

## KARTA KURSU

Nazwa	Programowanie II	
Nazwa w j. ang.	<i>Programming II</i>	
Koordynator	dr Roman Czapla	Zespół dydaktyczny
		dr Roman Czapla mgr Michał Frontczak
Punktacja ECTS*	2	

## Opis kursu (cele kształcenia)

Celem kursu jest nauka podstaw programowania obiektowego oraz poznanie technik średnio zaawansowanego programowania z zastosowaniem języka Python 3.x. Kurs prowadzony jest w języku polskim.

## Warunki wstępne

Wiedza	Student zna podstawy składni, podstawowe typy danych oraz struktury kontrolne i funkcje języka Python 3.x.
Umiejętności	Student potrafi pisać proste programy użytkowe w języku Python 3.x stosując paradygmat strukturalny i proceduralny.
Kursy	Programowanie I

## Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Wiedza	Po zakończeniu kursu student: <b>W01:</b> zna podstawowe pojęcia związane z programowaniem obiektywnym: klasa, obiekt, dziedziczenie, polimorfizm, itp.	K_W04
	<b>W02:</b> zna techniki programowania zorientowanego obiektowego;	K_W04
	<b>W02:</b> posiada wiedzę na temat wybranych zaawansowanych elementów programowania w języku Python;	K_W04
	<b>W04:</b> zna wybrane biblioteki pozwalające tworzyć interfejsy graficzne i proste gry.	K_W04
Umiejętności	Po zakończeniu kursu student: <b>U01:</b> potrafi projektować oraz implementować własne klasy różnorodnych obiektów i tworzyć ich instancje;	K_U03
	<b>U02:</b> projektuje i tworzy kompletne programy z wykorzystaniem metodologii obiektowej;	K_U03
	<b>U03:</b> wykorzystuje zaawansowane elementy języka Python w swoich programach;	K_U03
	<b>U04:</b> potrafi wykorzystać zdobyte umiejętności do projektowania interfejsów graficznych i/lub prostych gier 2D z wykorzystaniem odpowiednich bibliotek	K_U03
Kompetencje społeczne	Po zakończeniu kursu student: <b>K01:</b> potrafi korzystać z różnych źródeł informacji (w tym zasobów sieciowych) do poszerzania własnej wiedzy i zdobywania nowych umiejętności;	K_K02
	<b>K02:</b> wykazuje umiejętność stosowania w praktyce zdobytej wiedzy przedmiotowej i potrafi działać kreatywnie w celu rozwiązywania postawionych mu zadań;	K_K02
	<b>K03:</b> potrafi przekazywać wiedzę informatyczną w sposób zrozumiały dla innych i współpracować w zespole	K_K02

Organizacja										
Forma zajęć	Wykład (W)	Ćwiczenia w grupach								
		A		K		L		S		P
Liczba godzin	10					20				

### Opis metod prowadzenia zajęć

Kurs składa się z ćwiczeń prowadzonych w formie laboratoriów oraz wykładów.

W ramach wykładów studentom przekazywana jest podstawa wiedzy na temat konkretnych zagadnień dot. programowania obiektowego w języku Python wraz z przykładami. Materiały z wykładów są udostępniane studentom na platformie e-learningowej.

W ramach laboratoriów studenci projektują i tworzą zadane przez prowadzącego programy w języku Python, które następnie są omawiane.

### Formy sprawdzania efektów uczenia się

	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	
W02					X	X	X	X				X	
W03					X	X	X	X				X	
W04					X	X	X	X					
U01					X	X	X	X				X	
U02					X	X	X	X				X	
U03					X	X	X	X				X	
U04					X	X	X	X					
K01					X	X							
K02					X	X	X	X					
K03					X	X	X	X					

### Kryteria oceny

Ocena jest wystawiana na podstawie wykonanych przez studentów projektów programistycznych i/lub testu sprawdzającego wiedzę z zakresu omawianych treści merytorycznych. Ocenę dobrą i bardzo dobrą może uzyskać student, który:

