

SYLABUS
DOTYCZY CYKLU KSZTAŁCENIA 2023-2026
(skrajne daty)
Rok akademicki 2024/2025

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	Programowanie 1
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	Kolegium Nauk Przyrodniczych
Nazwa jednostki realizującej przedmiot	Kolegium Nauk Przyrodniczych Instytut Matematyki
Kierunek studiów	Matematyka
Poziom studiów	studia I stopnia
Profil	ogólnoakademicki
Forma studiów	stacjonarne
Rok i semestr/y studiów	rok II, semestr 3
Rodzaj przedmiotu	specjalnościowy
Język wykładowy	polski
Koordynator	dr inż. Michał Kępski
Imię i nazwisko osoby prowadzącej / osób prowadzących	

* -opcjonalnie, zgodnie z ustaleniami w Jednostce

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt. ECTS
3	15			45					5

1.2. Sposób realizacji zajęć

☒ zajęcia w formie tradycyjnej

☐ zajęcia realizowane z wykorzystaniem metod i technik kształcenia na odległość

1.3 Forma zaliczenia przedmiotu (z toku) (egzamin, zaliczenie z oceną, zaliczenie bez oceny)
zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

- | |
|---|
| <ul style="list-style-type: none">- Umiejętność logicznego myślenia.- Znajomość matematyki na poziomie szkoły ponadgimnazjalnej. |
|---|

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 o szerokim zastosowaniu i rozbudowanej bibliotece standardowej.
C2	Nabycie umiejętności programowania prostych skryptów
C3	Nabycie umiejętności wykorzystywania bibliotek języka Python

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych ¹
EK_01	Student zna składnię języka Python oraz środowisko programowania w tym języku.	K_Wo5, K_Wo7
EK_02	Potrafi posługiwać się językiem Python w zakresie pisania nieskomplikowanych programów. Potrafi posługiwać się biblioteką standardową języka Python i wykorzystywać ją we własnych programach.	K_U14; K_U22; K_Ko4; K_Ko5; K_Ko7

3.3 Treści programowe

A. Problematyka wykładu

Treści merytoryczne
1. Wstęp do języka Python. Struktura programu. Typy podstawowe i operatory.
2. Instrukcje sterujące. Zmienna, referencja, obiekt.
3. Funkcje. Klasy. Zasięg nazw.
4. Moduły i pakiety. Wyjątki. Asercje.
5. Narzędzia wbudowane – biblioteka standardowa języka Python. Pliki.
6. Typowe zadania w języku Python. Biblioteka NumPy.
7. Warsztat programisty: wzorce projektowe (wprowadzenie), system kontroli wersji (wprowadzenie, założenia, proste komendy)

B. Problematyka ćwiczeń laboratoryjnych

Treści merytoryczne
Wstęp do języka Python. Typy danych.
Sterowanie przepływem, wyjątki. Instrukcje sterujące.
Funkcje, moduły, pakiety.
Działania matematyczne, operacje na wektorach i macierzach. Numpy. Biblioteka Matplotlib.
Operacje na plikach.
Programowanie obiektowe.

¹ W przypadku ścieżki kształcenia prowadzącej do uzyskania kwalifikacji nauczycielskich uwzględnić również efekty uczenia się ze standardów kształcenia przygotowującego do wykonywania zawodu nauczyciela.

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną.

Ćwiczenia laboratoryjne: rozwiązywanie zadań programistycznych przy stanowisku komputerowym.

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	zaliczenie pisemne (test)	w
EK_02	kolokwium, ocena aktywności na zajęciach	ćw.

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Wykład : zaliczenie pisemne (test)

Warunkiem zaliczenia jest uzyskanie co najmniej 50% punktów. Ocena końcowa jest wówczas ustalana według skali:

poniżej 50% pkt. – niedostateczny,
[50 – 60%) pkt. – dostateczny,
[60 – 70%) pkt. – plus dostateczny,
[70 – 80%) pkt. – dobry,
[80 – 90%) pkt. – plus dobry,
[90 – 100%] pkt. – bardzo dobry.

Zajęcia laboratoryjne: kolokwium

Możliwość podwyższenia lub obniżenia oceny o pół stopnia za aktywności (lub brak aktywności) podczas zajęć. Kolokwium ma za zadanie sprawdzić praktyczną umiejętność w zakresie pisania programów, wg poniższych kryteriów:

Na ocenę „dostateczny”:

Student potrafi poprawnie stworzyć program rozwiązujący proste problemy wykorzystując odpowiednie struktury danych oraz konstrukcje programistyczne dostępne w języku Python. Student składnię języka Python w stopniu pozwalającym mu pisać poprawne, proste skrypty.

Na ocenę „dobry”:

Kryteria na ocenę „dostateczny”, a ponadto: student potrafi poprawnie stworzyć programy rozwiązujące średniej trudności problemy wykorzystując odpowiednie struktury danych oraz konstrukcje programistyczne dostępne w języku Python. Potrafi wykorzystać nieskomplikowane funkcje z biblioteki standardowej. Rozumie pojęcie obiektowości i potrafi definiować proste klasy.

Na ocenę „bardzo dobry”:

Kryteria na ocenę „dobry”, a ponadto: student potrafi poprawnie stworzyć programy rozwiązujące średnio-zaawansowane problemy wykorzystując odpowiednie struktury danych oraz konstrukcje programistyczne dostępne w języku C. Potrafi poprawnie zastosować dziedziczenie w języku

obiektywnym. Korzysta bez większych problemów (posiłkując się dokumentacją) z wybranej biblioteki języka Python (np. NumPy) i potrafi ją wykorzystać do tworzenia własnych skryptów.

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

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny z harmonogramu studiów	60
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	3
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	62
SUMA GODZIN	125
SUMARYCZNA LICZBA PUNKTÓW ECTS	5

* Należy uwzględnić, że 1 pkt ECTS odpowiada 25-30 godzin całkowitego nakładu pracy studenta.

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa:

1. Summerfield M.: *Python 3 : kompletne wprowadzenie do programowania*, Helion, 2010.
2. Sweigart A.: *Automatyzacja nudnych zadań z Pythonem : nauka programowania*, Helion, 2017.

Literatura uzupełniająca:

1. Lutz M.: *Python. Wprowadzenie*. Wydanie IV, Helion, 2010
2. Oficjalna dokumentacja języka Python:
<http://www.python.org/doc/>

Akceptacja Kierownika Jednostki lub osoby upoważnionej