekretów, uprawnienia.

10. Mini-funkcjonalność full-stack z code review.

Projekt (P)

1. Definicja tematu, wymagań i kryteriów akceptacji.
2. Repozytorium Git, backlog i podział zadań.
3. Projekt API i dokumentacja OpenAPI.
4. Implementacja back-endu i bazy danych.
5. Implementacja front-endu.
6. Testy, dostępność, bezpieczeństwo.
7. README, instrukcja uruchomienia, zmienne środowiskowe.
8. Prezentacja, obrona decyzji projektowych i raport wkładu.

Literatura podstawowa

1. Vanderkam D., "TypeScript. Skuteczne programowanie, wyd. II". Promise 2024
2. Flanagan D., "JavaScript. Przewodnik. Wydanie VII". Wydawnictwo Helion, 2021.
3. Ater T., Progresywne aplikacje webowe, Promise 2018

Literatura uzupełniająca

1. Martin, R. C., "Czysty kod. Podręcznik dobrego programisty". Wydawnictwo Helion, 2018
2. Haverbeke M., "Zrozumieć JavaScript. Wprowadzenie do programowania. Wydanie III". Wydawnictwo Helion, 2020
3. Sommerville I., "Inżynieria oprogramowania". Wydawnictwo WNT.
4. Oficjalna dokumentacja React (react.dev)
5. Oficjalna dokumentacja Node.js LTS (nodejs.org/docs)
6. Oficjalna dokumentacja Express (expressjs.com)
7. Oficjalna dokumentacja MongoDB (mongodb.com/docs)

Źródła dodatkowe

1. MDN Web Docs (Mozilla Developer Network) – obszerna dokumentacja technologii webowych
2. Serwisy edukacyjne: freeCodeCamp, The Odin Project
3. OpenAPI Specification (swagger.io/specification)
4. OWASP Top 10 (owasp.org/www-project-top-ten)
5. WCAG 2.2 - Web Content Accessibility Guidelines (w3.org/TR/WCAG22)
6. Dokumentacja Vitest (vitest.dev) i Testing Library (testing-library.com)

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Zapoznanie się z literaturą przedmiotu.....	10 godz.....	12 godz.
• Samodzielne wykonanie zadań.....	25 godz.....	38 godz.
• Przygotowanie projektu.....	25 godz.....	40 godz.
• Przygotowanie się do kolokwium.....	5 godz.....	5 godz.
• Przygotowanie się do zajęć.....	5 godz.....	8 godz.
• Wypełnienie dokumentacji zajęć.....	5 godz.....	7 godz.
Suma obciążeń:.....	150 godz.....	150 godz.

Zaliczenie zajęć:

Ostateczna ocena z przedmiotu jest średnią ważoną, odzwierciedlającą wkład pracy we wszystkie formy zajęć oraz stopień osiągnięcia założonych efektów uczenia się.

Warunkiem koniecznym do uzyskania zaliczenia jest otrzymanie pozytywnych ocen (minimum 51% punktów) ze wszystkich trzech komponentów: wykładu, ćwiczeń i projektu.

Wagi poszczególnych komponentów:

Projekt: 50%

Ćwiczenia: 30%

Wykład: 20%

Warunki zaliczenia wykładu:

Zaliczenie testowe końcowe (obejmujące materiał z wykładów): 100% oceny z wykładu.

Weryfikuje efekty: PEU_W01–PEU_W06.

Warunki zaliczenia ćwiczeń:

Kolokwium praktyczne przy komputerze: 100% oceny z ćwiczeń.

Weryfikuje efekty: PEU_U01–PEU_U05.

Warunki zaliczenia projektu:

Rozliczenie projektu, na które składa się:

- Działająca aplikacja full-stack (20%)
- API i baza danych – poprawność, walidacja, obsługa błędów (15%)
- Front-end – funkcjonalność, responsywność, UX (15%)
- Testy – jednostkowe, integracyjne lub API (15%)
- Bezpieczeństwo – walidacja, uwierzytelnianie, ochrona danych (10%)
- Dostępność – zgodność z WCAG 2.2 (10%)
- Dokumentacja – README, OpenAPI, instrukcja uruchomienia (10%)
- Prezentacja i obrona decyzji projektowych (5%)

Weryfikuje efekty: PEU_U01–PEU_U06, PEU_K01–PEU_K04.