- który wykazuje się dobrą lub bardzo dobrą znajomością składni i narzędzi języka Python 3.x pozwalających programować z wykorzystaniem metodologii obiektowej;
- potrafi wykorzystywać techniki programowania obiektowego do tworzenia wyspecjalizowanych klas i budowy odpowiednich ich hierarchii;
- potrafi dbać o jakość pisanych programów i wykorzystywać zaawansowane elementy języka Python;
- potrafi projektować i tworzyć interfejsy graficzne oraz/lub proste gry 2D.z wykorzystaniem metodologii obiektowej

## Treści merytoryczne (wykaz tematów)

1. Wprowadzenie do projektowania obiektowego w języku Python
  - własne klasy, atrybuty i metody, tworzenie instancji obiektów
  - pojęcie konstruktora/inicjalizatora i metody specjalne
  - atrybuty i metody klasy – atrybuty statyczne
  - prywatne i chronione atrybuty w języku Python
  - kontrola dostępu do atrybutów – właściwości
  - dziedziczenie i polimorfizm
  - rodzaje wyjątków, zastosowanie i podstawy działania
  - sposób przechwytywania wyjątków, tworzenie i zgłaszanie wyjątków
2. Programowanie zorientowane obiektowo w języku Python i wybrane zaawansowane elementy programowania w języku Python
  - iteratory i generatory, obiekty iterowalne
  - przeciążanie operatorów
  - tworzenie w pełni zintegrowanych typów danych
  - menadżery kontekstu
  - deskryptory, ich rodzaje i zastosowania
  - dekoratory funkcji i klas
  - obsługa plików tekstowych i binarnych, serializacja obiektów
  - moduły i pakiety
  - nowe elementy języka Python wprowadzane w kolejnych wersjach
3. Zastosowanie technik programowania obiektowego (jedna wybrana koncepcja)
  - tworzenie prostych gier 2D z wykorzystaniem biblioteki pygame
  - projektowanie interfejsów graficznych z użyciem np. biblioteki PyQt5

## Wykaz literatury podstawowej

## Wybrane fragmenty:

1. S. F. Lott, D. Phillips, Python Object-Oriented Programming: Build robust and maintainable object-oriented Python applications and libraries, 4th Edition, Packt Publishing 2021;
2. Al Sweigart, Programowanie w Pythonie dla średnio zaawansowanych. Najlepsze praktyki tworzenia czystego kodu, Helion, 2021;
3. E. Matthes, Python. Instrukcje dla programisty. Wydanie II, Helion, 2020;
4. B. Lubanovic, Python. Nowoczesne programowanie w prostych krokach. Wydanie II, Helion, 2020;
5. N. Ceder, Python. Szybko i prosto. Wydanie III, Helion, 2019;
6. T. Gaddis, Python dla zupełnie początkujących. Owoce programowania. Wydanie IV, Helion 2019.

## Wykaz literatury uzupełniającej

1. Ch. Mayer, Kod Pythona w jednym wierszu. Jak profesjonaliści piszą programy doskonałe, Helion 2021;
2. P. J. Deitel, H. Deitel, Python dla programistów. Big Data i AI. Studia przypadków, Helion, 2020;
3. D. Kopec, Klasyczne problemy informatyki w Pythonie, PWN, 2020;
4. M. Lutz, Python. Leksykon kieszonkowy. Wydanie V, Helion 2014;
5. M. Lutz, Python. Wprowadzenie. Wydanie IV, Helion 2010.

## Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta)

liczba godzin w kontakcie z prowadzącymi	Wykład	10
	Konwersatorium (ćwiczenia, laboratorium itd.)	20
	Pozostałe godziny kontaktu studenta z prowadzącym	
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	
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	10
	Przygotowanie do egzaminu/zaliczenia	5
Ogółem bilans czasu pracy		50
Liczba punktów ECTS w zależności od przyjętego przelicznika		2