

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:	Język angielski II	
Nazwa w języku angielskim:	English II	
Język wykładowy:	angielski (wspomagany językiem polskim)	
Kierunek studiów, dla którego przedmiot jest oferowany:	Informatyka	
Jednostka realizująca:	Centrum Języków Obcych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):	obowiązkowy	
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):	pierwszego stopnia	
Rok studiów:	drugi	
Semestr:	trzeci	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:	Mgr J. Madej- Borychowska	
Imię i nazwisko prowadzących zajęcia:	nauczyciele języka angielskiego	
Założenia i cele przedmiotu:	Osiągnięcie językowej kompetencji komunikacyjnej na poziomie B2 ESOKJ Rady Europy.	
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Student zna słownictwo i struktury gramatyczne niezbędne do skutecznej komunikacji językowej w różnorodnych sytuacjach życia codziennego i zawodowego, zgodnie z treściami modułu kształcenia.	
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Student potrafi zrozumieć znaczenie głównych wątków przekazu zawartego w złożonych tekstach na tematy konkretne i abstrakcyjne, łącznie ze zrozumieniem dyskusji na tematy z zakresu swojej specjalności.	K_U03, K_U04
U_02	Student potrafi formułować przejrzyste wypowiedzi ustne i pisemne dotyczące tematów ogólnych i specjalistycznych.	K_U03, K_U04
U_03	Student potrafi zdobywać informacje oraz udzielać ich.	K_U03, K_U04
U_04	Student potrafi brać udział w dyskusji, argumentować, wyrażać aprobatę i sprzeciw, negocjować.	K_U03, K_U04

U_05	Student potrafi kontrolować swoje wypowiedzi pod względem poprawności gramatycznej i leksykalnej.	K_U03
U_06	Student potrafi pracować samodzielnie z tekstem specjalistycznym.	K_U03, K_U04
U_07	Student potrafi współpracować i pracować w grupie, przyjmując w niej różne role.	K_05
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Student jest gotów do krytycznej oceny posiadanej wiedzy oraz ma świadomość potrzeby znajomości języka obcego w życiu prywatnym i przyszłej pracy zawodowej.	K_K01
Forma i typy zajęć:	Studia stacjonarne: konwersatorium (60 godz.), Studia niestacjonarne: konwersatorium (32 godz).	
Wymagania wstępne i dodatkowe:		
Umiejętność posługiwania się językiem angielskim na poziomie „Język angielski I”.		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> 1. Internet: bezpieczeństwo, strony internetowe. 2. Roboty. 3. Grafika i multimedia. 4. Języki programowania. 5. Zawody w informatyce. 6. Sieci komputerowe. 7. Rzeczywistość wirtualna. 8. Bezpieczeństwo pracy. 		
Literatura podstawowa:		
<i>Infotech</i> – English for Computer Users, Santiago Remacha Esteras, wyd. Cambridge		
Literatura dodatkowa:		
<ol style="list-style-type: none"> 1. Wielki słownik angielsko-polski / polsko-angielski, red. nauk. B. Lewandowska-Tomaszczyk, 2014, PWN-OUP; 2. Oxford Advanced Learner’s Dictionary, red. J. Turnbull, 2010, OUP; 3. Słownik komputerów i internetu, C.M.H. Collins, C. Głowiński, 1999, Warszawa, wyd. Wilga; 4. Słownik skrótów informatycznych, A Faudrowicz, W. Sikorski, 1996, Warszawa, Mikom; 5. Duży słownik Informatyczny angielsko-polski, J. Szaniawski, 2003, Warszawa, ArsKom; 6. Słownik informatyki stosowanej, angielsko-polski, polsko-angielski, M. Trojański, 2007, Warszawa, wyd. C.H. Beck 2007 7. Dictionary of ICT, S.M.H. Collins, 2004, London, Bloomsberry; 8. Angielsko-polski słownik informatyczny, 2004, Warszawa, WNT. 		
Planowane formy/działania/metody dydaktyczne:		

Podejście eklektyczne, umożliwiające indywidualizację nauczania, czyli dostosowanie technik, form pracy, typów zadań i treści do danej grupy studentów. Stosowane formy pracy to, między innymi: praca w parach (np.: odgrywanie ról, wymiana informacji), praca w grupach (projekty, konkursy, rozwiązywanie problemów, zebranie słownictwa itp.), praca indywidualna studentów, czy też nauczanie tradycyjne – frontalne (prezentacja materiału leksykalnego, zasad gramatycznych, treści ilustracji itp.). Ćwiczenia wspomagane są technikami multimedialnymi.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Pisemne testy sprawdzające, ocenianie na bieżąco zadań wykonanych w domu i w trakcie zajęć (w tym wypowiedzi ustnych).

Forma i warunki zaliczenia:

Zaliczenie semestru na ocenę na podstawie:

- co najmniej dwóch testów sprawdzających stopień opanowania wiedzy i umiejętności;
- jakości wykonanych prac domowych oraz zadań na zajęciach;
- aktywności na zajęciach oraz frekwencji.

Kryteria oceniania: 0-50% – niedostateczna (2,0); 51-60% – dostateczna (3,0); 61-70% – dostateczna plus (3,5); 71-80% – dobra (4,0); 81-90% – dobra plus (4,5); 91-100% – bardzo dobra (5,0).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w konwersatorium	60 godz.
Samodzielne przygotowanie się do zajęć	30 godz.
Samodzielne przygotowanie się do kolokwium	10 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w konwersatorium	32 godz.
Samodzielne przygotowanie się do zajęć	48 godz.
Samodzielne przygotowanie się do kolokwium	20 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Syllabus of education module			
Course name / unit of study in Polish:		Podstawy systemów teleinformatycznych dla informatyków	
Name in English:		Information and Communication Technology Fundamentals for Computer Scientists	
Language of lecture:	english		
Field of study for which the subject is offered:		Computer Science	
Implementing entity:	Institute of Computer Science		
Type of course / unit of study (obligatory/ optional):		optional	
Level of education module (e.g. first or second degree):		first degree	
Year of study:	second		
Semester:	third		
ECTS credits:	2		
Name of coordinator of the course:		dr Anna Wawrzyńczak-Szaban	
Assumptions and objectives of the course:		dr Anna Wawrzyńczak-Szaban	
Symbol of the outcome	Educational outcomes		Symbol of the directional outcome
	KNOWLEDGE		
W_01	He has structured and theoretically founded knowledge in computer systems and networks, including knowledge necessary to understand the basics of both local and wide area networks. He knows the standards and operating principles of telecommunications systems: cable, radio lines, radio and satellite		K_W05 K_W09
W_02	Knows the issues of standardization and standardization in broadband transmission routes and telecommunications networks, as well as standard transmission of traditional telecommunications signals and digital information and control signals between computer systems.		K_W05 K_W09
SKILLS			
U_01	Can acquire information from literature, databases, and other sources; can integrate the obtained knowledge, make their interpretation, as well as draw conclusions and formulate, present and justify opinions.		K_U01, K_U04
SOCIAL COMPETENCE			
K_01	Can independently and in a team present their knowledge to other people.		K_K01, K_K02
Form and types of classes	Full-time studies: lectures (30 hours) Part-time studies: lectures (18 hours)		
Prerequisites:			
None			

Course content:

- 1. Telecommunications and teleinformatics.** Basic definitions and classification of telecommunications. Classification and properties of signals - continuous and discrete signal. Model of the telecommunications system. Evolution of telecommunications systems.
- 2. Transmission media.** Tracks and natural channels. Tracks and radio channels. Satellite and fiber optic.
- 3. Characteristics of channels.** Attenuation. Distortion. Noise.
- 4. Analog signals.** The concept, properties, and parameters of the analog signal. Continuous-wave modulations of amplitudes, frequencies, and phases of the signal.
- 5. Digital signals.** Discrete and digital signal - properties, parameters, examples. Pulse modulation. PCM modulation.
- 6. Data transmission.** The concept and the essence of data transmission. Distortion and disruption of data transmission.
- 7. Data transmission.** Security methods for data transmission against errors. Detection and correction codes
- 8. Computer networks topologies.** The concept of computer networks. Types of networks. Topologies of computer networks. Transmission media and components of computer networks. Media bandwidth and limitations.
- 9. The layered architecture of computer networks.** Network logical architecture and elementary operations. Model ISO / OSI. Protocols, frames, packages. TCP / IP model. Encapsulation.
- 10. Physical layer.** Transmission media and line codes. Standards and protocols of the physical layer. Connectors and interfaces
- 11. Data link layer.** Protocols. IEEE 802.x standards.
- 12. Network layer.** IP, ICMP and ARP, protocols.
- 13. The Transport Layer.** Role and services. Connectionless service. Connection-oriented service. TCP and UDP protocols.
- 14. Session layer - types of services.** The HTTP protocol. WWW and FTP services. SMTP, POP3 and IMAP4 protocols.
- 15. Application Layer.** Domain Name System. Distribution of name space. DNS on the internet.

Basic literature:

1. Alberto Leon-Garcia, Indra Widjaja, Communication Networks: Fundamental Concepts and Key Architectures, McGraw-Hill, 2004
2. Forouzan, Behrouz A, Catherine A. Coombs, and Sophia C. Fegan. Data Communications and Networking. Boston: McGraw-Hill, 2007
3. B.P. Lathi, Zhi Ding, Modern Digital and Analog Communication Systems (The Oxford Series in Electrical and Computer Engineering), Oxford University Press; 4 edition (January 23, 2009)

Additional literature:

1. Baran Z. (red.): Podstawy transmisji danych. Wyd. KiŁ, Warszawa 1992.
2. Barczak A., Florek J., Sydoruk T.: Podstawy telekomunikacji dla informatyków. Wyd. Akademii Podlaskiej, Siedlce 2011
3. Haykin S.: Systemy telekomunikacyjne. t. 1 i 2. Wyd. KiŁ, Warszawa 2000.
4. Krysiak K.: Sieci komputerowe. Kompendium. Wyd. Helion, Gliwice 2005
5. Norris M.: Teleinformatyka. Wyd. KiŁ, Warszawa 2002.

Planned forms / activities / teaching methods::

Traditional lecture supported by multimedia techniques.

Methods of verifying the learning outcomes achieved by a student:

Outcomes W_01, W_02 and the effect U_01 will be verified in a written test at the last lecture.

Sample questions:

- Provide the basic elements of the teleinformation system.
- How is the conversion of an analogue to digital signal carried out?
- What are the similarities / differences between an Ethernet network and a Token Ring?
- What are the similarities/differences between an OSI and TSP/IP model?
- Distinguish between baseband transmission and broadband transmission.
- What does the Nyquist theorem have to do with communications?

The outcomes of U_01 and K_01 will be verified by preparing a short presentation on a given topic related to modern solutions in the field of IT systems. Presentation in English.

Terms and conditions:

The module ends with getting credit with a grade. The final grade is issued on the basis of a written colloquium on final classes and a short presentation on a given subject.

For the written test you can get up to 75 points, for the presentation of 25 points. Passing the colloquium is possible after obtaining at least 38 points, while passing the presentation after obtaining 13 points.

The final mark from the module (after completing all the components), depending on the sum of the points obtained (maximum 100 points), may be the following (in brackets the ECTS grade):

- 0 – 50 pkt: ndst (F),
- 51 – 60 pkt: dst (E),
- 61 – 70 pkt: dst+ (D),
- 71 – 80 pkt: db (C),
- 81 – 90 pkt: db +(B),
- 91 – 100 pkt: bdb (A).

Corrections:

Improvement of the colloquium during the examination session.

