

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

<b>Nazwa przedmiotu</b>		<b>Kod ECTS</b>	
Zaawansowane języki skryptowe		11.3.1505	
<b>Nazwa jednostki prowadzącej przedmiot</b>			
Instytut Informatyki			
<b>Studia</b>			
<b>wydział</b>	<b>kierunek</b>	<b>poziom</b>	<b>pierwszego stopnia</b>
Wydział Matematyki, Fizyki i Informatyki	Informatyka	<b>forma</b>	niestacjonarne (zaoczne)
		<b>moduł</b>	wszystkie
		<b>specjalnościowy</b>	wszystkie
		<b>specjalizacja</b>	wszystkie
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>			
dr Tomasz Borzyszkowski; mgr Michał Kassjański; dr inż. Arkadiusz Mirakowski; dr Robert Fidytek			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		8 Przedmiot w wymiarze 30h wykładu i 30h lab. + praca własna studenta	
Wykład, Ćw. laboratoryjne			
<b>Sposób realizacji zajęć</b>			
zajęcia w sali dydaktycznej			
<b>Liczba godzin</b>			
Wykład: 30 godz., Ćw. laboratoryjne: 30 godz.			
<b>Termin realizacji przedmiotu</b>			
2021/2022 letni			
<b>Status przedmiotu</b>		<b>Język wykładowy</b>	
obowiązkowy		polski	
<b>Metody dydaktyczne</b>		<b>Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne</b>	
<ul style="list-style-type: none"> <li>- Projektowanie doświadczeń</li> <li>- Wykonywanie doświadczeń</li> <li>- Wykład z prezentacją multimedialną</li> </ul>		<b>Sposób zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- Zaliczenie na ocenę</li> <li>- Egzamin</li> </ul>	
		<b>Formy zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- wykonanie pracy zaliczeniowej - projekt lub prezentacja</li> <li>- egzamin pisemny z pytaniami (zadaniami) otwartymi</li> <li>- kolokwium</li> </ul>	
		<b>Podstawowe kryteria oceny</b>	
		Laboratorium: <ul style="list-style-type: none"> <li>• 50% kolokwium</li> <li>• 50% punkty z projektu programistycznego</li> </ul> Egzamin pisemny	
<b>Sposób weryfikacji założonych efektów uczenia się</b>			

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