Próg osiągnięcia efektów uczenia się:

Każdy efekt uczenia się uznaje się za osiągnięty, jeżeli student uzyskał minimum 51% punktów z przypisanych do tego efektu zadań weryfikujących.

Skala ocen:

91% – 100%: bardzo dobry (5.0)

81% – 90%: dobry plus (4.5)

71% – 80%: dobry (4.0)

61% – 70%: dostateczny plus (3.5)

51% – 60%: dostateczny (3.0)

50% i poniżej: niedostateczny (2.0)

Programowanie aplikacji mobilnych

Nazwa zajęć po angielsku: Mobile application development

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VI

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 6

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Specjalność: Aplikacje internetowe i mobilne

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....30 / 16 godz.....0 / 0 godz.
- Ćwiczenia.....30 / 16 godz.....0 / 0 godz.
- Projekt.....15 / 8 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot Programowanie aplikacji mobilnych stanowi zaawansowany komponent praktycznego profilu kształcenia w zakresie informatyki. Jego celem jest przygotowanie studentów do samodzielnego projektowania, implementacji oraz testowania aplikacji mobilnych. Zakres treści obejmuje budowę interfejsów użytkownika, wykorzystanie baz danych (SQLite, Room), integrację z zewnętrznymi usługami API oraz zarządzanie cyklem życia aplikacji. Uczestnicy zajęć rozwiną umiejętności praktyczne w środowisku Android Studio oraz nauczą się stosować wzorce projektowe i dobre praktyki inżynierii oprogramowania mobilnego. Szczególny nacisk położono na aktywne formy pracy, zadania problemowe oraz realizację projektów zespołowych.

Efekty uczenia się:

<i>Wiedza (student zna i rozumie)</i>	
wyjaśnia zastosowania aplikacji mobilnych Android w systemach informatycznych, usługach cyfrowych, rozwiązaniach biznesowych i integracji z API	K_W03
charakteryzuje cykl życia aplikacji i komponentów Android oraz procesy zachodzące w aplikacji mobilnej - stan UI, nawigację, uprawnienia, przechowywanie danych, komunikację sieciową i publikację	K_W04
identyfikuje metody, techniki i narzędzia programowania mobilnego: Kotlin, Android Studio, Gradle, Jetpack Compose, Material 3, ViewModel, Room, Retrofit/Ktor, Git i emulator	K_W05
rozpoznaje standardy i dobre praktyki tworzenia aplikacji mobilnych, w tym target SDK, Material Design, accessibility, privacy, OWASP MASVS, bezpieczne przechowywanie danych i zasady publikacji w Google Play	K_W06
<i>Umiejętności (student potrafi)</i>	
pozyskuje i wykorzystuje dokumentację Android Developers, Kotlin, Jetpack, Material Design, Room, Retrofit/Ktor i Google Play do rozwiązywania problemów projektowych	K_U01
integruje wiedzę z programowania, baz danych, UI/UX, sieci, API, bezpieczeństwa i pracy zespołowej podczas realizacji aplikacji mobilnej	K_U02
dobiera architekturę, biblioteki, komponenty Jetpack, narzędzia testowania i sposób przechowywania danych do wymagań aplikacji mobilnej	K_U04
rozwiązuje praktyczny problem projektowy przez implementację aplikacji mobilnej z interfejsem użytkownika, logiką, lokalną bazą danych, integracją API i obsługą błędów	K_U05
wykorzystuje metody analityczne, symulacyjne i eksperymentalne do testowania aplikacji na	K_U07