ECTS credits:

Full-time studies

Activity	Student load
Participation in lectures	30 h
Participate in laboratory exercises	-
Preparing for presentation	7 h
Participation in consultations	3 h
Preparing for the colloquium	10 h
Total student workload	2 ECTS

Part-time studies

Activity	Student load
Participation in lectures	18 h
Participate in laboratory exercises	-
Preparing for presentation	7 h
Participation in consultations	3 h
Preparing for the colloquium	22 h
Total student workload	2 ECTS

Sylabus przedmiotu / modułu kształcenia			
Nazwa przedmiotu/modułu kształcenia:		Problemy społeczne i zawodowe informatyki	
Nazwa w języku angielskim:		Social and Vocational Issues in Computer Science	
Język wykładowy:	Polski		
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka	
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych		
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		Obowiązkowy	
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		Pierwszego stopnia	
Rok studiów:	Drugi		
Semestr:	Trzeci		
Liczba punktów ECTS:	1		
Imię i nazwisko koordynatora przedmiotu:		dr Jarosław Skaruz	
Imię i nazwisko prowadzących zajęcia:		dr Jarosław Skaruz	
Założenia i cele przedmiotu:		Celem zajęć jest zapoznanie studentów z podstawowymi problemami społecznymi i zawodowymi w informatyce	
Symbol efektu	Efekty uczenia się		Symbol efektu kierunkowego
	WIEDZA		
W_01	Zna i rozumie historię rozwoju informatyki w Polsce i na świecie		K_W03
W_03	Zna i rozumie podstawowe zagadnienia związane z ochroną danych osobowych		K_W12
W_04	Zna i rozumie podstawowe zagadnienia z zakresu ochrony własności intelektualnej oraz prawa do prywatności.		K_W04
UMIEJĘTNOŚCI			
U_01	Potrafi dokonać krytycznej analizy sposobu funkcjonowania istniejących systemów informatycznych		K_U15
U_02	Potrafi rozwiązywać praktyczne zadania inżynierskie wymagające korzystania ze standardów i norm inżynierskich oraz stosowania technologii informatycznych, wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską.		K_U24
U_03	Potrafi wykorzystać doświadczenie związane z utrzymaniem urządzeń, obiektów i systemów typowych dla kierunku informatyka zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską.		K_U25
KOMPETENCJE SPOŁECZNE			
K_01	Jest gotów do wypełniania zobowiązań społecznych, współorganizowania działalności na rzecz środowiska społecznego oraz inicjowania działań na rzecz interesu publicznego poprzez przekazywanie informacji i opinii dotyczących osiągnięć informatyki i		K_K02

	innych aspektów działalności inżyniera-informatyka; podejmuje starania, aby przekazać takie informacje i opinie w sposób zrozumiały.	
K_02	Jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
K_03	Jest gotów do myślenia i działania w sposób przedsiębiorczy.	K_K03
K_04	Jest gotów do odpowiedzialnego pełnienia roli zawodowej informatyka, w tym do przestrzegania zasad etyki zawodowej i wymagania tego od innych i do dbania o dorobek i tradycję zawodu informatyka.	K_K04
Forma i typy zajęć:	Studia stacjonarne: wykłady (15 godz.) Studia niestacjonarne: wykłady (9 godz.)	
Wymagania wstępne i dodatkowe:		
Umiejętność samodzielnego korzystania na najwyższym poziomie z dowolnego oprogramowania użytkowego, w tym z MS Office, a także umiejętność opisywania problemów społecznych i zawodowych informatyki w postaci rozszerzonych esejów typu: Nowe technologie informacyjne w zakresie handlu elektronicznego szkodzą, Nowe technologie informacyjne w zakresie handlu elektronicznego wspomagają, Nowe technologie informacyjne w zakresie handlu elektronicznego wyzwaniem, itp.		
Treści modułu kształcenia:		
<p>1. Wprowadzenie do problemów społecznych i zawodowych informatyki. Społeczeństwo informacyjne. Gospodarka elektroniczna. Cyfryzacja społeczeństwa. eCywilizacja, eEuropa, ePolska, eRząd, eUrząd, eDom, eDokument, eDekretacje, eKultura, Obszary zastosowań informatyki, Mikromodel i Makromodel Społeczeństwa informacyjnego, Ministerstwo Cyfryzacji, Pełnomocnik Rządu RP ds. informatyzacji, itp.</p> <p>2. Społeczność Internetu. Intelktualne i etyczne meandry Internetu, Przemiany wnoszone przez Internet, Sposoby inspirowania rozwoju Internetu i społeczeństwa informacyjnego, Użytkowanie Internetu jako metoda dostępu do kultury, Internet jako narzędzie autoekspresji.</p> <p>3. Problemy społeczne i zawodowe Internetu. Uzależnienie od Internetu, Ryzyko i odpowiedzialność związane z systemami informatycznymi: Internet i prawo, polityka, itp. Problemy i zagadnienia prawne dotyczące własności intelektualnej, Propaganda w Internecie i inżynieria manipulacji a formy i sposoby czerpania zasobów z sieci, System patentowy i prawne podstawy ochrony prywatności, prawo człowieka do komputera czy do ochrony przed komputerem</p> <p>5. Problemy związane z ochroną danych osobowych</p> <p>6. Problemy związane z prawem autorskim.</p> <p>7. Problematyka związana z umowami o przeniesienie praw autorskich.</p> <p>8. Umowy dotyczące współpracy B2B</p> <p>9. Umowy dotyczące wytworzenia oprogramowania</p> <p>9. Ryzyka realizacja przedsięwzięć informatycznych.</p>		
Literatura podstawowa:		
1. Castells M., Społeczeństwo sieci, Wyd. 2, WN PWN, Warszawa 2010. 2. Cieciora M.: Wybrane problemy społeczne i zawodowe informatyki, VIZJA, wyd. III, Warszawa 2012 3. Kłopotek M.: Inteligentne wyszukiwarki internetowe. EXIT. Warszawa 2001 4. Tadeusiewicz R.: Społeczność Internetu. EXIT. Warszawa 2002.		
Literatura dodatkowa:		
. Cieciora M.: Podstawy technologii informacyjnych z przykładami zastosowań. Opolgraf S.A. Warszawa 2006		

2. Dyrektywa Rady EWG z dnia 14 maja 1991 roku w sprawie ochrony prawnej programów komputerowych

3. Heller M., Elementy mechaniki kwantowej dla filozofów, Wyd. 2, Wyd. Diecezji Tarnowskiej Biblos, Tarnów 2011.

4. Szczepańska K., Metody i techniki TQM, OW PW, Warszawa 2009.

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi. Przewiduje się możliwość przygotowania artykułu i ewentualnie referatu przez studentów na zaproponowany przez studenta temat z zakresu wiedzy objętej wykładem.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty **W_01-W_04, U_01-U_03** sprawdzane będą jednokrotnie, po przygotowaniu sprawdzianu pisemnego na przedostatnim wykładzie. Na sprawdzianie pisemnym zadania będą dotyczyły wybranych problemów społecznych i zawodowych informatyki,

przykładowe zadania:

1. Wymienić rodzaje kart elektronicznych i wynikające z ich posiadania problemy społeczne informatyki. Omówić System Plików Karty Smart Card.
2. Wymienić społeczne problemy informatyki wynikające z Internetu. Omówić pojęcie rozpadliny cyfrowej.
3. Omówić zawodowe problemy wynikające z korzystania z komputera.
4. Wymienić obszary zastosowań technologii informacyjnych. Omówić obszar informatyki ludycznej.
5. Jakie są zdefiniowane w kk przestępstwa komputerowe przeciwko ochronie informacji. Omówić odpowiedzialność karną z tytułu sabotażu komputerowego.
6. Wymienić rodzaje kart magnetycznych i wynikające z ich posiadania problemy zawodowe informatyki. Omówić istotę zapisu i odczytu informacji na karcie.
7. Wymienić zawodowe problemy informatyki wynikające z Internetu. Omówić pojęcie smogu informacyjnego.
8. Wymienić społeczne problemy wynikające z korzystania z komputera. Omówić problemy wynikające z funkcjonowania ludzi nieprzygotowanych informatycznie w społeczeństwie informacyjnym.
9. Wymienić współczesne bariery rozwoju informatyki. Omówić podstawowe bariery rozwoju komputerów kwantowych.
10. Wymienić przestępstwa komputerowe wynikające z kk przeciwko wiarygodności dokumentów. Omówić odpowiedzialność karną z tytułu szpiegostwa komputerowego.

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Ocena końcowa jest wystawiana na podstawie jednego kolokwium pisemnego przeprowadzonego na ostatnim wykładzie oraz na podstawie przygotowanych przez studentów artykułów.

Na zaliczenie składa się ocena za kolokwium wg skali ECTS:

- 0 – 50 pkt: niedostateczna (F),
- 51 – 60 pkt: dostateczna (E),
- 61 – 70 pkt: dostateczna plus (D),
- 71 – 80 pkt: dobra (C),
- 81 – 90 pkt: dobra plus (B),
- 91 – 100 pkt: bardzo dobra (A).

Zaliczenie zajęć następuje w przypadku uzyskania co najmniej 51 pkt. Przewiduje się dodatkowe punkty z tytułu aktywności studenta i wyróżnienie studenta po uzyskaniu co najmniej 101 pkt. z

przedmiotu dyplomem nadanym przez prowadzącego zajęcia za trafne rozwiązywanie zagadnień w zakresie Problemów Społecznych i Zawodowych Informatyki.

Bilans punktów ECTS*:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz
Udział w konsultacjach godz. z przedmiotu	2 godz
Samodzielne przygotowanie się do kolokwium	8 godz
Sumaryczne obciążenie pracą studenta	25 godz
Punkty ECTS za przedmiot	1 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	9 godz
Udział w konsultacjach godz. z przedmiotu	2 godz
Samodzielne przygotowanie się do kolokwium	14 godz
Sumaryczne obciążenie pracą studenta	25 godz
Punkty ECTS za przedmiot	1 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Algorytmy i złożoność
Nazwa w języku angielskim:		Algorithms and Complexity
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:		Wydział Nauk Ścisłych i Przyrodniczych
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		Pierwszego stopnia
Rok studiów:	drugi	
Semestr:	Trzeci	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		Dr E. Szczepanik
Imię i nazwisko prowadzących zajęcia:		Dr Artur Niewiadomski, Dr E. Szczepanik
Założenia i cele przedmiotu:		Celem kursu jest przekazanie studentom wiedzy na temat różnych aspektów korzystania z algorytmów, w tym ich projektowania oraz złożoności obliczeniowej oraz zdobycie praktycznych umiejętności ich wykorzystania w implementowanych systemach
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia na temat algorytmów, ich własności oraz rodzajów, notacje wykorzystywane do szacowania złożoności oraz najważniejsze klasy złożoności obliczeniowej. Zna i rozumie zagadnienia związane z problemami NP - zupełnymi	K_W06
W_02	Zna i rozumie zagadnienia dotyczące reprezentacji grafów, algorytmów przeszukiwania grafów, operacji słownikowych na drzewach i ich złożoności obliczeniowej	K_W06
W_03	Zna i rozumie zagadnienia związane z algorytmami sortowania, wyszukiwania wzorca w tekście i ich złożoności obliczeniowej	K_W06
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego

U_01	Potrafi implementować i analizować klasyczne algorytmy związane z przetwarzaniem list, grafów i drzew, a także znanych algorytmów sortowania i algorytmów tekstowych	K_U22
U_02	Potrafi implementować algorytmy uwzględniające programowanie strukturalne i obiektowe, zarówno podczas pracy na zajęciach jak i w ramach przygotowywania się do zajęć	K_U22
U_03	Potrafi dobierać i wykorzystywać klasyczne algorytmy adekwatnie do postawionego zadania	K_U22
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
Forma i typy zajęć:	Studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (30 godz.), Studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.).	
Wymagania wstępne i dodatkowe:		
<p>Warunkiem uczestnictwa w zajęciach jest wcześniejsze uzyskanie zaliczenia z następujących przedmiotów:</p> <ul style="list-style-type: none"> • Matematyka dyskretna • Podstawy programowania <p>lub znajomość literatury obowiązującej w tym przedmiocie. Student musi mieć opanowane podstawy kombinatoryki.</p> <p>Ponadto wymagana jest znajomość podstaw strukturalnego lub obiektowego języka programowania, a w tym umiejętność definiowania typów, posługiwanie się instrukcjami iteracyjnymi, podprogramami i rekurencją.</p>		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> 1. Algorytmy komputerowe: Pojęcie i własności algorytmu, Metody konstruowania algorytmów, Schematy blokowe, Algorytm NWD, Typy danych 2. Złożoność obliczeniowa algorytmów, Klasy złożoności czasowej, rzędy wielkości, porównanie i ocena złożoności algorytmów, złożoność pamięciowa 3. Podstawowe techniki i struktury, podstawowe typy algorytmów, listy, operacje na listach, stosy, kolejki 4. Rodzaje i reprezentacje grafów: grafy nieskierowane i skierowane, grafy ważone, reprezentacje macierzowe, reprezentacja listowa: listy incydencji 5. Algorytmy grafowe: przeszukiwanie grafu w głąb, przeszukiwanie grafu wszerz, znajdowanie najkrótszej drogi między wierzchołkami w grafach ważonych, obliczanie silnie spójnych składowych 6. Drzewa: drzewa kopcowe i spadowe, drzewa BST, AVL, RST, operacje na drzewach 7. Drzewa: drzewa TRIE, PATRICIA, 2-3 drzewa, drzewa turniejowe, operacje na drzewach 8. Problemy NP-trudne i trudniejsze: lista problemów NP-trudnych, dowodzenie NP-trudności, problemy nierozstrzygalne, 9. Algorytmy obliczeniowe: algorytmy kombinatoryczne (obliczanie permutacji, silni, wariacji, kombinacji), algorytmy operacji na macierzach, algorytmy przekształcania liczb, algorytm znajdowania najmniejszego lub największego elementu 10. Proste algorytmy sortowania tablic: sortowanie metodą wstawiania, wybierania, zamiany, sortowanie bąbelkowe, sortowanie metodą Shella 		

