

KARTA KURSU

Nazwa	Wstęp do programowania
Nazwa w j. ang.	Introduction to Computer Programming

Kod		Punktacja ECTS*	3
-----	--	-----------------	---

Koordinator	prof. dr hab. Jacek Migdałek	Zespół dydaktyczny: prof. dr hab. Jacek Migdałek dr Olaf Bar
-------------	------------------------------	--

Opis kursu (cele kształcenia)

Celem realizacji kursu jest zapoznanie studentów kierunku Bioinformatyka z podstawami programowania: kodowaniem informacji, językami formalnymi, problemami translacji, podziałem języków programowania i rodzajami programowania oraz elementami algorytmiki.

W ramach ćwiczeń student pozna podstawy programowania w języku Python, pozna wybrane środowisko programistyczne (Anaconda, VirtualEnv). Będzie wykorzystywał podstawowe konstrukcje algorytmiczne (pętle i warunki) oraz podstawowe struktury danych (zmienne proste, listy, słowniki). Zapozna się także z pewnymi aspektami programowania obiektowego. Po zakończeniu kursu student będzie umiał omawiać i analizować programy w tym języku oraz posługiwać się modułami języka realizującymi wybrane funkcjonalności. Program ćwiczeń będzie w szczególności ukierunkowany na analizę danych (np. pomiarowych) oraz analizę statystyczną z uwzględnieniem specyfiki nauk przyrodniczych.

Kurs prowadzony jest w języku polskim.

Warunki wstępne

Wiedza	Podstawowe wiadomości z matematyki i informatyki na poziomie szkoły średniej.
Umiejętności	Uruchamianie komputera, obsługa systemu operacyjnego Windows
Kursy	Wstępne kursy nie są wymagane.

Efekty kształcenia

	Efekt kształcenia dla kursu	Odniesienie do efektów kierunkowych
Wiedza	Po zakończeniu kursu student:	
	W01: zna pojęcie algorytmu, jego strukturę, cechy, sposoby zapisu, metody badania poprawności algorytmów, optymalizację złożoności czasowej i pamięciowej algorytmów, wskaże problemy obliczalności i rozstrzygalności algorytmów	K_W09
	W02: scharakteryzuje i sklasyfikuje języki formalne, rozpoznaje problemy translacji języków formalnych i naturalnych, zna paradygmaty programowania i ich podział języków programowania, omawia zasady programowania strukturalnego i obiektowego	K_W10, K_W03
	W03: zna możliwości wykorzystania osiągnięć informatyki w biologii	K_W03

	Efekt kształcenia dla kursu	Odniesienie do efektów kierunkowych
Umiejętności	U01: umie tworzyć proste algorytmy; analizuje poprawność częściową i całkowitą prostego algorytmu; ocenia złożoność czasową i pamięciową przykładowych algorytmów i dokona ich optymalizacji.	K_U06
	U02: projektuje i tworzy programy komputerowe zgodnie z zadaną specyfikacją, używając właściwych metod, technik i narzędzi, umie pisać i omawiać proste programy w języku Python	K_U13
	U03 Tworzy proste programy wspomagające analizę danych pomiarowych	K_U06
	U04 Umie wykorzystać moduły języka wspomagające analizę danych	K_U06

	Efekt kształcenia dla kursu	Odniesienie do efektów kierunkowych
Kompetencje społeczne	Po zakończeniu kursu student:	
	K01: jest świadomy konieczności stałego uaktualniania wiedzy z zakresu informatyki, posiada zdolność do weryfikowania pozyskiwanych informacji, a także analizy podstawowych problemów informatyki i jej zastosowań. K02: jest świadomy konieczności dzielenia się wiedzą informatyczną w sposób zrozumiały dla innych, współdziała i pracuje w grupie	K_K01 K_K02

Studia stacjonarne

Organizacja											
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	15					30					

Studia niestacjonarne

Forma zajęć	Organizacja										
	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin											

Opis metod prowadzenia zajęć

Omawianie zagadnień teoretycznych, prezentacje multimedialne, omawianie, rozwiązywanie zadań (ćwiczenia), wykonywanie projektów, referowanie zagadnień z użyciem środków multimedialnych.