emulatorze i urządzeniu, weryfikacji działania API, danych, uprawnień i wybranych scenariuszy użytkownika	
projektuje i realizuje aplikację mobilną Android z użyciem właściwych metod, technik i narzędzi, dokumentując jej architekturę i sposób uruchomienia	K_U11
stosuje dobre praktyki programowania mobilnego, bezpieczeństwa, prywatności, dostępności, testowania, kontroli wersji i publikacji aplikacji	K_U12
korzysta ze źródeł angielskojęzycznych podczas przygotowywania się do zajęć oraz realizacji projektu	K_U16
planuje i organizuje pracę własną oraz zespołu projektowego z użyciem repozytorium, tablicy zadań, przeglądu kodu i dokumentacji	K_U17
współdziała z innymi osobami przy projektowaniu, implementacji, testowaniu i prezentacji aplikacji mobilnej	K_U18
Kompetencje społeczne (student jest gotów do)	
jest gotów do zasięgania opinii ekspertów, prowadzącego i członków zespołu w przypadku trudności technicznych, projektowych, bezpieczeństwa lub publikacji aplikacji	K_K02
jest gotów do przestrzegania zasad etyki zawodowej, licencji, ochrony danych, prywatności użytkowników i uczciwego wykorzystania cudzych zasobów oraz AI	K_K05
jest gotów do samorozwoju w zakresie zmieniających się wersji Androida, bibliotek Jetpack, wymagań Google Play, bezpieczeństwa i praktyk mobile developmentu	U_K07

Metody i formy kształcenia:

- Wykład informacyjny
- Pogadanka
- Dyskusja
- Studium przypadku
- Metoda ćwiczeń
- Metoda projektu

Metody oceniania:

- Zaliczenie testowe
- Zaliczenie praktyczne
- Rozliczenie projektu

Wykład konwersatoryjny (W)

1. Architektura mobilnych systemów operacyjnych.
2. Ekosystem Android: Android Studio, Gradle, Kotlin, emulator, urządzenie fizyczne, Google Play.
3. Kotlin w Androidzie: null-safety, data classes, lambdy, coroutines, podstawy Flow.
4. Architektura aplikacji: UI layer, data layer, repository, opcjonalnie domain layer, MVVM/MVI, unidirectional data flow.
5. Jetpack Compose i Material 3: composables, state, recomposition, navigation, adaptive UI.
6. Cykl życia aplikacji i zarządzanie stanem: ViewModel, SavedStateHandle, lifecycle-aware components.
7. Dane lokalne: Room, DataStore, migracje, repozytorium, cache.
8. Sieć i API: Retrofit lub Ktor, JSON, obsługa błędów, timeouts, offline-first, retry.
9. Uprawnienia i funkcje urządzenia: geolokalizacja, sensory, aparat, powiadomienia, polityki prywatności.
10. Bezpieczeństwo mobilne: OWASP MASVS, Keystore, szyfrowanie, tokeny, TLS, sekrety, bezpieczne logowanie.
11. Testowanie: unit tests, UI tests, testy ViewModel, testy integracji API, testy na emulatorze i urządzeniu.
12. Jakość: logging, crash reporting, performance, battery, accessibility, localization.
13. Publikacja i utrzymanie: target SDK, podpisywanie aplikacji, Play Console, release notes, polityki Google Play.
14. Wprowadzenie do alternatyw: Flutter/React Native/Kotlin Multiplatform — porównawczo, nie zamiast rdzenia Androida.

Ćwiczenia (Ć)

1. Konfiguracja projektu Android Studio, Gradle, repozytorium Git.
2. Pierwsza aplikacja Kotlin + Jetpack Compose.
3. Material 3: motyw, komponenty, formularze, walidacja.

4. Nawigacja między ekranami i przekazywanie parametrów.
5. ViewModel, stan UI, zdarzenia, obsługa rotacji i procesu.
6. Room: encje, DAO, relacje, migracje, Flow.
7. DataStore: preferencje użytkownika i ustawienia aplikacji.
8. Integracja z API: klient HTTP, DTO, mapowanie, obsługa błędów.
9. Uprawnienia runtime i geolokalizacja / wybrany sensor.
10. Bezpieczeństwo: przechowywanie tokenu, brak sekretów w repozytorium, Keystore, logi.
11. Testy jednostkowe ViewModel/repository i proste testy UI Compose.
12. Przygotowanie builda, podpisanie, README i instrukcja uruchomienia.