11. **Skomplikowane algorytmy sortowania tablic:** sortowanie kopcowe, sortowanie przez podział (sortowanie szybkie), sortowanie rzędowe (pozycyjne)
12. **Algorytmy sortowania plików:** sortowanie na trzech plikach, sortowanie przez łączenie proste i naturalne, sortowanie wielofazowe,
13. **Algorytmy wyszukiwania wzorca w tekście:** dopasowywanie wzorca i wyszukiwanie 'naiwne', drzewo sufiksowe i graf podslów
14. **Algorytmy ze strukturą drzewiastą (z nawrotami) i rekurencyjne,** algorytm ustawiania 8 hetmanów, algorytm znalezienia drogi skoczka szachowego, wieże Hanoi
15. **Heurystyki dla problemów NP-zupełnych i podsumowanie wykładu,** heurystyki dla problemów NP-zupełnych, przekształcenia między problemami NP-zupełnymi, problemy o złożoności wykładniczej i nierozstrzygalne, omówienie zakresu i zasad egzaminu

Literatura podstawowa:

1. Banachowski L., Diks K., Rytter W., Algorytmy i struktury danych, WNT, Warszawa, 2006,
2. Aho A.V., Hopcroft J.E., Ullman J.D., Projektowanie i analiza algorytmów komputerowych, PWN, Warszawa 2003,
3. Niklaus Wirth, Algorytmy + struktury danych = programy, WNT Warszawa, 1999
4. Timofiejew A. Algorytmy i struktury danych w językach programowania. Siedlce, Wydawnictwo Akademii Podlaskiej, 2006.

Literatura dodatkowa:

1. A. V. Aho, J. E. Hopcroft, J.D. Ullman, Projektowanie i analiza algorytmów komputerowych, Helion 2003
2. Ch. H. Papadimitriou: Złożoność obliczeniowa, Helion 2003.
3. A. Niewiadomski, P. Świtalski, T. Sidoruk, W. Penczek, Applying Modern SAT-solvers to Solving Hard Problems. Fundam. Inform. 165(3-4): 321–344 (2019), <https://doi.org/10.3233/FI-2019-1788>

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi. Zajęcia laboratoryjne – zajęcia praktyczne z wykorzystaniem języka Java lub C# i wybranego środowiska programistycznego (np. IntelliJ, Visual Studio).

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W_01 – W_03 będą sprawdzane na egzaminie pisemnym. Na egzaminie pisemnym zadania będą dotyczyły wybranych algorytmów i złożoności obliczeniowej, przykładowe zadania:

- Dla danego grafu podaj kolejność odwiedzanych wierzchołków przez algorytm DFS i BFS
- Dla danego drzewa AVL, podaj algorytm usunięcia wybranego elementu i przywrócenia drzewa do postaci AVL.
- Podaj kilka przykładów problemów NP – zupełnych.

Efekty U_01, U_02 i U_03 sprawdzane będą sukcesywnie i oceniane po każdym laboratorium, również poprzez sprawdzanie prac domowych i kolokwium. Przykładowe zadania:

- Dana jest abstrakcyjna klasa AGraph, utworzyć własną klasę dziedziczącą z podanej klasy AGraph. Zaimplementować metody abstrakcyjne check, connect, writeMatrix oraz writeList.
- Zaimplementować metodę, która wypisze wierzchołki grafu nie mające żadnych sąsiadów.

Effekt K_01 będzie weryfikowany, w oparciu o posiadaną wiedzę i umiejętności w czasie zajęć laboratoryjnych, a także będzie sprawdzany na egzaminie.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim. Na tej formie zajęć student może maksymalnie uzyskać 50 pkt. Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania powyżej 25 punktów.

Poprawa laboratorium: najpóźniej w ciągu 3 tygodni od danych zajęć. Poprawa nie jest możliwa po zakończeniu semestru.

Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 50 pkt. Egzamin będzie zaliczony w przypadku uzyskania powyżej 25 pkt.

Ocena końcowa z przedmiotu, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 50 pkt: niedostateczna (F),
- 51 – 60 pkt: dostateczna (E),
- 61 – 70 pkt: dostateczna plus (D),
- 71 – 80 pkt: dobra (C),
- 81 – 90 pkt: dobra plus (B),
- 91 – 100 pkt: bardzo dobra (A).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	30 godzin
Udział w ćwiczeniach laboratoryjnych	30 godzin
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	18 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny
Studia indywidualne z tematów realizowanych na wykładach	10 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	10 godzin
Sumaryczne obciążenie pracą studenta	100 godziny
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godzin
Udział w ćwiczeniach laboratoryjnych	15 godzin
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	28 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny

Studia indywidualne z tematów realizowanych na wykładach	20 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	20 godzin
Sumaryczne obciążenie pracą studenta	100 godziny
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Technologie sieciowe
Nazwa w języku angielskim:		Computer Networks
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:		Wydział Nauk Ścisłych i Przyrodniczych
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	drugi	
Semestr:	trzeci	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr hab. Stanisław Ambroszkiewicz Prof. uczelni
Imię i nazwisko prowadzących zajęcia:		dr hab. Stanisław Ambroszkiewicz Prof. uczelni dr Andrzej Salamończyk dr Grzegorz Terlikowski
Założenia i cele przedmiotu:		Celem modułu jest zapoznanie ze klasycznymi (Internet) oraz współczesnymi protokołami i technologiami używanymi w sieciach komputerowych z uwzględnieniem SDN i NFV, oraz samodzielne zaprojektowanie i realizacja prostej aplikacji w architekturze chmury (CNApp): użytkownik CLI, API Gateway, mikroserwisy (FaaS), oraz BaaS (DB i HD).
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Student zna i rozumie podstawowe pojęcia z dziedziny sieci komputerowych.	K_W07
W_02	Zna i rozumie zasady działania warstw sieci komputerowych w Internetowym modelu warstwowym	K_W07
W_03	Zna i rozumie protokoły komunikacyjne wykorzystywane w sieciach komputerowych,	K_W07
W_04	Zna i rozumie adresowanie, protokoły i standardy wykorzystywane powszechnie w Internecie	K_W07, K_W09
W_05	Zna i rozumie podstawy programowania sieciowego w oparciu o język Java,	K_W07
W_06	Zna i rozumie podstawy projektowania i zarządzania sieciami LAN.	K_W07
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego

U_01	Potrafi implementować aplikacje sieciowe, także działające w oparciu o bazę danych, Potrafi zaprojektować i zrealizować proste aplikacje sieciowe w architekturze cloud-native na mikroserwisach	K_U22
U_02	Potrafi zaprojektować, zrealizować i skonfigurować prostą sieć (routery, serwery, hosty), także z wykorzystaniem DHCP	K_U21
U_03	Potrafi diagnozować i usuwać usterki w sieciach komputerowych	K_U11, K_U21
U_04	Potrafi posługiwać się symulatorem sieci.	K_U17, K_U07
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
Forma i typy zajęć:	studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (30 godz.), studia niestacjonarne: wykłady (18 godz.), ćwiczenia laboratoryjne (21 godz.).	
Wymagania wstępne i dodatkowe:		
<ol style="list-style-type: none"> Umiejętność programowania w języku obiektowym (Java). Znajomość architektury systemu komputerowego. 		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> Podstawowe pojęcia i definicje związane z technologiami sieciowymi. Pojęcie protokołu, warstwy protokołów w sieciach komputerowych. Model warsrwowy OSI i Internetowy TCP-UDP/IP - porównanie. Rys historyczny Internetu, rola aplikacji takich jak email a zwłaszcza WWW. Warstwa aplikacji – wprowadzenie. Model programowanie sieciowego klient-serve, gniazda (sockety) TCP, oraz UDP. P2P – Gnutella oraz Torrnet. Aplikacje sieciowe w chmurze obliczeniowej oraz architektura serverless. Przegląd podstawowych aplikacji i protokołów sieciowych: ftp, email, WWW, oraz DNS. Warstwa aplikacji - adresacja i nazewnictwo. Adresacja IP i omówienie DNS. Warstwa aplikacji - zastosowania internetowe. Omówienie protokołu HTTP, SMTP i POP3, telnet, FTP, adresacji URL i HTML. Video Streaming oraz Content Distribution Networks Warstwa transportu – wprowadzenie. Zasady pewnego przesyłania danych w sieciach komputerowych. Algorytmy Go-Back-N i Selective Repeat. Warstwa transportu. Protokół TCP oraz struktura segmentu TCP, protokół UDP. Warstwa sieci – wprowadzenie. Zasady routingu, algorytm routingu Link State. Warstwa sieci. Algorytm routingu Distance Vector. Routing hierarchiczny i BGP Data Plane oraz Control Planes. Generalized Forwarding oraz SDN. Warstwa łączy danych - dostęp do medium. Kodowanie sygnałów w sieci. Rodzaje protokołów dostępu do medium (MAC), adresowanie fizyczne MAC. Techniki wykrywania błędów. Warstwa łączy danych – technologie. Prekursor Ethernetu - ALOHA net. Protokół CSMA/CD. Ethernet oraz protokół ARP. Token Ring oraz FDDI.Wi-Fi. Warstwa łączy danych - urządzenia oraz elementy projektowania sieci lokalnych. Karty sieciowe, modemy, koncentratory, mostki, przełączniki Video streaming oraz CDN content distribution networks <ul style="list-style-type: none"> Multimedia: video Content distribution networks (CDNs) Przykład: Netflix Ewolucja funkcjonalności warstwy transportu <ul style="list-style-type: none"> QUIC: Quick UDP Internet Connections Generalized Forwarding, SDN 		

- dopasuj+wykonaj
 - OpenFlow
 - Middleboxes
18. **Software defined networking (SDN) control plane**
- Komponenty Kontrolera SDN
 - Protokół OpenFlow
19. **Bezpieczeństwo sieci komputerowych.** Zagrożenia bezpieczeństwa, techniki włamań, metody zapewniania bezpieczeństwa. Zarys technologii DES, RSA, podpis cyfrowy oraz PGP.

