



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Nazwa przedmiotu		Kod ECTS	
Programowanie		11.3.0341	
Nazwa jednostki prowadzącej przedmiot			
Faculty of Mathematics, Physics and Informatics			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Bioinformatyka	forma	stacjonarne
		moduł	Podstawowa
		specjalnościowy	Podstawowa
		specjalizacja	Podstawowa
Nazwisko osoby prowadzącej (osób prowadzących)			
dr hab. Piotr Gnaciński; dr Janusz Młodzianowski; prof. dr hab. Danuta Makowiec; prof. UG, dr hab. Marek Krośnicki			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		5 Przedmiot w wymiarze 30h wykładu i 60h ćwiczeń w laboratorium komputerowym + praca własna	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Wykład: 30 godz., Ćw. laboratoryjne: 60 godz.			
Cykl dydaktyczny			
2016/2017 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"> - praca własna - przygotowanie się do zaliczenia - ćwiczenia laboratoryjne w pracowni komputerowej, praca własna - realizacja projektów w domu 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Zaliczenie (zał) 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - zaliczenie na podstawie obecności, albo kolokwium - kolokwium - wykonanie pracy zaliczeniowej - wykonanie określonej pracy praktycznej 	
		Podstawowe kryteria oceny	
		biegłość w stosowaniu zasad programowania opanowanie zasad konstrukcji projektów programistycznych biegłość w opanowaniu języka C, projektowanie i implementowanie w języku C prostych funkcji realizujących obliczenia na zmiennych złożonych.	
Sposób weryfikacji założonych efektów kształcenia			

zakładany efekt kształcenia	Kolokwium	Wykonanie programów	mtd. dydakt 3	mtd. dydakt 4	mtd. dydakt 5	mtd. dydakt 6	mtd. dydakt 7	mtd. dydakt 8
Wiedza								
K_W03	+	+						
K_W04	+	+						
Umiejętności								
K_U01	+	+						
K_U05	+	+						
K_U06	+	+						
K_U07	+	+						

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Zaliczony Wstęp do informatyki z sem. 1

B. Wymagania wstępne

znajomość środowiska systemu operacyjnego (Windows lub Linux)

posiada umiejętność projektowania i implementowania prostych jednomodułowych programów

Cele kształcenia

Opanowanie podstaw teoretycznych tworzenia wydajnych programów komputerowych

Nabycie praktycznych umiejętności konstrukcji prostych projektów programistycznych w oparciu o język C.

Treści programowe

Budowanie programów w języku wyższego rzędu. Kompilacja i konsolidacja. Pliki źródłowe, nagłówkowe, obiektowe. Preprocesor i jego dyrektywy. Struktura programu. Podstawowe typy danych i ich zakresy. Rzutowania i konwersje. Stałe, zmienne i ich zasięg, literały. Operatory i priorytety. Instrukcje sterujące wykonaniem programu. Funkcje. Złożone typy danych: tablice, struktury, unie. Wskaźniki. Funkcje i wskaźnik. Dynamiczne zarządzanie pamięcią. Biblioteki – tworzenie własnych bibliotek statycznych i dynamicznych. Błędy w programowaniu. Elementy programowania obiektowego. Podstawowe pojęcia programowania obiektowego: obiekt, atrybuty obiektu, metody, komunikaty, interfejs, klasy, wystąpienia klas i hierarchia dziedziczenia.

Wykaz literatury

B.W. Kernighan, D. M. Ritch, Język ANSI C, WNT 2002.

Ch. Allison, Thinking in C, <http://mindview.net/CDs/ThinkingInC>

Stephen Prata, Język C. Szkoła programowania.

Kenneth A. Ross, Charles R.B. Wrigth, Matematyka dyskretna

Efekty kształcenia

(obszarowe i kierunkowe)

K_W03 ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie programowania, algorytmów i złożoności, języków i paradygmatów programowania, baz danych, inżynierii oprogramowania

K_W04 zna podstawowe konstrukcje programistyczne oraz pojęcia składni i semantyki języków programowania; zna podstawowe metody projektowania, analizowania i programowania algorytmów; zna podstawowe struktury danych i wykonywane na nich operacje

K_U01 potrafi zastosować wiedzę matematyczną i informatyczną do formułowania, analizowania i rozwiązywania prostych zadań związanych z bioinformatyką

K_U05 potrafi projektować wykorzystując podstawowe techniki algorytmiczne i struktury danych, analizować, pisać uruchamiać i testować programy w wybranym środowisku programistycznym

K_U06 projektuje, analizuje pod kątem poprawności i złożoności obliczeniowej oraz programuje algorytmy; wykorzystuje podstawowe techniki algorytmiczne i struktur danych

Wiedza

Student zna:

- reprezentacje różnych typów danych i ich ograniczenia
- zasady programowania z funkcjami
- rozumie semantykę wartości i referencji
- rozumie koncepcje podstawowych abstrakcyjnych typów danych: stos, kolejka, lista, drzewo, graf

Umiejętności

Student potrafi:

- zaprojektować prosty algorytm, ocenić jego efektywność, zaprogramować funkcję w języku C realizującą algorytm
- zarządzać prostymi zmiennymi: zakresem ich nazw oraz zarządzać dynamicznie pamięcią
- zaprojektować proste struktury (klasy) i zarządzać nimi
- konstruować proste wieloskładnikowe projekty programistyczne w języku C

Kompetencje społeczne (postawy)

K_U07 posługuje się przyjętymi formatami reprezentacji różnego rodzaju danych stosownie do sytuacji, pamiętając o ich ograniczeniach	
--	--

Kontakt

fizpg@univ.gda.pl
