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

Nazwa przedmiotu	Kod ECTS
Podstawy programowania	11.1.0342

Nazwa jednostki prowadzącej przedmiot

null

Studia

wydział	kierunek		
Wydział Matematyki, Fizyki i Informatyki	Matematyka	poziom	pierwszego stopnia
		forma	stacjonarne
		moduł specjalnościowy	matematyka nauczycielska, matematyka
		specjalizacja	wszystkie

Nazwisko osoby prowadzącej (osób prowadzących)

dr Monika Wrzosek; dr Rafał Lutowski; dr Maciej Niebrzydowski; dr Iwona Krzyżanowska; dr Marek Hałenda; dr Milena Matusik

Formy zajęć, sposób ich realizacji i przypisana im liczba godzin

Formy zajęć

Wykład, Ćw. audytoryjne, Ćw. laboratoryjne

Sposób realizacji zajęć

zajęcia w sali dydaktycznej

Liczba godzin

Ćw. audytoryjne: 15 godz., Wykład: 30 godz., Ćw. laboratoryjne: 30 godz.

Liczba punktów ECTS

5

• Bilans nakładu pracy przeciętnego studenta:
1. udział w wykładach: 15 * 2h = 30h
2. udział w ćwiczeniach: 15 * 1h = 15h
3. udział w laboratoriach: 15 * 2h = 30h
4. cotygodniowe przygotowanie do ćwiczeń i laboratorium: 15 * 1h = 15h
5. realizacja zadań projektowych: 30h
(obejmuje także instalację oprogramowania, opanowanie umiejętności wykorzystania go do realizacji projektu
oraz przygotowanie sprawozdania)
6. obrona sprawozdania z projektów: 3h
7. obecność na egzaminie: 2h
Łączny nakład pracy wynosi 125h, co odpowiada 5 punktom ECTS.
• nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich:
30 + 15 + 30 + 3 + 2 = 75h, co odpowiada 3 punktom ECTS
• nakład pracy związany z zajęciami o charakterze praktycznym:
15 + 30 + 30 = 75h, co odpowiada 3 punktom ECTS

Cykl dydaktyczny

2017/2018 letni

Status przedmiotu	Język wykładowy
obowiązkowy	polski

Metody dydaktyczne

Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne

Sposób zaliczenia

- Zaliczenie na ocenę
- Egzamin

Formy zaliczenia

- Rozwiązywanie zadań
- Wykład problemowy
- Wykład z prezentacją multimedialną
- zespołowe (2-3 osoby) projekty programistyczne, 2 lub 3 w semestrze, wymagające studium dodatkowej literatury oraz konsultacji
- ćwiczenia laboratoryjne - rozwiązywanie zadań programistycznych

- 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

Egzamin: test z treści programowych wykładu, oceniany procentowo wg przelicznika z obowiązującego ,,Regulaminu Studiów UG".
Ćw. laboratoryjne: 70-100%: 2-3 kolokwia, 0-30%: oceny cząstkowe wystawiane w czasie zajęć, przy uwzględnieniu: czasu wykonania zadań, samodzielności studenta, jakości wytworzonego kodu; punkty przeliczane j.w.
Ćw. audytoryjne: 40%: oceny cząstkowe wystawiane w czasie zajęć, 60%: 2-3 projekty oceniane wg: a) zakresu wyczerpania tematu,
b) poprawności merytorycznej, c) oryginalności zaproponowanych rozwiązań, d) jakości współpracy wewnątrz zespołu; punkty przeliczane j.w.

Sposób weryfikacji założonych efektów kształcenia

zakładany efekt kształcenia	Egzamin	Projekt	Obserwacja postawy studenta
	Wiedza		
K_W10	+		
K_W12			+
	Umiejętności		
K_U09	+		
K_U10		+	
K_U11		+	
K_U12		+	
K_U13		+	
	Kompetencje		
K_K03		+	

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne
Wstęp do matematyki

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 programowe

Problematyka wykładu i ćwiczeń labratoryjnych:
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 teorioliczbowe, 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.

Efekty kształcenia (obszarowe i kierunkowe)

Zaznajomienie z podstawami języka C/C++ i z niskopoziomowym 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. K_W10, K_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 formalizację np. w postaci schematu blokowego, aż do stworzenia działającego programu;
- 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.

Ma znajomość techniki programowania, pozwalającą na rozpoczęcie pracy nad bardziej złożonymi zagadnieniami dykretnymi oraz nad wybranymi zagadnieniami analizy i algebry. K_U09, K_U10, K_U11, K_U12, K_U13.

Kompetencje społeczne (postawy)

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. K_K03.

Kontakt

Monika.Wrzosek@mat.ug.edu.pl