Projekt (P)

1. Wybór problemu i wymagań funkcjonalnych/niefunkcjonalnych.
2. Backlog, role zespołowe, repozytorium, tablica zadań.
3. Projekt UI/UX w Material 3, makiety ekranów, flow użytkownika.
4. Architektura aplikacji: UI/data/domain, repozytoria, źródła danych.
5. Implementacja aplikacji z co najmniej: UI Compose, lokalną bazą Room/DataStore, integracją API, walidacją, obsługą błędów.
6. Element bezpieczeństwa: uprawnienia, prywatność, bezpieczne przechowywanie danych lub komunikacja.
7. Testy i weryfikacja na emulatorze oraz urządzeniu.
8. Dokumentacja: README, instrukcja uruchomienia, architektura, opis API, baza danych, znane ograniczenia.
9. Prezentacja projektu, rozliczenie indywidualnego wkładu.

Literatura podstawowa

1. <https://developer.android.com/courses>
2. <https://developers.google.com/?hl=pl>

Literatura uzupełniająca

1. M. Płonkowski, Android Studio. Tworzenie aplikacji mobilnych. Helion 2018
2. D. Griffiths, D. Griffiths, Android. Programowanie aplikacji. Rusz głową! Wydanie II. Helion 2018
3. B. Phillips, C. Stewart, K. Marsicano, Programowanie aplikacji dla Androida. The Big Nerd Ranch Guide. Wydanie III. Helion 2017
4. J. Lehtimäki, Android UI. Podręcznik dla projektantów. Smashing Magazine. Helion 2013

Źródła dodatkowe

1. <https://www.springboard.com/resources/learning-paths/android/>
2. <https://www.codecademy.com/catalog/subject/mobile-development>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	75 godz.....	40 godz.
• Zapoznanie się z literaturą przedmiotu.....	15 godz.....	25 godz.
• Przygotowanie się do zajęć.....	15 godz.....	25 godz.
• Przygotowanie projektu.....	30 godz.....	45 godz.
• Przygotowanie się do zaliczenia końcowego.....	15 godz.....	15 godz.
Suma obciążeń:.....	150 godz.....	150 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest, co najmniej w stopniu podstawowym, przyswojenie wiadomości na temat systemów mobilnych, zdobycie umiejętności projektowania i wykonania aplikacji funkcjonującej na urządzeniach przenośnych oraz nabycie kompetencji polegających na grupowym rozwiązywaniu problemów zawodowych. Zaliczenie części wykładowej realizowane jest w formie testu, części ćwiczeniowej w formie praktycznego zadania do wykonania w Android Studio oraz projektowej poprzez grupową realizację zadania praktycznego. Ocena końcowa zależy od liczby punktów zdobytych z testowego zaliczenia wykładu (30%), wykonanej aplikacji w trakcie zaliczenia ćwiczeń (40%) oraz wykonanej pracy projektowej (30%). Ocena końcowa wyliczana jest w stosunku do maksymalnej liczby możliwych do uzyskania punktów i wynosi:

0% - 50% - niedostateczny
51%-60% - dostateczny

61%-70% - dostateczny plus

71%-80% - dobry

81%-90% - dobry plus

91%-100% - bardzo dobry.

Projektowanie interfejsów graficznych

Nazwa zajęć po angielsku: Graphic interface design

Kierunek: Informatyka

Obowiązuje od roku akademickiego: 2024/2025

Poziom: I stopnia

Profil: praktyczny

Semestr: VII

Forma zaliczenia: Zaliczenie na ocenę

Punkty ECTS: 4

Materiały: materiały na platformie

Język zajęć: polski

Zajęcia do wyboru: Tak

Specjalność: Aplikacje internetowe i mobilne

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych (w tym e-learning):

- Wykład konwersatoryjny.....15 / 8 godz.....0 / 0 godz.
- Ćwiczenia.....45 / 24 godz.....0 / 0 godz.

