

SYLABUS**DOTYCZY CYKLU KSZTAŁCENIA 2021-2025**

Rok akademicki 2022/2023

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>język skryptowy</i>
Kod przedmiotu	
Nazwa jednostki prowadzącej kierunek	<i>Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>informatyka</i>
Poziom studiów	<i>studia inżynierskie I-go stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr studiów	<i>rok II, semestr 4</i>
Rodzaj przedmiotu	<i>inżynierski przedmiot kształcenia kierunkowego</i>
Język wykładowy	<i>polski</i>
Koordynator	<i>dr inż. Michał Kępski</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Michał Kępski, mgr inż. Marcin Mrukowicz mgr inż. Dawid Kosior</i>

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt ECTS
4	15			15					2

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Znajomość zagadnień z przedmiotów Algorytmy i struktury danych, Programowanie obiektowe, Inżynieria oprogramowania.

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

3.1 Cele przedmiotu

C1	Zapoznanie studentów z językiem programowania wysokiego poziomu Python jako językiem skryptowym o szerokim zastosowaniu i rozbudowanej bibliotece standardowej.
C2	Wykonanie praktycznej aplikacji z wykorzystaniem języka skryptowego.

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
EK_01	Student zna składnię języka Python i struktury danych w tym języku, rozumie różnice między językami interpretowanymi a kompilowanymi	K_w07,
EK_02	Potrafi samodzielnie programować w języku Python, tworząc programy złożone z kilku funkcji, wykorzystujące obiekty różnych klas, edytujące pliki	K_U11, K_U12

3.3 Treści programowe

A. Problematyka wykładu

1. Wstęp do języka Python. Interpreter języka Python. Składnia języka. Typy i operatory. Instrukcje sterujące.
2. Składnia języka (ciąg dalszy). Definiowanie funkcji. Zmienna, obiekt, referencja. Przestrzeń nazw.
3. Moduły i pakiety. Obsługa wyjątków, asercje. Logowanie. Definiowanie klas – podstawy.
4. Klasy, obiekty, programowanie obiektowe.
5. Środowisko programisty Pythona. Zarządzanie pakietami i ścieżka wyszukiwania modułów. Wirtualne środowiska. Narzędzia do zarządzania instalacjami: <i>conda</i> , <i>pip</i> .
6. Narzędzia wbudowane – biblioteka standardowa języka Python. Typowe zadania w języku Python: manipulacja plikami, katalogami, wczytywanie danych. Prosta komunikacja sieciowa. Działania w internecie. Biblioteka Numpy.
7. Zagadnienia zaawansowane języka Python: dekoratory, zarządzanie atrybutami, generatory. Wyrażenia regularne.

B. Problematyka ćwiczeń laboratoryjnych

Interpreter języka Python. Typy danych i operatory.
Struktury danych. Listy, krotki, słowniki. Definiowanie funkcji.
Moduły i pakiety. Definiowanie funkcji (ciąg dalszy). Obsługa wyjątków.
Definiowanie własnych klas.
Operacje na plikach. Biblioteka standardowa języka Python.
Implementacja prostego programu w języku Python.
Komunikacja sieciowa z wykorzystaniem gniazd.

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną.

Laboratorium: rozwiązywanie zadań programistycznych.

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
EK_01	Test	wykład
EK_02	Kolokwium przy komputerze	laboratorium

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Wykład: zaliczenie bez oceny na podstawie wiedzy zweryfikowanej na kolokwium

Zajęcia laboratoryjne: zaliczenie z oceną - na podstawie kolokwium i aktywności podczas zajęć. Prowadzący na podstawie oceny aktywności podczas zajęć podwyższyć lub obniżyć końcową ocenę o 0.5 stopnia.

Na ocenę dostateczny – student zna składnię języka Python w stopniu średnim. Potrafi wskazać różnicę pomiędzy językiem kompilowanym a interpretowanym. Zna podstawowe typy danych i instrukcje sterujące. Student potrafi napisać prosty skrypt w języku Python i uruchomić go z linii poleceń. Potrafi poprawnie zaimportować elementy biblioteki standardowej. Potrafi definiować funkcje wykonujące zestaw instrukcji i zwracające wynik. Potrafi wywołać zdefiniowane funkcje. Potrafi korzystać z gotowych klas – tworzyć obiekty i wywoływać ich metody.

Na ocenę dobry – jw., a ponadto student zna składnię języka Python w stopniu dobrym, potrafi definiować własne klasy, poprawnie implementować zależności dziedziczenia.

Na ocenę bardzo dobry – jw., a ponadto student wykazuje się umiejętnością płynnej pracy z językiem Python, zna i umie stosować zagadnienia zaawansowane tego języka: dekoratory, zarządzanie atrybutami, generatory.

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	30
Inne z udziałem nauczyciela (udział w konsultacjach, egzaminie)	
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	25
SUMA GODZIN	55
SUMARYCZNA LICZBA PUNKTÓW ECTS	2

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa: 1. Lutz, M.: <i>Python. Wprowadzenie</i>. Wydanie V, 2020, Helion 2. Oficjalna dokumentacja języka Python: http://www.python.org/doc/ 3. Slajdy z wykładów (dostępne na githubie prowadzącego https://github.com/majk3l/JS2017)
Literatura uzupełniająca: 1. Barry, P.: <i>Python. Rusz głową!</i> Wydanie II, 2017, Helion

Akceptacja Kierownika Jednostki lub osoby upoważnionej