


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 Projekt współfinansowany przez
 Unię Europejską w ramach
 Europejskiego Funduszu
 Społecznego

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Nazwa przedmiotu		Kod ECTS	
Wstęp do programowania		11.3.1990	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	stacjonarne
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
prof. UG, dr Jakub Neumann; mgr Radosław Ziemann; dr inż. Anna Nenca; mgr Grzegorz Madejski; mgr Konrad Sołtys; dr Maciej Dziemiańczuk; mgr Michał Zakrzewski			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		7	
Wykład, Ćw. laboratoryjne		Udział w zajęciach dydaktycznych, objętych planem studiów: 60h	
Sposób realizacji zajęć		Praca własna studenta: 115h	
zajęcia w sali dydaktycznej		RAZEM: 175h	
Liczba godzin			
Wykład: 15 godz., Ćw. laboratoryjne: 45 godz.			
Termin realizacji przedmiotu			
2023/2024 zimowy			
Status przedmiotu		Język wykładowy	
obowiązkowy		polski	
Metody dydaktyczne		Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne	
<ul style="list-style-type: none"> - Projektowanie doświadczeń - Rozwiązywanie zadań - Wykład problemowy - Wykład z prezentacją multimedialną - ćwiczenia laboratoryjne -- sporządzanie i uruchamianie programów komputerowych 		Sposób zaliczenia	
		Zaliczenie na ocenę	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - wykonanie pracy zaliczeniowej - projekt lub prezentacja - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru - kolokwium 	
		Podstawowe kryteria oceny	
		próg zaliczeniowy	składowa oceny końcowej
	kolokwium	60%	60%
	wejściówki	50%	20%
	projekt	50%	20%
Sposób weryfikacji założonych efektów uczenia się			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność	obserwacja postawy i umiejętności
	Wiedza						
K_W04		X					
	Umiejętności						
K_U04		X	X				
K_U03			X				X
	Kompetencje						
K_K02							X

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi**A. Wymagania formalne**

Brak wymagań formalnych

B. Wymagania wstępne

Brak wymagań wstępnych

Cele kształcenia

Celem przedmiotu jest poznanie podstawowych konstrukcji i technik programistycznych, struktur danych oraz zdobycie umiejętności projektowania, analizy i implementacji podstawowych algorytmów.

Treści programowe

- Programy w różnych sposobach zapisu (opis słowny, instrukcje języka programowania). Ręczna symulacja działania algorytmu.
- Środowisko programistyczne, uruchamianie i debuggowanie programów
- Zmienne i podstawowe typy danych. Zakres deklaracji i widoczność zmiennych, zmienne lokalne i globalne
- Podstawowe sterujące konstrukcje programistyczne: instrukcje warunkowe, pętle (w tym zagnieżdżone)
- Funkcje, parametry, wartość zwracana, czyste funkcje
- Podstawowe struktury danych: listy, słowniki
- Obsługa błędów

Wykaz literatury

A. Literatura wymagana do ostatecznego zaliczenia zajęć (zdania egzaminu):

A.1. wykorzystywana podczas zajęć

A.2. studiowana samodzielnie przez studenta

B. Literatura uzupełniająca

Kierunkowe efekty uczenia się

K_W04 ma uporządkowaną, podbudowaną teoretycznie wiedzę w zakresie programowania, algorytmów i złożoności, języków i paradygmatów programowania
K_U04 potrafi tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz wzorców projektowych
K_U03 potrafi pracować w zespole informatyków, zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminy, porozumiewać się przy użyciu różnych technik w środowisku zawodowym w tym z wykorzystaniem dedykowanych narzędzi
K_K02 potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania

Wiedza

Student zna podstawowe konstrukcje programistyczne i struktury danych.

Umiejętności

Student potrafi zaprojektować, zaimplementować, uruchomić i analizować proste programy przy użyciu podstawowych struktur danych. Potrafi zaimplementować prosty algorytm posługując się podstawowymi konstrukcjami programistycznymi. Potrafi podejmować zobowiązania i dotrzymywać terminy realizacji zadań programistycznych

Kompetencje społeczne (postawy)

potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania

Kontakt

Jakub.Neumann@ug.edu.pl