Formy sprawdzania efektów kształcenia

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Inne
W01	X				X	X		X	X				X
W02	X				X	X		X	X				X
W03									X				X
U01					X	X	X						X
U02					X	X	X						X
U03					X	X							
K01							X	X					
K02							X	X					

Kryteria oceny	Ocenę dobrą lub bardzo dobrą może uzyskać student, który: - bierze udział w dyskusji i wykazuje zaangażowanie - w czasie ćwiczeń: - prawidłowo dobiera poznane struktury danych do rozwiązywanego problemu, - zna i umie zaimplementować wybrane moduły - samodzielnie projektuje program realizujący zadane funkcjonalności

Uwagi	
-------	--

Treści merytoryczne (wykaz tematów)

Wykład:

- 1) Podstawy asemantycznej teorii informacji i teorii kodowania. Systemy kodowania binarnego, oktalnego i heksadecymalnego liczb. Konwersje. Arytmetyka binarna.
- 2) Języki i gramatyki formalne: Hierarchia Chomsky'ego a klasyfikacja języków formalnych. Problemy translacji
- 3) Język programowania: alfabet, składnia, semantyka. Klasyfikacja języków programowania: języki maszynowe, symboliczne, strukturalne (proceduralne), obiektowe. Metody programowania. Generacje języków programowania. Obszary zastosowań języków programowania.
- 4) Elementy algorytmiki:
 - zadanie algorytmiczne, struktura algorytmów;
 - poprawność częściowa i całkowita algorytmów, metody dowodzenia (metoda Floyda i metoda zbieżnika) oraz przykłady zastosowań, współczesne tendencje w badaniu poprawności algorytmów, poprawność algorytmów a dowodzenie twierdzeń matematycznych (na przykładzie tw. o czterech barwach i tw. Keplera o upakowaniu kul)
 - złożoność czasowa i pamięciowa algorytmów, typy złożoności czasowej, optymalizacja algorytmów, problemy zamknięte i luka algorytmiczna, ograniczenia górne i dolne na złożoność czasową algorytmów, złożoność czasowa a złożoność pamięciowa
 - problemy obliczalności i rozstrzygalności algorytmów, przykłady algorytmów, problemy P, NP i NP- zupełne, teza Churcha-Turinga-Markowa i jej konsekwencje.
- 5) Maszyna Cyfrowa von Neumanna jako przykład programowania bezpośredniego.
- 6) Elementy architektury komputera: twierdzenie Posta. Logika maszyny cyfrowej a algebra Boole'a. Bramki i układy logiczne. Budowa sumatora, koder a i dekoder a
- 7) Bioinformatyka i biokomputery

Ćwiczenia

- środowisko Anaconda
- zmienne i instrukcje
- iteracja, pojęcie iteratora, wstęp do list
- warunkowe instrukcje sterujące
- Python jako język obiektowy – pojęcie obiektu i metody
- funkcje
- moduły – instrukcja import, przegląd wybranych modułów
- operacje na plikach tekstowych
- podstawy analizy danych w Pythonie

Wykaz literatury podstawowej

Wskazane rozdziały publikacji:

1. N. Abramson „Teoria informacji i kodowania” PWN, Warszawa 1969.
2. N. Wirth „Wstęp do programowania systematycznego”, Wydawnictwa Naukowo - Techniczne 1999.
3. D. Harel „Rzecz o istocie informatyki – algorytmika ” WN-T, Warszawa 1992
4. Peter Norton, „Python : od podstaw”, Gliwice : Helion, 2006
5. Mark Lutz, David Ascher, „Python : wprowadzenie”, Gliwice :Helion, 2002.

Wykaz literatury uzupełniającej

1. W. Turski „Propedeutyka informatyki, PWN 1975
2. I. Pohl, A. Shaw „The nature of computations:Introduction to Computer Science” Comp. Sci. Press Inc. USA 1981
3. Strona internetowa przedmiotu: <http://ux.ap.krakow.pl/~tpi/>
4. Mark Summerfield „Python 3 : kompletne wprowadzenie do programowania”,Gliwice : Helion, cop. 2010.
5. Materiały dydaktyczne opracowane przez prowadzącego kurs w formie opisowej i prezentacji zamieszczane na platformie e-learningowej.

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) – **studia stacjonarne**

Liczba godzin w kontakcie z prowadzącymi	Wykład	15
	Konwersatorium (ćwiczenia, laboratorium itd.)	30
	Pozostałe godziny kontaktu studenta z prowadzącym	5
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	5
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	10
	Przygotowanie do egzaminu	25
Ogółem bilans czasu pracy		90
Liczba punktów ECTS w zależności od przyjętego przelicznika		3

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) – studia niestacjonarne

Liczba godzin w kontakcie z prowadzącymi	Wykład	
	Konwersatorium (ćwiczenia, laboratorium itd.)	
	Pozostałe godziny kontaktu studenta z prowadzącym	
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	
	Przygotowanie do egzaminu	
Ogółem bilans czasu pracy		
Liczba punktów ECTS w zależności od przyjętego przelicznika		