Nazwiska osób odpowiedzialnych za zajęcia:

dr inż. Robert Żak

Opis zajęć:

Przedmiot skierowany jest do studentów kierunku Informatyka, którzy ukończyli przedmioty wprowadzające z zakresu programowania i technologii webowych. Celem zajęć jest nabycie umiejętności projektowania estetycznych, ergonomicznych i dostępnych interfejsów graficznych z wykorzystaniem aktualnych narzędzi i metod. Student zdobędzie wiedzę na temat historii i teorii interfejsów graficznych, kolorystyki, typografii, tekstur, struktury informacji, projektowania zorientowanego na użytkownika oraz wzorców projektowych. Nabędzie praktyczne umiejętności w zakresie organizacji przestrzeni, doboru typografii i kolorów, projektowania interfejsów dla osób z niepełnosprawnościami, a także tworzenia nowoczesnych rozwiązań dla urządzeń mobilnych. Zajęcia wspierane będą przez środowiska programistyczne oraz graficzne, takie jak Figma, Uizard, Motiff, Creatie, Visual Studio Code, co pozwoli na zdobycie kompetencji technicznych.

Efekty uczenia się:

Wiedza (student zna i rozumie)	
charakteryzuje genezę, rozwój i zastosowania graficznych interfejsów użytkownika w systemach informatycznych	K_W03
wyjaśnia wpływ koloru, typografii, hierarchii wizualnej, organizacji treści i architektury informacji na percepcję oraz obsługę interfejsu	K_W04
identyfikuje metody, techniki i narzędzia projektowania GUI, w tym makietowanie, prototypowanie, audyt heurystyczny, testy użyteczności, Figma, design systems i narzędzia wspierane AI	K_W05
rozpoznaje standardy i dobre praktyki projektowania interfejsów, w tym WCAG 2.2, EN 301 549, ISO 9241-210, Material Design 3, Apple HIG i Microsoft Design	K_W06
wyjaśnia społeczne znaczenie dostępności cyfrowej, inkluzywności i projektowania interfejsów dla różnych grup użytkowników, w tym osób z niepełnosprawnościami i seniorów	K_W07
krytycznie ocenia rozwój interfejsów graficznych w kontekście potrzeb użytkowników	U_W11
Umiejętności (student potrafi)	
integruje dane z analizy użytkowników, heurystyk, checklist UX, testów i standardów dostępności w celu rozwiązania problemu projektowego	K_U02
dobiera narzędzia do projektowania, prototypowania, testowania i dokumentowania interfejsów graficznych zależnie od platformy, użytkownika i celu systemu	K_U04
projektuje interfejs użytkownika z wykorzystaniem zasad kompozycji, typografii, koloru, wzorców projektowych, dostępności i teorii UX	K_U05

planuje i przeprowadza test użyteczności albo audyt heurystyczny interfejsu, interpretuje wyniki i formułuje zalecenia projektowe	K_U06
uwzględnia aspekty dostępności, inkluzywności, etyki, ograniczeń poznawczych i konsekwencji decyzji projektowych dla użytkownika końcowego	K_U08
projektuje i dokumentuje prototyp graficznego interfejsu aplikacji webowej, mobilnej, desktopowej lub wearable z użyciem właściwych metod, technik i narzędzi	K_U11
Kompetencje społeczne (student jest gotów do)	
jest gotów do inicjowania działań na rzecz dostępności cyfrowej i przeciwdziałania wykluczeniu użytkowników w projektowanych rozwiązaniach	K_K03
jest gotów do przestrzegania zasad etyki zawodowej w projektowaniu interfejsów, w tym odpowiedzialności za dostępność, przejrzystość, brak manipulacji i poszanowanie użytkownika	K_K05
jest gotów do samorozwoju w zakresie nowych narzędzi projektowania, standardów dostępności, wzorców GUI i technologii interakcji człowiek–komputer	U_K07

Metody i formy kształcenia:

- Wykład problemowy
- Dyskusja
- Studium przypadku
- Metoda ćwiczeń

Metody oceniania:

- Zaliczenie testowe
- Zaliczenie praktyczne
- Praca domowa

Wykład konwersatoryjny (W)