Treści zajęć laboratoryjnych

Część I: realizacja prostej aplikacji sieciowej opartej na mikroserwisach w architekturze chmury obliczeniowej

- 1 **Elementy implementacji aplikacji sieciowej typu klient/serwer:** Prosty klient TCP i serwer TCP
- 2 **Realizacja prostego czatu na gniazdach TCP w języku Java.**
- 3 **Implementacja wielowątkowego serwera TCP,** separacja logiki wątku od logiki wykonania.
- 4 **Asynchroniczna transmisja danych w Javie.**
- 5 **Zaawansowane mechanizmy transmisji danych:** klasy strumieniowe, dzielenie i łączenie strumieni, buforowanie i formatowanie przesyłanych danych.
- 6 **Przesyłanie zawartości plików:** bajtowo i za pomocą stringów (kodowanie w Base64)
- 7 **Programowanie aplikacji sieciowych wykorzystujących bazy danych.** Interfejs JDBC do połączeń z bazami danych. Przykład prostej aplikacji sieciowej wykorzystującej bazę danych.
- 8 **Architektura CNAApp:** projekt prostej aplikacji
- 9 **Implementacja tej aplikacji CNAApp**

Część II: Administrowanie i projektowanie LAN-ów

- 10 **Identyfikacja podstawowych urządzeń sieciowych i weryfikacja ustawień sieciowych.**
- 11 **Wykonanie i testowanie kabla prostego, krosowego, odwrotnego.**
- 12 **Rozwiązywanie problemów związanych z warstwą fizyczną.** Budowa sieci LAN złożonej z kilku komputerów, testowanie i monitorowanie sieci.
- 13 **Podstawy konfigurowania routerów Cisco.** Weryfikacja i modyfikacja plików konfiguracyjnych routera. Instalacja, konfiguracja i umiejętność wykorzystania serwera TFTP do wysyłania i odbierania plików.
- 14 **Adresowanie IP - adresacja routerów w zadanej topologii.** Znaczenie klas adresów IP, podsieci, masek podsieci, konfiguracja adresu IP na interfejsach routera.
- 15 **Konfiguracja i weryfikacja działania protokołu RIP i OSPF w sieciach LAN.**
- 16 **Topologie sieci lokalnych oraz zasady projektowania.**
- 17 **Zespołowa realizacja zadanej topologii sieci.** Adresowanie sieci dysponując "nie zaadresowaną" topologią. Symulacja zaprojektowanej sieci.
- 18 **Elementy administrowania i zarządzania sieciami.** Serwery proxy oraz ściany ogniowe. Protokół SNMP.
- 19 **Praktyczna realizacja prostej sieci (routery, serwery, hosty) z wykorzystaniem DHCP.** Analiza ruchu w czasie rzeczywistym.
- 20 **Elementy projektowania sieci komputerowych.** Analiza wymagań użytkownika, wybrane zagadnienia projektowania sieci w warstwach 1 2 i 3 modelu OSI.
- 21 **Diagnozowanie i rozwiązywanie problemów w sieciach LAN.**
- 22 **Obrona zadań indywidualnych.**

Literatura podstawowa:

1. James Kurose, Keith Ross - Computer Networking_ A Top-Down Approach, 7th Edition (2017) and 8th Edition (2022)
2. Kurose J. F., Ross K. W., Sieci komputerowe. Od ogółu do szczegółu z Internetem w tle, Wydawnictwo Helion, Gliwice 2006
3. K. Krysiak. Sieci Komputerowe - Kompendium. Wydawnictwo Helion 2005
4. T. Sheldon. Wielka Encyklopedia Sieci Komputerowych. Wydawnictwo Robomatic s.c. 1999.
5. Tom Laszewski, Kamal Arora, Erik Farr, Piyum Zonooz.
6. Cloud Native Architectures: Design high-availability and cost-effective applications for the cloud. 1st ed. Packt Publishing, 2018. Web. 14 Oct. 2022.
7. Jim Doherty. SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization. Copyright © 2016 Pearson Education, Inc. ISBN-13: 978-0-13-430640-7, ISBN-10: 0-13-430640-6

Literatura dodatkowa:

1. Akademia Sieci Cisco. CCNA Exploration, Semestr 1-4. PWN, Warszawa 2011
2. Leinwand, B. Pinsky. Konfiguracja Routerów Cisco. Podstawy. Mikom, Warszawa 2002.
3. M. Sportack. Routing IP - podstawowy podręcznik. Mikom, Warszawa 2000.
4. M. Sportack. Sieci komputerowe. Wydawnictwo Helion 2004,
5. R. Wright. Elementarz routingu IP. Mikom, Warszawa 1999.

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, Laboratoria z wykorzystaniem sprzętu sieciowego. Zamieszczanie na stronach internetowych zadań i materiałów do ćwiczeń

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekt U_01 jest sprawdzany przy obronie programistycznego zadania indywidualnego.

Efekty U_02 – U_04, są sprawdzane w czasie ocenianych zadań na laboratoriach. Niektóre z tych zadań są realizowane w grupach, w ten sposób są sprawdzana jest umiejętność K_03.

Efekty W_01 – W_06, K_01 sprawdzane są egzaminie.

Przykładowe pytania efektu W_01:

- *Omów topologie fizyczne i logiczne w sieciach komputerowych.*
- *Porównaj architektury klient-serwer i peer-to-peer.*

Przykładowe pytania efektu W_02:

- *Wymień warstwy modelu OSI i omów rolę tych warstw.*
- *Podaj przykłady protokołów działających w poszczególnych warstwach. Uzasadnij dlaczego ten protokół działa w tej warstwie.*

Przykładowe pytania efektu W_03:

- *Technologia Ethernet. Format ramki Ethernet.*
- *Protokół IP, format pakietu IP.*
- *Protokół TCP, nawiązywanie połączenia w TCP. Porty dobrze znanych usług.*
- *Protokół UPD, różnice pomiędzy TCP i UDP.*

Przykładowe pytania efektu W_04:

- *Omów adresację IPv4, Podziel sieć na podsieci, określ typ adresu (hosta, rozgłoszeniowy, sieci).*
- *Omów protokół http/smpt/pop3/ftp.*

Przykładowe pytania efektu W_05 i W_06:

Zaprojektuj adresację małej sieci. Dobierz do niej odpowiednią topologię fizyczną i logiczną.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium.

Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim oraz z samodzielnie wykonanego zadania indywidualnego według schematu:

- Regularne zajęcia – 26 pkt.,
- Obrona zadania indywidualnego – 14 pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 13 pkt., obrona indywidualnego zadania – co najmniej 7 pkt.

Na tej formie zajęć student może maksymalnie uzyskać 40 pkt.

Egzamin jest egzaminem ustnym. Można na nim uzyskać do 60 pkt. Egzamin będzie zaliczony w przypadku uzyskania, co najmniej 30 pkt. Ocena końcowa z przedmiotu, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 50 pkt: niedostateczna (F),
- 51 – 60 pkt: dostateczna (E),
- 61 – 70 pkt: dostateczna plus (D),
- 71 – 80 pkt: dobra (C),
- 81 – 90 pkt: dobra plus (B),
- 91 – 100 pkt: bardzo dobra (A).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	30 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	22 godz.
Udział w konsultacjach godz. z przedmiotu	8 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	10 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	18 godz.
Udział w ćwiczeniach laboratoryjnych	21 godz.

Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	36 godz.
Udział w konsultacjach godz. z przedmiotu	4 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	21 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:	Bazy danych	
Nazwa w języku angielskim:	Database	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:	Informatyka	
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):	obowiązkowy	
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):	pierwszego stopnia	
Rok studiów:	drugi	
Semestr:	trzeci	
Liczba punktów ECTS:	5	
Imię i nazwisko koordynatora przedmiotu:	prof. dr hab. inż. Andrzej Barczak	
Imię i nazwisko prowadzących zajęcia:	prof. dr hab. inż. Andrzej Barczak Mgr W. Nabiałek, Mgr Z. Młynarski	
Założenia i cele przedmiotu:	Celem zajęć jest zapoznanie studentów z podstawami teoretycznymi baz danych oraz z zasadami projektowania prostych baz danych w środowisku MY SQL.	
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie podstawowe zagadnienia z zakresu baz danych, wykorzystywanych modeli, projektowania relacyjnych baz danych i języków zapytań	K_W08
W_02	Zna i rozumie podstawowe modele oraz podstawowe zasady projektowania baz danych oraz język zapytań SQL	K_W08
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	potrafi pozyskiwać informacje na temat relacyjnych baz danych z literatury oraz innych źródeł, w tym zwłaszcza internetowych; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, a także wyciągać wnioski oraz formułować i uzasadniać opinie	K_U01

Sylabus przedmiotu / modułu kształcenia		
U_02	potrafi zaprojektować, zaimplementować oraz przetestować prosty bazodanowy system informatyczny	K_U10, K_U16, K_U19
U_03	potrafi rozwiązywać praktyczne zadania inżynierskie w dziedzinie relacyjnych baz danych wymagające korzystania ze standardów i norm inżynierskich oraz stosowania technologii informatycznych, wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską	K_U24
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
Forma i typy zajęć:	studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (30 godz.) studia niestacjonarne: wykłady (18 godz.), ćwiczenia laboratoryjne (21 godz.)	
Wymagania wstępne i dodatkowe:		
<p>Warunkiem uczestnictwa w zajęciach jest wcześniejsze uzyskanie zaliczenia z następujących przedmiotów:</p> <ul style="list-style-type: none"> • Podstawy programowania, • Analiza matematyczna, • Matematyka dyskretna, • Algebra liniowa, • Architektura systemów komputerowych, • Programowanie deklaratywne, • Algorytmy i złożoność. <p>lub znajomość literatury obowiązującej w tych przedmiotach.</p>		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> 1. Podstawowe pojęcia baz danych. Definicja i rodzaje baz danych. Obiekty i związki, system bazy danych. Modele danych, model danych jako architektura; Funkcje zarządzania bazą danych, system zarządzania bazą danych. Projektowanie baz danych, języki baz danych. Właściwości baz danych, korzyści stosowania baz danych. <i>Zapoznanie ze środowiskiem MySQL.*</i> 2. Relacyjny model danych. Definicja relacji, atrybuty/dziedzina i schemat relacji. Właściwości relacji, przykłady relacji. Klucze relacji, typy związków (relacji). Typy i stopień uczestnictwa, diagramy związków. Pułapki połączeń. Przekształcanie diagramu E - R w schemat relacji. <i>Zapytania wybierające cz. I.</i> 3. Relacyjny model danych, cd. Integracja schematu relacji. Integracja danych. Reguły integralności wewnętrznej. Więzy propagacji. integralność dodatkowa. postulaty Codda. <i>Zapytania wybierające cz. II.</i> 4. Hierarchiczny i sieciowy model danych. Struktura danych. Relacje/powiązania. Operowanie danymi. 		

Sylabus przedmiotu / modułu kształcenia

Integralność danych. Wady i zalety. Porównanie klasycznych modeli danych. *Zapytania wybierające cz. III.*

5. Obiektowy model danych(OMD). Pojęcie obiektowości, składniki OMD. Mechanizmy uogólniania i agregacji. Procesy dziedziczenia. Integralność wewnętrzna. Notacje diagramów E - R dla OMD. ujednoczony język modelowania UML. *Struktury danych w MySQL.*

6. Rozproszone bazy danych. Zadania i zalety rozproszenia. Systemy zarządzania rozproszoną bazą danych.

Systemy klient-serwer. Jednorodna i niejednorodna rozproszona baza danych. Federacyjny system baz danych.

Zapoznanie ze środowiskiem XAMP

7. Normalizacja bazy danych. Pojęcie normalizacji. Typy zależności 1 NF, wady 1 NF. Pełna zależność funkcyjna

- 2 NF, wady 2 NF. Przechodnie zależności funkcyjne - 3 NF, wady 3NF. Akomodacja zależności funkcyjnych i niefunkcyjnych. Diagramy zależności, 4 NF i 5NF. *Projekt bazy danych*

8. Interfejs SZBD - język SQL. Podstawowe pojęcia SQL. Klauzula SELECT. Wyrażenia z kilkoma operatorami. Wyrażenia w klauzuli WHERE. Wyrażenia w klauzuli ORDER BY. *Projekt bazy danych*

9. SQL - funkcje sumaryczne. Konstrukcje GROUP BY i HAVING. Tabele sumaryczne. Procent całości. Użycie indeksu. Metody złączenia zewnętrznego. *Projektowanie formularzy cz. I*

10. SQL – podzapytania. Zapytania złożone. Podzapytania. ANY i ALL. Podzapytania skorelowane. EXISTS i NOT EXISTS. Projektowanie formularzy cz. II 11. SQL – złączenia. Przegląd złączeń. Zastępowanie podzapytań złączeniami. Złączenia z tabelami sterującymi. Złączenia z podzapytaniem. UNION. Perspektywy ze złączeniami. Projektowanie raportów i zestawień

11. SQL – perspektywy. Perspektywy ze złożonymi zapytaniem. Modyfikowanie danych za pomocą perspektyw.

Stosowanie perspektyw w celu zwiększenia bezpieczeństwa danych. Usuwanie perspektyw. Stosowanie tabel słownikowych do badania perspektyw. Autoryzacja dostępu do danych

12. SQL - wstęp do sterowania transakcjami. Transakcje. COMMIT, AUTOCOMMIT, ROLLBACK. Współdziałanie

COMMIT, AUTOCOMMIT, ROLLBACK. CREATE TABLE ... AS SELECT, DESCRIBE. Tabele słownikowe. COMMENT ON, ALTER TABLE. *Aplikacja w środowisku sieciowym*

13. Fizyczne projektowanie bazy danych. Proces projektowania. Definiowanie parametrów systemu.

Definiowanie procesów działania. Model pojęciowy danych. Schemat bazy danych. Dokumentowanie projektu. *Praca z danymi zewnętrznymi*

14. Projektowanie aplikacji bazy danych. Strategia (analiza wstępna problemu). Analiza szczegółowa problemu. Projektowanie systemu. Implementacja systemu. Wdrażanie systemu. *Prezentacja projektu. Zaliczenie ćwiczeń laboratoryjnych przedmiotu.*

