

**KAPITAŁ LUDZKI**
NARODOWA STRATEGIA SPÓJNOŚCIProjekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego**UNIA EUROPEJSKA**
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Nazwa przedmiotu		Kod ECTS	
Podstawy programowania		11.1.0526	
Nazwa jednostki prowadzącej przedmiot			
Instytut Matematyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Matematyka	forma	stacjonarne
		moduł	matematyka nauczycielska, matematyka ogólna
		specjalnościowy	
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Monika Wrzosek; dr Adam Dzedzej; dr Rafał Lutowski; dr Milena Matusik; dr Iwona Krzyżanowska; dr Maciej Niebrzydowski; dr Marek Hałenda			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		5	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Wykład: 15 godz., Ćw. laboratoryjne: 45 godz.			
Termin realizacji przedmiotu			
2020/2021 letni			
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"> - Wykład problemowy - Wykład z prezentacją multimedialną - ćwiczenia laboratoryjne - rozwiązywanie zadań programistycznych 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - egzamin pisemny testowy - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru - Projekty programistyczne w grupach 2-3 osobowych. - kolokwium 	
		Podstawowe kryteria oceny	
Sposób weryfikacji założonych efektów kształcenia			

zakładany efekt kształcenia	Egzamin	Projekt	Obserwacja postawy studenta
	Wiedza		
M_W10	+		
M_W12			+
	Umiejętności		
M_U09	+		
M_U10		+	
M_U11		+	
M_U12		+	
M_U13		+	
	Kompetencje		
M_K03		+	

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

Brak

B. Wymagania wstępne

Znajomość podstawowych pojęć logiki matematycznej i teorii mnogości.

Cele kształcenia

Przygotowanie studentów do samodzielnego konstruowania algorytmów dla rozwiązywania nowych problemów w oparciu o podstawowe struktury danych i elementy języka programowania: funkcje, tablice, iteracje i rekurencję. Wykształcenie umiejętności kontroli i weryfikacji własnego i cudzego rozumowania i kodu źródłowego. Przygotowanie do pogłębionych poszukiwań intelektualnych, literaturowych i technicznych wobec problemów informatycznych, dla których proste narzędzia nie wystarczają.

Treści programoweProblematyka wykładu i ćwiczeń laboratoryjnych:

Elementy języka C/C++: struktura programu, pojęcie zmiennych i stałych oraz dostępne dla nich typy danych, operatory i ich hierarchia, instrukcje warunkowe, pętle for i while, konstrukcja i użycie tablic, obsługa łańcuchów znaków oraz standardowych strumieni danych, pojęcie wskaźnika; elementy biblioteki standardowej języka C. Organizacja komputera.

Problematyka ćwiczeń audytoryjnych:

Schematy blokowe: algorytmy badające własności geometryczne, naiwne algorytmy teorii liczb, algorytmy dokonujące konwersji między systemami n-narnymi, przeszukujące tablice, realizujące proste wejście/wyjście znakowe, realizujące proste przeszukiwanie tekstów.

Podstawowe algorytmy rekurencyjne: przeszukiwanie drzew.

Wykaz literatury

Literatura wykorzystywana podczas zajęć oraz studiowana samodzielnie przez studenta:

- Język ANSI C, Brian W. Kernighan, Dennis M. Ritchie, WNT Warszawa 2000.
- Język ANSI C. Programowanie. Wydanie II, Brian W. Kernighan, Dennis M. Ritchie, Wydawnictwo Helion, Gliwice 2010.
- Język C++, Bjarne Stroustrup, WNT Warszawa 2002.

Literatura uzupełniająca:

- Wprowadzenie do algorytmów, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, PWN Warszawa 2012.

Kierunkowe efekty kształcenia

Zaznajomienie z podstawami języka C/C++ i z niskopoziomym przechowywaniem i przetwarzaniem informacji, a także z pracą programisty w konsoli Windows i Linux. Świadomość ograniczeń języka oraz podatności tworzonego kodu na błędy. Umiejętność współpracy przy rozwiązywaniu zadań, przy zachowaniu odpowiedzialnego podejścia do podziału pracy.

Wiedza

Student, który zaliczył przedmiot, zna:

- niektóre klasy problemów matematycznych, których rozwiązań można poszukiwać, przekładając je na język programowania wysokiego poziomu,
- podstawowe konstrukcje, reguły i polecenia języka C/C++,
- model funkcyjny przetwarzania danych (czarna skrzynka);
- zna podstawowe zasady bezpieczeństwa i higieny pracy

jak również: rozróżnia algorytmy iteracyjne i rekurencyjne oraz odczytuje algorytm na podstawie schematu blokowego. Ma też świadomość różnorodności zagadnień dyskretnych. M_W10, M_W12

Umiejętności

Student, który zaliczył przedmiot, umie:

- rozwiązywać proste problemy dyskretne, zarówno znane, jak i nowe, począwszy od opracowania algorytmu w luźnej postaci, poprzez jego

	<p>formalizację np. w postaci schematu blokowego, aż do stworzenia działającego programu;</p> <ul style="list-style-type: none"> • doprecyzować, przedyskutować i przełożyć własne wyobrażenie o algorytmie na schemat blokowy; • na podstawie danego schematu blokowego, napisać program; w szczególności, wie jak przełożyć rozgałęzienia drzewa algorytmu na instrukcje warunkowe; • ocenić złożoność obliczeniową zadania oraz jego realizacji programistycznej; • opracowywać algorytmy w grupie, ale także samodzielnie pisać programy w domu i na zajęciach; • współpracować przy tworzeniu modularnego kodu w języku C/C++ • ocenić poziom trudności implementacji danego zagadnienia; • stworzyć prosty interfejs programu, stosując podstawowe procedury wejścia/wyjścia (klawiatura/ekran); • znajdować i wyjaśniać błędy we własnych i cudzych algorytmach, a także w napisanych przez siebie programach; • poruszać się wprawnie w tekstowym środowisku programistycznym w systemie Linux. <p>Ma znajomość techniki programowania, pozwalającą na rozpoczęcie pracy nad bardziej złożonymi zagadnieniami dyskretnymi oraz nad wybranymi zagadnieniami analizy i algebry. M_U09, M_U10, M_U11, M_U12, M_U13.</p>
	<p>Kompetencje społeczne (postawy)</p> <p>Student uczy się współpracować systematycznie z innymi przy rozwiązywaniu problemów matematycznych, algorytmicznych i programistycznych, przy zachowaniu odpowiedzialności za swoją część pracy i efektywnego jej podziału. Poznaje ograniczenia swoich umiejętności programistycznych i jest w stanie tropić błędy we własnym rozumowaniu i w kodzie źródłowym. Potrafi kreować pytania, które pomagają w wyborze drogi rozwiązania. Uczy się korzystać z dostępnej literatury drukowanej i elektronicznej, ale nie z gotowych odpowiedzi. Czuje potrzebę tworzenia oryginalnych rozwiązań. Umie rozwiązywać zadania zajmujące więcej czasu, niż jeden blok zajęciowy. M_K03.</p>
<p>Kontakt</p> <p>Monika.Wrzosek@mat.ug.edu.pl</p>	