


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 Projekt współfinansowany przez
 Unię Europejską w ramach
 Europejskiego Funduszu
 Społecznego

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Nazwa przedmiotu		Kod ECTS	
Zaawansowane języki skryptowe		11.3.1505	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	niestacjonarne (zaoczne)
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Tomasz Borzyszkowski; dr Robert Fidytek; mgr Michał Kassjański; dr inż. Arkadiusz Mirakowski			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		8 Przedmiot w wymiarze 30h wykładu i 30h lab. + praca własna studenta	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. laboratoryjne: 30 godz., Wykład: 30 godz.			
Termin realizacji przedmiotu			
2022/2023 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"> - Projektowanie doświadczeń - Wykonywanie doświadczeń - Wykład z prezentacją multimedialną 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - wykonanie pracy zaliczeniowej - projekt lub prezentacja - egzamin pisemny z pytaniami (zadaniami) otwartymi - kolokwium 	
		Podstawowe kryteria oceny	
		Laboratorium: <ul style="list-style-type: none"> • 50% kolokwium • 50% punkty z projektu programistycznego Egzamin pisemny	
Sposób weryfikacji założonych efektów uczenia się			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	sprawdzian	referat	raport	aktywność w dyskusji	obserwacja postawy
Wiedza								
K_W05	X	X						
P_W1	X	X						
Umiejętności								
K_U07		X						X
K_U10								X
P_U1		X						X
Kompetencje społeczne								
K_K01								X
K_K03							X	X
P_K1								X
P_K2							X	X

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

Zaliczenie z przedmiotów: Języki programowania i Środowisko programistyczne.

Cele kształcenia

Nauczenie studentów podstawowych konstrukcji programistycznych dostępnych w językach skryptowych na przykładzie języka Python.

Treści programowe

- Podstawowe koncepcje:** kompilatory i interpretery, podstawowe typy danych, dynamiczne typy danych.
- Funkcje:** konstrukcja funkcji w Pythonie, dokumentowanie funkcji, filtrowanie i lambda-funkcje.
- Moduły i pakiety:** koncepcja przestrzeni nazw, definicja modułu oraz sposoby jego importowania, tworzenie pakietów.
- Klasy i obiekty:** definicja klas i tworzenie instancji, dziedziczenie i atrybuty, metody specjalne.
- Pliki i wyjątki:** podstawowe operacje na plikach, moduł pickle, instrukcja try i jej sekcje, tworzenie i podnoszenie wyjątków.
- Wyrażenia regularne:** podstawowe konstrukcje, kompilacja wyrażeń, grupy i podgrupy.
- Przetwarzanie języka HTML:** biblioteka sgmlib.py, funkcje locals() i globals(), przykład Dialectizer.
- Moduł unittest:** testowanie poprawności, błędów i zdrowości programu, unit tests a poprawianie programu.

Wykaz literatury

- Guido van Rossum, Python Tutorial, <http://docs.python.org/tut/>.
- Mark Pilgrim, Dive into Python. <http://diveintopython.org/>.
- Bruce Eckel, Thinking in Python, <http://www.mindview.net/Books/TIPython>.
- Python's official documentation, <http://docs.python.org/>.

Kierunkowe efekty uczenia się

K_W05: ma ogólną wiedzę na temat różnych paradygmatów programowania i języków programowania; szczególnie zna metody i wzorce projektowania i programowania obiektowego

K_U07: potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców

K_U10: potrafi oceniać przydatność paradygmatów i narzędzi programistycznych do rozwiązywania problemów różnego typu

K_K01: zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego uczenia się

K_K03: potrafi i jest gotów formułować opinie na temat podstawowych zagadnień informatycznych

Wiedza

P_W1: Student rozumie zasady działania skryptowych języków programowania i wie jak działają w tych językach podstawowe konstrukcje programistyczne (K_W05)

Umiejętności

P_U1: Student umie zastosować wybrane biblioteki języka Python do rozwiązania prostych problemów programistycznych (K_U07, K_U10)

Kompetencje społeczne (postawy)

P_K1: Student zdaje sobie sprawę z tempa rozwoju technologicznego i rozumie potrzebę dalszego uczenia się (K_K01)

P_K2: Student potrafi formułować opinie na temat zastosowania podstawowych konstrukcji programistycznych (K_K03)

Kontakt

t.borzyszkowski@ug.edu.pl