15.* *Kursywą zaznaczono zagadnienia realizowane w ramach ćwiczeń laboratoryjnych*

Literatura podstawowa:

1. Banachowski L., Mrówka-Matejewska E., Stencel K.; Systemy baz danych. Wykłady i ćwiczenia; Wydawnictwo PJWSTK, 2006
2. Barczak A., Florek J., Sydoruk T.: Bazy danych; Wyd. Akademii Podlaskiej, Siedlce 2006
3. Colby J., Wilton P.; SQL od podstaw; Wydawnictwo Helion, 2005
4. Date C.J.; Relacyjne bazy danych dla praktyków; Wydawnictwo Helion, 2005
5. Marklyn B., Whitehorn M.; Relacyjne bazy danych; Wydawnictwo Helion, 2003

Sylabus przedmiotu / modułu kształcenia

Literatura dodatkowa:

1. Beynon-Davies P.; Systemy baz danych; Wydawnictwa Naukowo Techniczne, 2003
2. Banachowski L., Bazy danych : wykłady i ćwiczenia / Lech Banachowski [et al.]. Warszawa : Wydaw. PJWSTK, 2003
3. Jan L. Harrington; SQL dla każdego; Wydawnictwo Mikom, 2005
4. Hernandez M. J., Viescas J. L.; Zapytania SQL dla zwykłych śmiertelników; Wydawnictwo MIKOM; 2001 r.
5. Hernandez M. J.; Bazy danych dla zwykłych śmiertelników; Wydawnictwo MIKOM, 2004 r.
6. Kowalski P.; Podstawowe zagadnienia baz danych i procesów przetwarzania; Wydawnictwo MIKOM, 2005
7. Riordan R.M.; Projektowanie systemów relacyjnych baz danych. Wyd. RM, Warszawa 2000

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratoria – praktyczna praca na komputerze. Zamieszczanie na stronach internetowych zagadnień teoretycznych i zadań ćwiczeniowych.

Sposoby weryfikacji efektów uczenia się osiąganym przez studenta:

Efekty W_01 i W_02 będą weryfikowane na egzaminie pisemnym. Przykładowe pytania:

- Wyjaśnij podstawowe pojęcia: baza danych, rodzaje baz danych, obiekty baz danych, funkcje baz danych, relacyjny model danych,
- Wymień i scharakteryzuj właściwości BD,
- Scharakteryzuj systemy zarządzania bazą danych,
- Określ wynik działania operatorów relacji na podanych tabelach, □ Opracuj model logiczny bazy danych,
- Napisz w języku SQL dla zdefiniowanej bazy danych zestaw zapytań, □ Przeprowadź proces normalizacji bazy danych.

Efekty U_01 - U_03 weryfikowane będą w trakcie zajęć oraz sprawdzane na egzaminie pisemnym.

Przykładowe zadania:

- Zaproponuj strukturę podanej bazy danych właściwą dla 1NF, 2NF, 3NF postaci normalnej.
- Podaj polecenie SQL umożliwiające np. : o pobranie z bazy danych rekordów studentów o stypendium > od średniej kwoty stypendium liczonej dla studentów danego roku i kierunku
 - o aktualizację wierszy tabeli spełniających podany warunek (np. aktualizacja płacy pracowników polegająca na powiększeniu poborów wybranych pracowników, np. o stażu większym od 5, o 2%,
- Uwzględniając strukturę bazy danych i podane zależności funkcyjne określ w jakiej postaci normalnej jest podana baza danych. .

Efekt K_01 będzie weryfikowane, w oparciu o posiadaną wiedzę i umiejętności, w czasie zajęć laboratoryjnych.

Przykładowe zadania:

- Zaprojektuj strukturę rozmów z potencjalnym użytkownikiem bazy danych w celu określenia diagramu związków encji.

Przed egzaminem studenci będą mieli dostęp do pełnej listy pytań.

Zadania na następne laboratorium muszą być dostępne co najmniej tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich przygotować samodzielnie lub korzystając z konsultacji.

Sylabus przedmiotu / modułu kształcenia

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny częściowe uzyskane na regularnych zajęciach z nauczycielem akademickim oraz z samodzielnie wykonanego zadania indywidualnego według schematu:

- regularne zajęcia – 30 pkt.
- zadanie indywidualne – 20pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów za regularne zajęcia (15 pkt) i co najmniej połowę punktów za zadanie indywidualne (10pkt.). Na tej formie zajęć student może maksymalnie uzyskać 50 pkt.

Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 50 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 25 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) jest następująca (w nawiasach ocena wg skali ECTS): Kryteria oceniania:

- 0 – 50 - niedostateczna (2,0) (F)
- 51 -60 - dostateczna (3,0) (E)
- 61 -70 - dostateczna plus (3,5) (D)
- 71 -80 - dobra (4,0) (C)
- 81 -90 - dobra plus (4,5) (B)
- 91 -100 - bardzo dobra (5,0).(A)

Poprawy: Jeden termin poprawkowy egzaminu pisemnego. Dodatkowy termin zaliczenia laboratorium w sesji egzaminacyjnej.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	30 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	40 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	20 godz.
Sumaryczne obciążenie pracą studenta	125 godz.
Punkty ECTS za przedmiot	5 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	18 godz.

Sylabus przedmiotu / modułu kształcenia	
Udział w ćwiczeniach laboratoryjnych	21 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	50 godz.
Udział w konsultacjach godz. z przedmiotu	6 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	30 godz.
Sumaryczne obciążenie pracą studenta	125 godz.
Punkty ECTS za przedmiot	5 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Programowanie niskopoziomowe
Nazwa w języku angielskim:	Low level programming	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	drugi	
Semestr:	Trzeci	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Andrzej Salamończyk
Imię i nazwisko prowadzących zajęcia:		dr Andrzej Salamończyk mgr Wojciech Nabałek mgr inż. M. Przychodzki
Założenia i cele przedmiotu:		Celem modułu jest zapoznanie ze specyfiką programowania niskopoziomowego na przykładzie asemblera MASM dla procesora x86
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Student zna i rozumie architekturę procesorów z rodziny x86 i sposoby programowania niskopoziomowego na tych procesorach	K_W06
W_02	Zna i rozumie rolę i zastosowanie asemblerów w systemach informatycznych	K_W06
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi posługiwać się językiem asemblera MASM używając: instrukcji warunkowych, pętli, operacji na liczbach całkowitych, tablic i łańcuchów znaków	K_U10, K_U11
U_02	Potrafi tworzyć w języku asemblera aplikacje konsolowe i okienkowe	K_U11

Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
Forma i typy zajęć:		studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:		
<ol style="list-style-type: none"> 1. Wiedza na temat architektury komputerów 2. Wiedza na temat podstaw programowania oraz umiejętność programowania w jednym z języków wysokiego poziomu 		
Treści modułu kształcenia:		
<p>Wykład</p> <ol style="list-style-type: none"> 1. Podstawy asemblera. Rola i znaczenie asemblerów. Narzędzia programowania. Tworzenie programu w języku asemblera. Reguły zapisu programu w języku asemblera. 2. Systemy komputerowe (komputery) na bazie procesorów firmy Intel. Rejestry procesorów Intel. Rozkazy procesorów. Adresowanie operandów. Znaczniki wyników operacji. 3. Assembler Microsoft Macro Assembler (MASM). Elementy języka asemblera MASM. Instrukcje. Dyrektywy. Wskaźnik pozycji. Przesyłanie danych. 4. Operacje arytmetyczne i logiczne. Operacje bitowe. 5. Sterowanie wykonaniem programu w języku Macro Assembler (MASM). Porównania. Skoki. Pętle. 6. Procedury. Makroinstrukcje. Tworzenie własnych procedur i makr. 7. Programowanie z zastosowaniem funkcji API Win32. Programowanie aplikacji konsolowej. Operacje na wierszach. Operacje na plikach. 8. Stosowanie jednostki zmiennoprzecinkowej i jednostki MMX. Alokacja i przesyłanie danych. Operacje arytmetyczne. Operacje trygonometryczne. Operacje porównania. 9. Programowanie aplikacji graficznych. Współdziałanie aplikacji graficznej z systemem Windows. Tworzenie okna. Standardowe obiekty graficzne. Kontekst urządzenia. Korzystanie z zasobów. 10. Programowanie wstawek asemblerowych w języku C (C++). Wywoływanie w języku C (C++) funkcji napisanych w języku asemblera. Wywoływanie w asemblerze funkcji napisanych w języku C (C++). 11. Kolokwium <p>Laboratorium</p> <ol style="list-style-type: none"> 1. Tworzenie i uruchamianie programów asemblerowych. Przejście do trybu konsolowego. Kompilacja, opcje kompilatora. Konsolidacja, opcje konsolidatora. Opracowanie aplikacji konsolowej 2. Zarządzanie danymi. Przesyłanie danych. Praca z łańcuchami. Operacje na stosie. Tryby adresowania. 3. Operacje arytmetyczne i logiczne. Przesuwanie i rotacja bitów. 4. Sterowanie przebiegiem wykonania programu. Porównania i skoki warunkowe. Pętle. 5. Podprogramy i makrodefinicje. 		

6. Operacje na plikach i katalogach. Tworzenie plików i katalogów. Otwieranie i zapisywanie.
7. Obsługa sprzętu. Klawiatura i mysz. Tworzenie okna konsolowego.
8. Koprocesor i jednostka MMX.
9. Tryb graficzny. Tworzenie okna. Obiekty graficzne.
10. Korzystanie z plików zasobów.
11. Wstawki assemblerowe.
12. Prezentacja i zaliczenie zadania indywidualnego.

Literatura podstawowa:

1. Timofiejew. Praktyczny kurs programowania w językach assemblerów. Wydawnictwo UPH 2012
2. Vlad Pirogow. Asembler. Podręcznik programisty. Helion 2005.

Literatura dodatkowa:

1. Błaszczuk. Win32ASM. Asembler w Windows. Helion 2004
2. K. R. Irvine. Asembler dla procesorów Intel – Vademecum profesjonalisty. Helion 2003
3. E. Wróbel. Programowanie w języku assemblera MASM. Wydaw. Pol. Śląskiej, 2006

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi. Zamieszczanie na stronach internetowych zadań i materiałów do laboratoriów.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekt W_01 sprawdzany jest na egzaminie. Przykładowe pytania:

Określ czy następująca operacja jest dopuszczalna w języku MASM (wybrane przykłady):

```
mov si, dword ptr [esi]
```

```
mov zmA,zmB
```

Określ jaka będzie zawartość rejestru eax po wykonaniu następującego fragmentu kodu (przykład)

```
xor eax,eax
```

```
mov ecx,10
```

```
petla:
```

```
inc eax
```

```
loop petla
```

Efekt W_02 sprawdzany jest na egzaminie. Przykładowe pytania:

Jakie są główne zastosowanie programu assemblerowego?

Na czym polega etap asemlacji i konsolidacji i jak jest realizowany w MASM32.

Efekt U_01 sprawdzany jest na laboratoriach 1-6.

Efekt U_02 sprawdzany jest na laboratoriach 7-10.

Efekt K_01 sprawdzany jest na laboratoriach i egzaminie.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Ocena końcowa jest wystawiana na podstawie zajęć laboratoryjnych i egzaminu pisemnego. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na laboratoriach oraz z samodzielnie wykonanego zadania indywidualnego:

- Oceniane laboratoria (10 zajęć po 10pkt.)
- Zadanie indywidualne 40pkt

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów tj. co najmniej 71pkt. Za pisemne kolokwium można na nim uzyskać 100 pkt. Ostateczny wynik punktowy modułu oblicza się wg wzoru:

$$P=60(L/140)+40(E/100),$$

gdzie P-końcowy wynik punktowy (maksymalnie 100pkt.) , L-punkty uzyskane z części laboratoryjnej,

E-punktowy wynik egzaminu

Ocena z zajęć zależy od końcowego wyniku punktowego i wyznacza się w następujący sposób.