1. Ewolucja GUI: desktop, web, mobile, wearables, interfejsy dotykowe, głosowe i multimodalne.
2. Human-centred design i ISO 9241-210: potrzeby użytkownika, kontekst użycia, wymagania, iteracja.
3. Architektura informacji: struktura treści, hierarchia, nawigacja, user flows, model mentalny użytkownika.
4. Kompozycja wizualna: siatki, odstępy, Gestalt, hierarchia, balans, kontrast, rytm.
5. Kolor i typografia: czytelność, kontrast, skala typograficzna, fonty, semantyka koloru, dark mode.
6. Wzorce GUI: formularze, listy, tabele, karty, modal, nawigacja, onboarding, dashboard, wyszukiwarka, empty/error/loading states.
7. Design systems: komponenty, warianty, stany, tokeny projektowe, biblioteki, dokumentacja i spójność produktu.
8. Dostępność cyfrowa: WCAG, EN 301 549, EAA, klawiatura, kontrast, focus, tekst alternatywny, target size, dostępne formularze.
9. Projektowanie responsywne i adaptacyjne: mobile-first, desktop, tablet, wearables, foldables.
10. Testowanie GUI: audyt heurystyczny, test użyteczności, checklisty, metryki, iteracja.
11. Przekazanie projektu do implementacji: specyfikacja komponentów, stany, style, tokeny, assets, accessibility annotations.
12. Etyka projektowania: dark patterns, manipulacja, inkluzywność, prywatność, odpowiedzialność projektanta.

Ćwiczenia (Ć)

1. Analiza porównawcza interfejsów aplikacji typu desktop, mobilnych, webowych.
2. Audyt heurystyczny istniejącego interfejsu.
3. Audyt dostępności wybranego ekranu według WCAG 2.2.
4. Projekt struktury informacji i user flow.
5. Makiety low-fidelity i wireframes.
6. Projekt wizualny: kolor, typografia, siatka, komponenty.
7. Projekt komponentów i stanów: default, hover, focus, disabled, error, loading, empty.
8. Prototyp interaktywny w Figma lub narzędziu podobnym.
9. Projekt dla grupy użytkowników, np. seniorzy, osoby słabowidzące, użytkownicy mobilni.
10. Test użyteczności lub test ekspercki prototypu.
11. Iteracja projektu po testach.

12. Dokumentacja dla programisty: komponenty, style, tokeny, dostępność, specyfikacja zachowań.

Literatura podstawowa

1. J. Tidwell, C. Brewer, A. Valencia-Brooks, Projektowanie interfejsów. Sprawdzone wzorce projektowe. Wydanie III. Helion 2021
2. S. Krug, Nie każ mi myśleć! O życiowym podejściu do funkcjonalności stron internetowych. Wydanie III. helion 2014

Literatura uzupełniająca

1. J. Nielsen, R. Budiu, Funkcjonalność aplikacji mobilnych. Nowoczesne standardy UX i UI. Helion 2013
2. C. Badura, UXUI. Design Zoptymalizowany. Manual Book. Helion 2019
3. C. Badura, UXUI. Design Zoptymalizowany. Workshop Book. Helion 2019

Źródła dodatkowe

1. <https://help.figma.com/>
2. <https://m3.material.io/>
3. <https://developer.apple.com/design/human-interface-guidelines/>
4. <https://learn.microsoft.com/en-us/windows/apps/design/>
5. <https://webaim.org/>
6. <https://www.nngroup.com/>

Obciążenie pracą studenta stacjonarnego i niestacjonarnego:

• Zajęcia z udziałem nauczyciela akademickiego.....	60 godz.....	32 godz.
• Przygotowanie się do zajęć.....	10 godz.....	22 godz.
• Samodzielne wykonanie zadań.....	15 godz.....	30 godz.
• Przygotowanie się do zaliczenia końcowego.....	15 godz.....	16 godz.
Suma obciążeń:.....	100 godz.....	100 godz.

Zaliczenie zajęć:

Warunkiem zaliczenia jest przyswojenie wiadomości na temat interfejsów graficznych i zdobycie umiejętności ich projektowania zgodnie ze standardami użyteczności oraz dostępności. Ocena końcowa zależy od liczby punktów zdobytych z realizowanych w trakcie ćwiczeń zadań (20%), testowego zaliczenia wykładu (30%), ulepszenia interfejsu w trakcie zaliczenia ćwiczeń (50%). Ocena końcowa zależy od liczby uzyskanych punktów:

do 50% - ocena niedostateczna,
od 50% do 60% - ocena dostateczna,
od 60% do 70% - ocena dostateczna plus,
od 70% do 80% - ocena dobra,
od 80% do 90% - ocena dobra plus,
od 90% - ocena bardzo dobra.