

Karta przedmiotu

Nazwa i kod przedmiotu	Wstęp do programowania, PG_00178695						
Kierunek studiów	Informatyka i ekonometria (O)						
Data rozpoczęcia studiów	październik 2026 r.		Rok akademicki realizacji przedmiotu		2026/2027		
Poziom kształcenia	I stopnia - licencjackie		Grupa zajęć		Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne		Sposób realizacji		na uczelni		
Rok studiów	1		Język wykładowy		polski		
Semestr studiów	2		Liczba punktów ECTS		5.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		zaliczenie		
Jednostka prowadząca	Rektor -> Wydział Zarządzania -> Katedra Informatyki Ekonomicznej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr hab. Jerzy Auksztol, prof. UG				
	Prowadzący zajęcia z przedmiotu						
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	45.0	0.0	0.0	60
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	60		4.0		61.0	125
Cel przedmiotu	Opanowanie umiejętności programowania komputerów, ze szczególnym uwzględnieniem podstawowych konstrukcji języków programowania C# oraz Python i związanych z nimi technik analizy danych.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[IiEL3_W06] W zaawansowanym stopniu zna i rozumie procesy i metody tworzenia, rozwoju i zapewnienia odpowiednich warunków użytkowania narzędzi informatycznych lub statystycznych, w szczególności usprawniających funkcjonowanie człowieka i organizacji	Zna zasady programowania obiektowego od strukturalnego. Posiada wiedzę potrzebną do oceny, który paradygmat programowania jest właściwy dla rozwiązania danego problemu algorytmicznego i/lub biznesowego. Zna metody matematyczne, statystyczne i ekonometryczne w programach napisanych w językach C# i Python.	[SW4] test/egzamin - ustny lub pisemny
	[IiEL3_U12] Potrafi projektować i implementować systemy informatyczne wspierające działalność przedsiębiorstw oraz wykorzystywać nowoczesne technologie ICT w zarządzaniu i komunikacji biznesowej	Potrafi pisać programy w języku C# i Python na podstawie zgłoszonych oczekiwań użytkowników. Potrafi krytycznie ocenić programy komputerowe w zakresie efektywności zastosowanych w nich algorytmów i struktur danych.	[SU2] prezentacja/projekt/referat/raport [SU4] test/egzamin - ustny lub pisemny
	[IiEL3_U02] Potrafi dobrać lub konstruować narzędzia ekonometryczne, informatyczne lub statystyczne oraz stosować je do opisu i rozwiązywania problemów ekonomicznych i społecznych	Potrafi rozwiązać złożone problemy algorytmiczne i biznesowe w języku C# oraz Python. Potrafi projektować programy komputerowe rozwiązujące problemy biznesowe zgodnie z podejściem obiektowym. Potrafi projektować programy komputerowe z wykorzystaniem technologii baz danych.	[SU2] prezentacja/projekt/referat/raport
Treści przedmiotu	A. Problematyka wykładu • omówienie języków formalnych w odniesieniu do składni, semantyki i pragmatyki; klasyfikacja języków programowania wg paradygmatu programowania deklaratywnego, funkcjonalnego, logicznego oraz obiektowego. • algorytmy i ich formalizacja - pojęcie algorytmu; techniki jego opisu na przykładzie algorytmów równań liniowych, kwadratowych oraz największego wspólnego dzielnika, • systemy numeryczne w komputerach: system dwójkowy, ósemkowy, dziesiętny i szesnastkowy; reprezentacja liczb całkowitych i zmiennoprzecinkowych, • elementy składowe programowania deklaratywnego: typy danych (podstawowe, tablice, rekordy, pliki, typy wyliczeniowe i wskaźnikowe), zmienne, operatory, wyrażenia, instrukcje pętli, wyboru; podział na moduły i podprogramy (metody), ciało podprogramu oraz instrukcja wywołania metody, przekazywania parametrów przez wartość i referencję; widoczność i czas życia zmiennych, • elementy składowe programowania obiektowego w języku C# i Python: klasy, obiekty, atrybuty, metody, konstruktory, destruktory, dziedziczenie, hermetyzacja, klasy abstrakcyjne, interfejsy • struktury danych: lista dwukierunkowa; stos; kolejka; tablica mieszająca (haszująca); drzewo binarne, • algorytmy wyszukiwania i sortowania sortowanie bąbelkowe, przez wybieranie, scalanie i szybkie; wyszukiwanie liniowe i binarne, • techniki programowania: rekurencja (rekursja) na przykładzie silni, ciągu Fibonacciego oraz algorytmu Euklidesa; programowanie dynamiczne; technika dzieli i rządź, • wybrane zastosowania: big data, web scraping, machine learning oraz text mining. B. Problematyka ćwiczeń laboratoryjnych • omówienie języków formalnych - omówienie środowiska programistycznego; napisanie i uruchomienie pierwszego programu; przedstawienie pojęć składni, semantyki i pragmatyki, • algorytmy i ich formalizacja napisanie algorytmów równań liniowych i kwadratowych przy wykorzystaniu jednej z technik opisu, • systemy numeryczne oraz elementy programowania deklaratywnego implementacja algorytmu równań liniowych i kwadratowych; opracowanie algorytmu i napisanie programu realizującego wzajemną zamianę reprezentacji dwójkowej, ósemkowej, dziesiętnej i szesnastkowej, • struktury danych - opracowanie podstawowych struktur danych takich jak: lista, drzewo binarne, stos, kolejka, • algorytmy wyszukiwania i sortowania implementacja wybranych algorytmów sortowania i wyszukiwania, złożoność i efektywność obliczeniowa przeprowadzenie porównania złożoności obliczeniowej wcześniej napisanych algorytmów sortowania, • wybrane techniki programowania opracowanie algorytmu silni i ciągu Fibonacciego przy wykorzystaniu technik rekurencyjnych i nie-rekurencyjnych ze wskazaniem wad i zalet każdej z nich, • implementacja wybranych zadań implementujących web scraping, machine learning oraz text mining.		
Wymagania wstępne i dodatkowe	Podstawowa wiedza z matematyki poziomu licealnego oraz umiejętność korzystania z systemu operacyjnego komputera stacjonarnego.		

Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Kolokwium	51.0%	30.0%
	Test wiedzy z wykładu	51.0%	40.0%
	Projekt semestralny	51.0%	30.0%
Zalecana lista lektur	Podstawowa lista lektur	• Informatyka ekonomiczna. (ang. Business Informatics), pod red. S. Wrycza., J. Maślankowski, Wydawnictwo Naukowe PWN, Warszawa 2019, rozdziały: 7, 8. • Boduch A., Wstęp do programowania w języku C#, Helion, Warszawa 2006. • Lutz M., Ascher D., Python. Wprowadzenie (ang. Python. Introduction), Wydawnictwo Helion, Gliwice 2011. • Müller A. C., Guido S., Machine learning, Python i data science. Wprowadzenie (ang. Machine learning, Python and data science. Introduction), Wydawnictwo Helion, Gliwice 2021.	
	Uzupełniająca lista lektur	Matthes E., Python. Instrukcje dla programisty (ang. Python. Programmer's Manual), Wydawnictwo Helion, Gliwice 2020.	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	• Przygotowanie programów komputerowych wyjaśniających funkcjonowanie podstawowych konstrukcji języka C# i Python. • Przygotowanie programów komputerowych rozwiązujących ten sam problem algorytmiczny w językach wielu paradygmatów,np. strukturalnego, obiektowego i funkcyjnego. • Przygotowanie programów komputerowych dla algorytmów równań liniowych i kwadratowych, • Przygotowanie programu komputerowego realizującego wzajemną zamianę reprezentacji dwójkowej, ósemkowej, dziesiętnej i szesnastkowej. • Przygotowanie programów komputerowych korzystających z podstawowych struktur danych takich jak: lista, drzewo binarne, stos, kolejka, • Przygotowanie programów komputerowych korzystających z wybranych algorytmów wyszukiwania i sortowania, • Przygotowanie programów komputerowych oceniających złożoność i efektywność obliczeniowa wybranych algorytmów sortowania. • Przygotowanie programów komputerowych implementujących algorytmy silni i ciągu Fibonacciego przy wykorzystaniu technik rekurencyjnych i nie-rekurencyjnych, np. iteracyjnych, ze wskazaniem wad i zalet każdej z nich. • implementacja wybranych zadań implementujących web scraping, machine learning oraz text mining.		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.