- 0-50 punktów – 2
- 51-60 punktów – 3
- 61-70 punktów - 3,5
- 71-80 punktów – 4
- 81-90 punktów – 4,5
- 91-100 punktów – 100

Sposób uzyskania punktów:

Laboratorium

1. Ocena udziału w laboratoriach oraz przygotowania się do tych zajęć: 140 pkt. (14 zajęć po 10 pkt.).

Wykład

2. Egzamin pisemny: 100 pkt.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godzin
Udział w ćwiczeniach laboratoryjnych	24 godziny
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	30 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny
Przygotowanie się do egzaminu i obecność na egzaminie	23 godzin
Sumaryczne obciążenie pracą studenta	100 godzin

Punkty ECTS za przedmiot	4 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godzin
Udział w ćwiczeniach laboratoryjnych	15 godziny
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	35 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny
Przygotowanie się do egzaminu i obecność na egzaminie	33 godziny
Sumaryczne obciążenie pracą studenta	100 godzin
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Wybrane paradygmaty programowania
Nazwa w języku angielskim:		Selected Programming Paradigms
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	drugi	
Semestr:	trzeci	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr hab. inż. Jerzy Tchórzewski, prof. uczelni
Imię i nazwisko prowadzących zajęcia:		dr hab. inż. Jerzy Tchórzewski, prof. uczelni dr Artur Niewiadomski, Mgr M. Nazarczuk mgr Wojciech Nabiałek, mgr Dariusz Ruciński
Założenia i cele przedmiotu:		Celem zajęć jest zapoznanie studentów z wybranymi paradygmatami programowania oraz wykorzystanie ich do praktycznej implementacji wybranych problemów
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia na temat popularnych i niszowych paradygmatów programowania i zna ich charakterystykę.	K_W06
W_02	Zna i rozumie zagadnienia na temat podstaw programowania deklaratywnego, w tym zna: klauzulową postać programów, rachunek zdań, rachunek predykatów, rezolucję i dowodzenie, programowanie w logice, mechanizmy wnioskowania, obiekty i relacje.	K_W06
W_03	Zna i rozumie zagadnienia z zakresu programowania w języku Prolog, w tym zna: składnię, znaki i operatory, struktury danych (m.in. pojęcie atomu, termu, zmiennej, struktury, drzewa, listy, fakty, itp.), zapytania, zmienne, koniunkcje, reguły, arytmetykę, itp. oraz posiada wiedzę o innych językach programowania deklaratywnego takich jak: LISP, CLIPS, itp.	K_W06
W_04	Zna i rozumie zagadnienia z zakresu: celów, nawracania i odcięcia, równości i unifikacji, przeszukiwania i porównywania rekurencyjnego, odwzorowania, łączenie struktur, akumulatorów, struktur różnicowych, itp.	K_W06

W_05	Zna i rozumie zagadnienia na temat urządzeń wejścia i wyjścia, czytania i pisania termów, zdań, plików, predykatów wbudowanych, tworzenia celów złożonych, obsługi plików, itp.	K_W06
W_06	Zna i rozumie zagadnienia z zakresu: przetwarzania list, zbiorów, sortowania, tworzenia bazy wiedzy, przeszukiwania grafów, śledzenia wykonywania programów, itp.	K_W06
W_07	Student zna podstawowe konstrukcje i typy danych wykorzystywane w programowaniu skrypcowym, proceduralnym i obiektowym, wspierane przez język Python.	K_W06
W_08	Student zna i rozumie charakterystyczne cechy funkcyjnego paradygmatu programowania. Zna podstawowe konstrukcje i typy danych wykorzystywane w programowaniu funkcyjnym wspierane przez język Python	K_W06
W_09	Zna i rozumie koncepcję lambda-wyrażeń, częściowej aplikacji argumentów funkcji, funkcji wyższego rzędu oraz typowe funkcje wyższego rzędu wspierane przez język Python, m.in. map, filter, reduce, zip.	K_W06
W_10	Student zna i rozumie ideę rekurencji, akumulatora, oraz spamiętywania i uleniwiania obliczeń.	K_W06
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi samodzielnie projektować i implementować proste systemy ekspertowe w środowisku SWI Prolog, przygotowywać projekty w postaci drzewa genealogicznego i na tej podstawie budować Bazę Wiedzy, w tym Bazę Faktów, Bazę Reguł, Bazę Zapytań oraz umie poprawiać i usuwać błędy w programach prologowych.	K_U22, K_U11
U_02	Potrafi zaprojektować z wykorzystaniem drzewa celów i zaimplementować w SWI Prolog oraz przetestować zaawansowany System Ekspertowy z wykorzystaniem list, arytmetyki, rekurencji, predykatów wbudowanych oraz posiada umiejętność modyfikacji Bazy Wiedzy, w tym umiejętność jej rozbudowywania, a także umie rozwijać swoje umiejętności programowania w językach programowania deklaratywnego.	K_U22
U_03	Potrafi projektować proste programy z wykorzystaniem inspirowanych kwantowo algorytmów klasycznych i je implementować na klasycznych komputerach z wykorzystaniem języków programowania bardzo wysokiego poziomu, takich jak m.in. j. Matlab lub j. Python.	K_U22
U_04	Potrafi sprawnie korzystać ze środowiska PyCharm (lub alternatywnego) w zakresie tworzenia aplikacji w języku Python.	K_U11
U_05	Potrafi definiować funkcje w języku Python, w tym funkcje rekurencyjne i funkcje wyższego rzędu. Potrafi wykorzystać deklarację sygnatury funkcji w celu lepszej kontroli nad typami danych.	K_U22
U_06	Potrafi wykorzystać mechanizmy typowe dla paradygmatu funkcyjnego, obiektowego, strukturalnego do rozwiązania postawionego problemu.	K_U22
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu

		kierunkowe go
K_01	Jest gotów do krytycznej oceny posiadanej wiedzy oraz jest przygotowany do uznania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych związanych z zagadnieniami programowania deklaratywnego, funkcyjnego, a także inspirowanych kwantowo zagadnień programowania obiektowego, systemowego i strukturalnego.	K_K01
Forma i typy zajęć:	Studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) Studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
<ol style="list-style-type: none"> 1. Wiedza z podstaw logiki matematycznej, rachunku zdań i rachunku predykatów. 2. Znajomość podstaw programowania, w tym imperatywnego, obiektowego, strukturalnego i systemowego oraz wybranych zagadnień z algorytmów i struktur danych. 3. Umiejętność samodzielnego rozwiązywania problemów i ich implementacji w dowolnych środowiskach programistycznych. 		
Treści modułu kształcenia:		
Część I.1 Wybrane Paradygmaty Programowania Deklaratywnego		
<ol style="list-style-type: none"> 1. Przegląd paradygmatów programowania. Programowanie imperatywne, systemowe, strukturalne, obiektowe, deklaratywne, funkcyjne, reaktywne, sterowane przepływem danych, itp. Rys historyczny i ewolucja języków programowania (1 godz.). 2. Projekty w języku Prolog oraz wybrane paradygmaty programowania deklaratywnego, w tym projektowanie Bazy Wiedzy w języku Prolog. Języki programowania deklaratywnego, w tym język Prolog. Fakty i reguły w reprezentacji wiedzy. Definicja terminu (stała, zmienna, struktura). Formuły atomowe. Prolog i programowanie w logice. Predykat podstawową jednostką języka Prolog. Fakty i reguły w Prologu. Zapytania. Składnia w Prologu. Obiekty i relacje. Równość i unifikacja, itp., konstruowanie Bazy Wiedzy (Drzewo Celów, Baza Faktów, Baza Reguł, Baza Zapytań, itp.). Spełnianie celów w regułach. Reguły proste i złożone. Przetwarzanie podprogramów. Klauzulowa postać Bazy Wiedzy (2 godz.). 3. Rachunek predykatów i rachunek zdań. Opis obiektów w Prologu. Proste predykaty niedeterministyczne i deterministyczne. Predykaty wejścia i wyjścia. Predykaty wbudowane. Predykaty standardowe. Postać klauzulowa. Zapis klauzul. Rezolucja i dowodzenie twierdzeń. Klauzule Horna. Rachunek predykatów. Przekształcenia w rachunku predykatów prowadzące od postaci predykatowej do postaci klauzulowej. Formalny opis języka predykatów (2 godz.). 4. Mechanizmy nawrotu i odcięcia w Prologu. Deklaratywność Prologu. Mechanizmy inteligentnego uzgadniania. Relacje i reguły. Koniunkcje. Nawracanie. Cele i nawracanie. Zapytania i nawroty. Typowe zastosowanie odcięcia. Potwierdzenie wyboru reguły. Użycie predykatu odcięcia. Korzystanie ze struktur danych. Arytmetyka i tautologie w Prologu. Struktury a drzewa. Listy. Łączenie struktur. Akumulatory. Struktury różnicowe. Złożone struktury danych. Przetwarzanie drzew i list. Przetwarzanie zbiorów (2 godz.). 		
Część I.2 Wybrane Paradygmaty Programowania Kwantowego		
<ol style="list-style-type: none"> 5. Informatyka Kwantowa. Kubit, w tym kubit, interpretacje kubit, stany systemu i stany kwantowe, kwantowe stany mieszane, superpozycja stanów wykluczających się, pomiar (odczyt) stanu kwantowego, ogólne operacje kwantowe, przestrzeń Hilberta, w tym macierze unitarne, postulaty informatyki kwantowej, obliczenia kwantowe z wykorzystaniem bramek kwantowych, liczby zespolone w obliczeniach kwantowych, obwody kwantowe w obliczeniach kwantowych, kwantowy stan splątany, dostępne symulatory obliczeń kwantowych, w tym symulator obliczeń kwantowych w środowisku MATLAB-a i Simulink-a, itp. 6. Obliczenia kwantowe na bramkach kwantowych. Kwantowe przetwarzanie informacji, rola operatorów w informatyce kwantowej, w tym wartości własnych operatorów hermitowskich, macierz gęstości, w tym wyznaczanie śladu, funkcja falowa, obserwabla, synteza obwodów kwantowych, 		

algorytmy kwantowe, teleportacja kwantowa, ewolucja unitarna, rozkład Schmidta stanów kwantowych, inspiracje kwantowe metod klasycznych, itp.

Część II. Wybrane Paradygmaty Programowania Funkcyjnego

- 7. Wprowadzenie do języka Python.** Podstawowe typy danych. Operatory. Wyrażenia i ich wartościowanie. Funkcje i ich argumenty. Wyrażenia warunkowe. Pętle.
- 8. Funkcje.** Definiowanie funkcji w języku Python. Zwracanie wartości przez funkcję. Sygnatury funkcji. Funkcje czyste. Efekty uboczne funkcji. Wyrażenia lambda. Funkcje rekurencyjne. Rekursja a wykorzystanie stosu.
- 9. Typy danych.** Krotki. Zbiory. Listy. Słowniki. Obiekty. Elementy programowania strukturalnego i obiektowego.
- 10. Listy.** Własności list. Operacje na listach. Przetwarzanie list za pomocą funkcji rekurencyjnych. Rekursja ogonowa i wykorzystanie akumulatora.
- 11. Funkcje wyższego rzędu i obliczenia leniwe.** Definicja i przykłady funkcji wyższego rzędu (FWR). Przetwarzanie list za pomocą FWR. Funkcje z modułu List. Obliczenia leniwe. Generatory. Iteratory.

Literatura podstawowa:

- G. Brzykcy, A. Meissner: Programowanie w Prologu i programowanie funkcyjne: materiały do ćwiczeń. Wyd. Politechniki Poznańskiej. Poznań 1999, stron 110.
- W. F. Clocksin, C. S. Mellish: Prolog. Programowanie. Helion. Warszawa 2003, stron 276.
- M. Gorelick, I. Ozsvald: Wysoko wydajny Python: efektywne programowanie w praktyce. Helion 2021.
- E. Matthes: Python. Instrukcje dla programisty. Helion 2016.
- Tchórzewski J.: Wykłady i instrukcje laboratoryjne z wybranych paradygmatów programowania deklaratywnego i kwantowego. Monografia dydaktyczna w przygotowywaniu do złożenia. WN UPH, Siedlce (wersja elektroniczna z września 2016 r., ostatnia aktualizacja rozszerzona o paradygmaty kwantowe z września 2021, jest dostępna dla studentów w postaci instrukcji laboratoryjnych oraz Print Screen-ów z wykładów przygotowanych w MS Power Point).

Literatura dodatkowa:

- M. Ben-Ari: Logika matematyczna w informatyce. WNT. Warszawa 2001, stron 346.
- U. Nilsson, J. Małuszynski: Logic. Programming and Prolog. John Wiley & Sons, 1990.
- R. Johansson: Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib. Helion 2021.
- P. V. Roy, S. Haridi: Programowanie. Koncepty, techniki i modele. Helion. Warszawa 2005.
- J. Tchórzewski: Metody sztucznej inteligencji i informatyki kwantowej w ujęciu teorii sterowania i systemów. Wydawnictwo UPH. Siedlce 2021.
- W. Wójcik: Zasada rezolucji. Metoda automatycznego wnioskowania. PWN. Warszawa 1991.

