UNIWERSYTET GDAŃSKI

Nazwa przedmiotu

(1) Nazwa przedmiotu
Wstęp do programowania

(2) Kod ECTS

(3) Nazwa jednostki prowadzącej kierunek
Instytut Fizyki Doświadczalnej

(4) Studia

Nazwa studiów podyplomowych	Poziom	Forma
Podyplomowe Studia Podstaw Informatyki	Studia podyplomowe	Niestacjonarne

(5) Nazwisko osoby prowadzącej (osób prowadzących)
dr Tomasz Borzyszkowski

(6) Formy zajęć, sposób ich realizacji i przypisana im liczba godzin

A. Formy zajęć, zgodne z zarządzeniem Rektora UG
Wykład, Ćw. laboratoryjne

B. Sposób realizacji zajęć
zajęcia w sali dydaktycznej

C. Liczba godzin
Ćw. laboratoryjne: 30 godz., Wykład: 20 godz.

(7) Liczba punktów ECTS
7

(8) Termin realizacji przedmiotu

(9) Status przedmiotu
obowiązkowy

(10) Język wykładowy
polski

(11) Metody dydaktyczne
- wykład z prezentacją multimedialną
- ćwiczenia laboratoryjne: projektowanie doświadczeń
- ćwiczenia laboratoryjne: wykonywanie
- doświadczeń

(12) Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne

A. Sposób zaliczenia, zgodny z Regulaminem Studiów UG
- Egzamin
- Zaliczenie na ocenę

B. Formy zaliczenia
- ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru
- egzamin: prezentacja samodzielnie przygotowanego rozwiązania programistycznego

C. Podstawowe kryteria oceny lub wymagania egzaminacyjne
- Egzamin: spełnienie wymagań programistycznych zadania egzaminacyjnego oraz prezentacja szczegółów technicznych jego rozwiązania
- Ćwiczenia: oceny uzyskiwane za projektowane i uruchamiane programy

D. Sposób weryfikacji założonych efektów uczenia się w ramach danego przedmiotu

(13) Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne - brak

B. Wymagania wstępne - brak

(14) Cele kształcenia

Nauczenie studentów projektowania, tworzenia i uruchamiania prostych programów w wybranym środowisku programistycznym. Wyrobienie właściwych nawyków programistycznych oraz algorytmicznego podejścia do rozwiązywania problemów.

(15) Treści programowe

- Istota, ewolucja oraz klasyfikacja komputerów, algorytmów, a także języków programowania. Klasyfikacja algorytmów oraz języków programowania.
- Wyrażanie algorytmów w języku diagramów blokowych
- Język Python:
  - Środowisko programistyczne i uruchamianie aplikacji.
  - Podstawowe struktury danych
  - Przegląd instrukcji języka
  - Definiowanie funkcji i modułów
- Korzystanie ze źródeł danych
  - Obsługa plików tekstowych
  - Obsługa baz danych
- Przetwarzanie danych:
  - Statystyczne przetwarzanie danych
  - Wizualizacja wyników algorytmów statystycznych

(16) Wykaz literatury

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

A.1. wykorzystywana podczas zajęć

- Python For Beginners, https://www.python.org/about/gettingstarted/, Dostęp: 1 września 2020.
- Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman: Algorytmy i struktury danych. Wyd. Helion, Gliwice 2019.

A.2. studiowana samodzielnie przez studenta

- Tutorial programowania w Pythonie. https://pythonprogramming.net/, Dostęp: 1 września 2020.
- Piotr Wróblewski: Algorytmy, struktury danych i techniki programowania. Wydanie IV, Helion, Gliwice 2018.

B. Literatura uzupełniająca

- M. M. Sysło, Algorytmy. Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1997.
- W. M. Turski, Propedeutyka informatyki. Państwowe Wydawnictwa Naukowe, Warszawa, 1989.
- N. Wirth, Algorytmy + struktury danych = programy. Wydawnictwa Naukowo-Techniczne, Warszawa, 1989.

(17) Kierunkowe efekty uczenia się	**(17 A) Wiedza**
	• Student zna zasób podstawowych konstrukcji algorytmicznych (warunki, pętle, wywołanie podprogramu) i jest w stanie wyrazić je w wybranym języku programowania. • Ma świadomość znaczenia pojęcia poprawności programu
	(17) Umiejętności • Student umie posłużyć się podstawowymi narzędziami służącymi do tworzenia, uruchamiania i diagnostyki programów w wybranym języku (edytor, kompilator, narzędzia do śledzenia wykonywania programu). • Umie przeprowadzać proste rozumowania prowadzące do wykrycia błędów w programach.

	(17) Kompetencje społeczne (postawy) • Student umie współpracować z zespołem programistów i wspólnie rozwiązywać napotkane problemy. • Potrafi dzielić się wiedzą z innymi studentami i korzystać z ich wiedzy.
(18) Kontakt t.borzyszkowski@ug.edu.pl	