Planowane formy/działania/metody dydaktyczne:

Wykład z aktywną komunikacją wspomagany technikami multimedialnymi. Udostępnianie studentom skanów wykładów w postaci prezentacji w MS Power Point oraz instrukcji do poszczególnych tematów ćwiczeń laboratoryjnych.

Ćwiczenia laboratoryjne są prowadzone według następującego układu tematów i laboratoriów:

Część I. Wybrane Paradygmaty Programowania Deklaratywnego i Kwantowego

Temat 1 (Lab1-2): Projektowanie Drzewa Genealogicznego i implementacja Bazy Wiedzy w środowisku SWI Prolog na temat rodziny (2 x 2godz.).

Temat 2 (Lab3-4): Projektowanie Drzewa Celów i implementacja Systemu Ekspertowego w SWI Prolog na wskazany przez prowadzącego problem (2 x 2 godz.).

Temat 3 (Lab5-6): Projektowanie inspirowanych kwantowo prostych algorytmów klasycznych z wykorzystaniem j. Matlab/j. Python/j. Sphinx, itp. lub z wykorzystaniem symulatora środowiska MATLAB i Simulink, jak np. Qubit4MATLA, w uzgodnionym z prowadzącym zakresie (2 x 2 godz.).

Część II.

Temat 4 (Lab 7): Wprowadzenie do języka Python. Środowisko PyCharm, Operatory. Wyrażenia i ich wartościowanie. Funkcje i ich argumenty. Wyrażenia warunkowe. Pętle (2 godz.).

Temat 5 (Lab 8): Funkcje. Definiowanie funkcji w języku Python. Zwracanie wartości przez funkcję. Sygnatury funkcji. Funkcje czyste. Efekty uboczne funkcji. Wyrażenia lambda. (2 godz.).

Temat 6 (Lab 9-10): Typy danych. Krotki. Zbiory. Listy. Słowniki. Obiekty. Elementy programowania strukturalnego i obiektowego (2 x 2godz.).

Temat 7. (Lab 11): Rekurencja. Funkcje rekurencyjne. Rekursja a wykorzystanie stosu. Własności list. Operacje na listach. Przetwarzanie list za pomocą funkcji rekurencyjnych, wykorzystanie akumulatora (2 godz.).

Temat 8. (Lab 12): Funkcje wyższego rzędu i obliczenia leniwe. Definicja i przykłady funkcji wyższego rzędu (FWR). Przetwarzanie list za pomocą FWR. Obliczenia leniwe. Generatory. Iteratory (2 godz.).

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekty W_01 - W_10 będą sprawdzane w formie kolokwium oraz na egzaminie.

Efekty U_01 - U_06 będą sprawdzane przy zaliczeniu każdego z tematów laboratoryjnych.

Efekt K_01 będzie sprawdzany przy każdym kontakcie ze studentem na laboratorium, wykładach, konsultacjach, egzaminie, itp.

Forma i warunki zaliczenia:

Moduł kończy się **egzaminem**.

Zaliczenie laboratorium.

Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na zajęciach z nauczycielem akademickim, za które można łącznie uzyskać maksymalnie 100 p., odpowiednio w części I oraz w części II do 50p. według zasady:

Część I:

ćwiczenia laboratoryjne 1-2 oraz 3-4 (2 tematy projektów indywidualnych x 20 p.) - 40 p.

ćwiczenie laboratoryjne 5-6 (1 temat ćwiczenia laboratoryjnego x 10p) - 10 p.

Razem – 50 p.

Część II:

ćwiczenia laboratoryjne 7, 8, 11, 12 (4 tematy ćwiczeń laboratoryjnych x 5 p.) - 20 p.

ćwiczenie laboratoryjne 9-10 (jeden temat ćwiczeń laboratoryjnych x 10 p.) – 10 p.

kolokwium - 20 p.

Razem – 50 p.

Ogółem (2 x 50p) - 100 p.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania więcej niż połowy punktów z każdej części laboratorium (od 51%, to jest **od 26 p.**) ze wszystkich form aktywności studenta odrębnie w zakresie Wybranych paradygmatów programowania deklaratywnego i kwantowego oraz odrębnie w zakresie Wybranych paradygmatów programowania funkcyjnego.

Egzamin

Egzamin pisemny (test lub zadania problemowe) składa się z dwóch części, odrębnie w z Wybranych paradygmatów programowania deklaratywnego i kwantowego oraz odrębnie z Wybranych paradygmatów programowania funkcyjnego. Za każdą z części egzaminu student może uzyskać do 50 p. Wykłady będą zaliczone w przypadku uzyskania z każdej części egzaminu odrębnie przynajmniej 26 p.

Razem (2 x 50 p) – 100p.

Ocena z każdej części przedmiotu

Ocena z Wybranych paradygmatów programowania deklaratywnego i kwantowego oraz ocena z Wybranych paradygmatów programowania funkcyjnego jest wyznaczana jako średnia ocen uzyskanych z ćwiczeń laboratoryjnych oraz z wykładów odrębnie odpowiedniej części.

Oceny z zaliczenia poszczególnych części przedmiotu (odrębnie z Wybranych paradygmatów programowania deklaratywnego i kwantowego oraz odrębnie z Wybranych paradygmatów programowania funkcyjnego).

Z każdej części przedmiotu student może uzyskać do 100 pkt (do 50 p. z egzaminu i do 50 p. z laboratoriów), przy czym istnieje możliwość uzyskania dodatkowych punktów z tytułu różnych aktywności studentów na ćwiczeniach laboratoryjnych, wykładach oraz po zajęciach w tematyce związanej z realizowanym programem w ramach modułu. W zależności od liczby punktów zdobytych w ramach ocenianej części, odrębnie ćwiczeń laboratoryjnych oraz odrębnie odpowiedniej części wykładów, student otrzymuje ocenę:

- 0-25 pkt. - 2 (ndst)
- 26-30 pkt. - 3 (dst)
- 31-35 pkt. – 3,5 (dst+)
- 36-40 pkt. - 4 (db)
- 41-45 pkt. - 4,5 (db+)
- 46-50 pkt. - 5 (bdb).

Ocena z danej części przedmiotu jest średnią ocen z laboratorium oraz wykładów i wynosi:

- 2,0; 2,5 - ndst (F)
- 3,00 - dst (E)
- 3,25; 3,5 - dst+ (D)
- 3,75; 4,0 - db (C)
- 4,25; 4,5 - db+ (B)
- 4,75; 5,0 - bdb (A).

Ocena końcowa z przedmiotu

Ocena końcowa z przedmiotu jest średnią z obu części modułu, to jest z Wybranych paradygmatów programowania deklaratywnego i kwantowego oraz z Wybranych paradygmatów programowania funkcyjnego. Ocena końcowa z przedmiotu w zależności od uzyskanej średniej jest następująca (w nawiasach ocena wg skali ECTS:

2,0; 2,5 - niedostateczna (F),
3,0 - dostateczna (E),
3,25; 3,5 - dostateczna plus (D),
4,0 - dobra (C),
4,25; 4,5 - dobra plus (B),
5,0 - bardzo dobra (A).

Poprawy:

Istnieje możliwość jednorazowej poprawy kolokwium w toku zajęć laboratoryjnych oraz jednej poprawy w sesji egzaminacyjnej, a także dwukrotnej poprawy jednego z tematów ćwiczeń laboratoryjnych w toku trwania semestru. Poprawa ćwiczeń laboratoryjnych nie jest możliwa po terminie wpisów do USOS.

Egzamin poprawkowy jest jednokrotny w sesji egzaminacyjnej, odrębnie z każdej części przedmiotu.

Studia stacjonarne

Bilans punktów ECTS:

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do tematów ćwiczeń laboratoryjnych i realizacja projektów indywidualnych, a także opracowanie sprawozdań	15 godz.
Udział w konsultacjach	2 godz.
Studia indywidualne z tematów realizowanych na wykładach i na ćwiczeniach laboratoryjnych oraz końcowe przygotowanie się do egzaminu pisemnego	13 godz.

Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS
Studia niestacjonarne	
Bilans punktów ECTS:	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do tematów ćwiczeń laboratoryjnych i realizacja projektów indywidualnych, a także opracowanie sprawozdań	15 godz.
Udział w konsultacjach z przedmiotu	2 godz.
Studia indywidualne z tematów realizowanych na wykładach i na ćwiczeniach laboratoryjnych oraz końcowe przygotowanie się do sprawdzianu pisemnego	28 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Grafika komputerowa
Nazwa w języku angielskim:		Computer Graphics
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	Drugi	
Semestr:	trzeci	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		Mirosław Barański
Imię i nazwisko prowadzących zajęcia:		Mirosław Barański
Założenia i cele przedmiotu:		Założono, że studenci poznają ważniejsze zagadnienia związane z grafiką komputerową. Ma to być podstawą do rozwijania dalszych zainteresowań związanych z tym zakresem jak również bazą do zastosowań grafiki komputerowej w projektowanych systemach informatycznych uwzględniających rozważane zagadnienia. Celem kursu jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z algorytmami i metodami grafiki komputerowej oraz zdobycie praktycznych umiejętności w implementacji tych algorytmów i metod.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia związane z algorytmami grafiki rastrowej i wektorowej	K_W01, K_W06
W_02	Zna i rozumie zagadnienia związane z przekształceniami 2D i 3D	K_W01, K_W06
W_03	Zna i rozumie zagadnienia związane z reprezentacją obiektów graficznych oraz zna i rozumie podstawowe operacje na tych obiektach	K_W06
W_04	Zna i rozumie zagadnienia związane z wykorzystaniem funkcji graficznych html5 oraz SVG	K_W06
W_05	Zna i rozumie standardy stosowane w przetwarzaniu obrazu	K_W09
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrąfi pozyskiwać informacje dotyczące z literatury, baz danych i innych źródeł związane z grafiką komputerową, potrafi integrować pozyskane informacje z innymi zadaniami	K_U01
U_02	Potrąfi samodzielnie planować i realizować uczenie się przez całe życie, m.in. w celu podnoszenia kompetencji zawodowych	K_U06

U_03	Potrafi wykorzystać właściwie dobrane środowiska programistyczne i narzędzia komputerowe do projektowania i weryfikacji systemów informatycznych związanych z grafiką komputerową	K_U10, K_U11
U_04	Potrafi zaimplementować podstawowe, wybrane, algorytmy związane z grafiką komputerową.	K_U01, K_U02, K_U10, K_U16
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do krytycznej oceny posiadanej wiedzy z zakresu grafiki komputerowej	K_K01
K_02	Jest gotów do uznania znaczenia wiedzy w rozwiązywaniu problemów praktycznych i poznawczych z zakresu grafiki komputerowej	K_K01
Forma i typy zajęć:	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
Znajomość analizy matematycznej, algebry, podstaw programowania oraz znajomość technologii WWW.		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> Wstęp. Prezentacja dziedziny. Podstawowe wiadomości i pojęcia grafiki komputerowej: formy przetwarzania danych obrazowych, grafika, przetwarzanie obrazów, rozpoznawanie obrazów, formy danych obrazowych, grafika wektorowa i rastrowa, przekształcenia form danych obrazowych, Algorytmy rastrowe: Algorytm rysowania odcinka. Algorytmy rysowania krzywych. Wypełnianie obszarów: Rodzaje wnętrza i brzegów, Algorytmy wypełniania obszarów dla grafiki rastrowej i wektorowej. Geometria na płaszczyźnie R2: przekształcenia punktu na płaszczyźnie: translacja, obrót, skalowanie, jednokładność, składanie przekształceń: obrót względem dowolnego punktu, skalowanie, Operacje na wielokątach: okienkowanie i obcinanie, wyznaczanie powłoki wypukłej zbioru punktów, triangulacja wielokątów, Geometria w przestrzeni R3: Podstawowe pojęcia i obiekty w przestrzeni R3, przekształcanie punktu w R3: translacja, obrót, skalowanie, obroty względem dowolnej prostej, przekształcenie 3-punktowe, Rzutowanie: przekształcenie układu danych do układu obserwatora, rzutowanie równoległe i perspektywiczne, własności rzutów, Reprezentacje obiektów: drzewa czwórkowe i ósemkowe, operacje na drzewach czwórkowych, reprezentacje wielościennie, Grafika w HTML5. Istota funkcji graficznych w HTML5, funkcje związane z rysowaniem podstawowych prymitywów graficznych w grafice komputerowej, przekształceniami geometrycznymi i z przetwarzaniem obrazów. Porównanie środowisk programistycznych HTML5 i SVG. Wybrane funkcje graficzne w SVG. 		
Literatura podstawowa:		
<ol style="list-style-type: none"> Michał Jankowski: Elementy grafiki komputerowej. WNT 2006. Eric Rowell: HTML5 Canvas Receptury. Helion, 2013. 		
Literatura dodatkowa:		
<ol style="list-style-type: none"> Theo Pavlidis: Grafika i przetwarzanie obrazów. WNT Jan Zabrodzki i inni: Grafika komputerowa, metody i narzędzia. WNT 1994. James D. Foley: Wprowadzenie do grafiki komputerowej. WNT 2001. 		
Planowane formy/działania/metody dydaktyczne:		
Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia laboratoryjne wspomagane technikami multimedialnymi. Zamieszczanie na stronach internetowych problemów i zadań laboratoryjnych.		
Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:		
<p>Efekty W_01 – W_05 będą sprawdzane na kolokwium ustnym. Zadania będą dotyczyły wybranych problemów przetwarzania obrazów, przykładowe zadania:</p> <ul style="list-style-type: none"> Omów istotę algorytmu okienkowania. Gdzie ten algorytm powinien być wykorzystywany. Na czym polega wyznaczanie powłoki wypukłej. Podaj przykład problemu (niekoniecznie związanego z grafiką komputerową), gdzie można go wykorzystać. 		

Efekty U_01 - U_02 będą systematycznie sprawdzane na zajęciach. Zadania na następne laboratorium muszą być dostępne co najmniej tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich samodzielnie lub korzystając z konsultacji przygotować.

Efekty U_03-U_04 będą sprawdzane systematycznie na zajęciach laboratoryjnych, przykładowe zadanie:

- Dany jest ciąg n-punktów w przestrzeni R^3 . Napisz program w języku java script, który zwizualizuje te punkty w rzucie równoległym nieortogonalnym aksonometrii wojskowej.

Zadania na następne laboratorium muszą być dostępne co najmniej tydzień przed zajęciami.

Efekty K_K01, K_K02 będą sprawdzane na kolokwium ustnym. Przykładowe zadanie:

- W jaki sposób metody i narzędzia grafiki komputerowej mogą być wykorzystane w bazach danych.

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Ocena końcowa jest wystawiana na podstawie zajęć laboratoryjnych i jednego kolokwium ustnego przeprowadzonego po ostatnim wykładzie w czasie konsultacji lub innym terminie uzgodnionym ze studentami przed końcem semestru. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać maksymalnie 60pkt. Zaliczenie zajęć laboratoryjnych następuje w przypadku uzyskania co najmniej 30pkt. Za ustne kolokwium można na nim uzyskać do 40 pkt. Zaliczenie kolokwium jest możliwe po uzyskaniu co najmniej 20 pkt. Ocena końcowa z modułu (po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) może być następująca (w nawiasach ocena wg skali ECTS):

- 0 – 50 pkt: niedostateczna (F),
- 51 – 60 pkt: dostateczna (E),
- 61 – 70 pkt: dostateczna plus (D),
- 71 – 80 pkt: dobra (C),
- 81 – 90 pkt: dobra plus (B),
- 91 – 100 pkt: bardzo dobra (A).

Poprawy:

Poprawa kolokwium ustnego w czasie sesji egzaminacyjnej.

Uwaga: Istnieje możliwość zwolnienia z kolokwium ustnego studentów wyróżniających się na zajęciach laboratoryjnych. Warunkiem koniecznym zwolnienia z egzaminu jest uzyskanie 90% punktów możliwych do zdobycia w trakcie regularnych zajęć laboratoryjnych. Decyzję o ewentualnym zwolnieniu podejmuje osoba przeprowadzająca egzamin po zasięgnięciu opinii (poprzez rozmowę) osób prowadzących zajęcia. Decyzję o zwolnieniu prowadzący wykład przekazuje studentom nie później niż 2 tygodnie przed końcem semestru.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	15 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do kolokwium i obecność na kolokwium	10 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	28 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do kolokwium i obecność na kolokwium	15 godz.

Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Wybrane zastosowania języka Python
Nazwa w języku angielskim:		Selected applications of the Python language
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:		Wydział Nauk Ścisłych i Przyrodniczych
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		Fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		Pierwszego stopnia
Rok studiów:	drugi	
Semestr:	Trzeci	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		Dr Piotr Świtalski
Imię i nazwisko prowadzących zajęcia:		Dr Piotr Świtalski, Dr Artur Niewiadomski, Mgr Maciej Przychodzki
Założenia i cele przedmiotu:		Celem kursu jest przekazanie studentom wiedzy na temat wybranych zastosowań i bibliotek języka Python oraz zdobycie praktycznych umiejętności ich wykorzystania w implementowanych systemach
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie podstawowe konstrukcje i typy danych wykorzystywane w programowaniu skryptowym, proceduralnym, obiektowym i funkcyjnym wspierane przez język Python.	K_W06
W_02	Zna i rozumie koncepcję lambda-wyrażeń, częściowej aplikacji argumentów funkcji, funkcji wyższego rzędu oraz standardowe funkcje wyższego rzędu wspierane przez język Python, m.in. map, filter, reduce, zip.	K_W06
W_03	Zna wybrane biblioteki języka Python, w szczególności te związane z obliczeniami naukowymi i analizą danych, m.in. NumPy, SciPy, Matplotlib	K_W06
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi sprawnie korzystać ze środowiska PyCharm (lub alternatywnego) w zakresie tworzenia aplikacji w języku Python. Potrafi posługiwać się	K_U11

	notatnikami Jupyter w wybranym środowisku chmurowym (np. Google Colab)	
U_02	Umie dobrać i zastosować odpowiednie konstrukcje języka Python do praktycznego rozwiązania problemu. Potrafi zaprojektować i zaimplementować program w języku Python wg zadanych wymagań	K_U22
U_03	Potrafi przeanalizować kod źródłowy napisany w języku Python, a także znaleźć i naprawić typowe błędy	K_U22
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
Forma i typy zajęć:	Studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) Studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
<p>Warunkiem uczestnictwa w zajęciach jest wcześniejsze uzyskanie zaliczenia z następujących przedmiotów:</p> <ul style="list-style-type: none"> • Matematyka dyskretna • Analiza matematyczna • Podstawy programowania <p>lub znajomość literatury obowiązującej w tym przedmiocie. Student musi mieć opanowane podstawy logiki matematycznej, kombinatoryki i statystyki.</p> <p>Ponadto wymagana jest znajomość podstaw strukturalnego lub obiektowego języka programowania, a w tym umiejętność definiowania typów, posługiwanie się instrukcjami iteracyjnymi, podprogramami i rekurencją.</p>		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> 1. Wprowadzenie do języka Python. Rys historyczny, wersje, charakterystyka. Python 2 i Python 3. Zastosowania. Mocne i słabe strony języka Python. 2. Środowisko programistyczne Pycharm, podstawowa składnia i konstrukcje. Funkcje. Skrypty. Zapoznanie z wybranymi narzędziami, np. anaconda, Jupyter Notebook, Google Colab. 3. Proste i złożone typy danych. Listy, krotki, zbiory, słowniki. Wyrażenia logiczne. Iteratory, generatory. 4. Podstawowe koncepcje i techniki przetwarzania funkcyjnego. Wyrażenia lambda, częściowa aplikacja argumentów funkcji, funkcje wyższego rzędu. 5. Podstawy programowania obiektowego w języku Python. Dziedziczenie, metody specjalne, dekoratory, wyjątki. 6. Aplikacje webowe w języku Python: biblioteki flask i Django. 7. Wstęp do obliczeń naukowych i symbolicznych, analizy danych i uczenia maszynowego. Biblioteki: NumPy, Matplotlib, pandas, SciPy, z3py, scikit-learn 		
Literatura podstawowa:		
<ol style="list-style-type: none"> 1. R. Johansson, Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib. Helion 2021. 2. M. Gorelick, I. Ozsvald. Wysoko wydajny Python: efektywne programowanie w praktyce. Helion 2021. 3. E. Matthes. Python. Instrukcje dla programisty. Helion 2016. 		

4. Literatura dodatkowa:
<p>5. J. Grus. Data science od podstaw : analiza danych w Pythonie. Helion 2018.</p> <p>6. S. Raschka, V. Mirjalili. Pyton. Uczenie maszynowe. Wydanie 2. Helion 2019.</p> <p>7. Peter Farrell. Matematyczne przygody z Pythonem: ilustrowany podręcznik do nauki matematyki przez programowanie. Wydanie 1. Wydawnictwo Naukowe PWN 2019.</p> <p>8. A. Niewiadomski, P. Switalski, T. Sidoruk, W. Penczek. SMT-Solvers in Action: Encoding and Solving Selected Problems in NP and EXPTIME. Sci. Ann. Comp. Sci. 28(2): 269-288, 2018, doi:10.7561/SACS.2018.2.269</p>
Planowane formy/działania/metody dydaktyczne:
<p>Wykład tradycyjny wspomagany technikami multimedialnymi. Zajęcia laboratoryjne – zajęcia praktyczne z wykorzystaniem języka Python i wybranego środowiska programistycznego (PyCharm lub alternatywne). Praca z notatnikami Jupyter w środowisku Google Colab, lub alternatywnym (np. Azure)</p>
Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:
<p>Efekty W_01 – W_03 będą sprawdzane na egzaminie pisemnym. Na egzaminie pisemnym pytania będą dotyczyły wybranych zagadnień teoretycznych oraz praktycznych związanych z zastosowaniami języka Python. Przykładowe pytania i zadania:</p> <ul style="list-style-type: none"> • Wymień wbudowane w język Python typy kolekcji i scharakteryzuj ich własności. • Co to jest funkcja wyższego rzędu? Podaj definicję i przykład w języku Python. • Na czym polega dziedziczenie? Podaj definicję i przykład w języku Python. • Porównaj wbudowane w Python listy z typem numpy.ndarray. Podaj przykłady obydwu struktur. <p>Efekty U_01, U_02 i U_03 sprawdzane będą sukcesywnie i oceniane po każdym laboratorium, również poprzez sprawdzanie prac domowych i kolokwii. Przykładowe zadania:</p> <ul style="list-style-type: none"> • Zdefiniuj funkcję w języku Python, która przyjmuje jako argumenty współczynniki równania kwadratowego (w postaci ogólnej) i wypisuje na ekran jego rozwiązania, lub komunikat o ich braku • Zdefiniuj klasę Ułamek i zaimplementuj operatory arytmetyczne: dodawanie, odejmowanie, mnożenie i dzielenie ułamków zwykłych jako odpowiednie metody specjalne <p>Efekt K_01 będzie weryfikowany, w oparciu o posiadaną wiedzę i umiejętności w czasie zajęć laboratoryjnych, a także będzie sprawdzany na egzaminie.</p>
Forma i warunki zaliczenia:
<p>Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny częściowe uzyskane na regularnych zajęciach z nauczycielem akademickim. Na tej formie zajęć student może maksymalnie uzyskać 60 pkt. Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania powyżej 30 punktów.</p> <p>Poprawa laboratorium: najpóźniej w ciągu 3 tygodni od danych zajęć. Poprawa nie jest możliwa po zakończeniu semestru.</p> <p>Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 40 pkt. Egzamin będzie zaliczony w przypadku uzyskania powyżej 20 pkt.</p> <p>Ocena końcowa z przedmiotu, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS):</p> <ul style="list-style-type: none"> • 0 – 50 pkt: niedostateczna (F), • 51 – 60 pkt: dostateczna (E), • 61 – 70 pkt: dostateczna plus (D), • 71 – 80 pkt: dobra (C), • 81 – 90 pkt: dobra plus (B),

- 91 – 100 pkt: bardzo dobra (A).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godzin
Udział w ćwiczeniach laboratoryjnych	24 godziny
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	12 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny
Studia indywidualne z tematów realizowanych na wykładach	8 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	10 godzin
Sumaryczne obciążenie pracą studenta	77 godzin
Punkty ECTS za przedmiot	3

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godzin
Udział w ćwiczeniach laboratoryjnych	15 godzin
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	15 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny
Studia indywidualne z tematów realizowanych na wykładach	15 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	15 godzin
Sumaryczne obciążenie pracą studenta	77 godzin
Punkty ECTS za przedmiot	4

