

SYLABUS
DOTYCZY CYKLU KSZTAŁCENIA 2025-2027
(skrajne daty)
Rok akademicki 2026/2027

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	Programowanie 2
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	Wydział Nauk Ścisłych i Technicznych
Nazwa jednostki realizującej przedmiot	Wydział Nauk Ścisłych i Technicznych Instytut Informatyki
Kierunek studiów	Matematyka
Poziom studiów	studia drugiego stopnia
Profil	ogólnoakademicki
Forma studiów	stacjonarne
Rok i semestr/y studiów	rok I, semestr 2
Rodzaj przedmiotu	specjalnościowy
Język wykładowy	polski
Koordinator	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				30					3

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

Znajomość podstaw programowania, znajomość podstaw algebry liniowej.

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

3.1 Cele przedmiotu

C ₁	Zapoznanie studentów z językiem programowania wysokiego poziomu Python jako językiem o szerokim zastosowaniu i rozbudowanej bibliotece standardowej oraz z wybranym środowiskiem programistycznym (np. jupyter notebook)
C ₂	Wypracowanie u studentów umiejętności stosowania języka Python do rozwiązywania problemów praktycznych i wizualizacji wyników.

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. Zna biblioteki języka Python odpowiednie do rozwiązywania problemów związanych z procesem analizy danych.	K_Wo5, K_Wo6
EK_02	Student potrafi posługiwać się językiem Python w zakresie pisania nieskomplikowanych programów. Potrafi posługiwać się wybranymi funkcjami bibliotek języka Python i wykorzystywać je w problemach związanych z analizą danych. Samodzielnie szuka funkcji bibliotecznych odpowiednich do rozwiązywanego problemu.	K_Uo7, K_Uo8, K_U13
EK_03	Student jest gotowy do kreatywnego korzystania z możliwości języka Python w zakresie analizy danych; rozumie znaczenie starannego przetestowania programu przed jego użyciem w celach komercyjnych i odpowiedzialności za działanie napisanych programów.	K_Ko2, K_Ko4, K_Ko6

3.3 Treści programowe

A. Problematyka ćwiczeń laboratoryjnych

Treści merytoryczne
Interpreter języka Python. Typy danych i operatory. Składnia języka. Instrukcje sterujące.
Struktury danych. Listy, krotki, słowniki. Definiowanie funkcji. Zmienna, obiekt, referencja. Przestrzeń nazw.
Moduły i pakiety. Definiowanie funkcji (ciąg dalszy). Obsługa wyjątków.
Klasy, obiekty, programowanie obiektowe. Definiowanie własnych klas.
Środowisko jupyter notebook i biblioteka numpy.
Operacje na plikach, obsługa formatu csv. Biblioteka standardowa języka Python. Wybrane elementy bibliotek do analizy danych.

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

Tworzenie własnych skryptów do analizy danych i prezentacji wyników z wykorzystaniem języka Python, środowiska jupyter notebook oraz biblioteki matplotlib.

3.4 Metody dydaktyczne

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	Kolokwium przy komputerze	Laboratorium
EK_02	Kolokwium przy komputerze, obserwacja w trakcie zajęć	Laboratorium
EK_03	Obserwacja w trakcie zajęć	Laboratorium

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

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

Na ocenę dostateczny – student zna składnię języka Python w stopniu średnim. 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. Korzysta prawidłowo z biblioteki numpy i bibliotek do analizy danych.

Na ocenę bardzo dobry – jw., a ponadto student wykazuje się umiejętnością płynnej pracy z językiem Python, zna i umie stosować zagadnienia zaawansowane związane z biblioteką numpy (np. broadcasting).

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	30
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	

Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	45
SUMA GODZIN	75
SUMARYCZNA LICZBA PUNKTÓW ECTS	3

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

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	Nie dotyczy
zasady i formy odbywania praktyk	Nie dotyczy

7. LITERATURA

Literatura podstawowa: 1. Python 3: kompletne wprowadzenie do programowania / Mark Summerfield; [tł. z ang. Robert Górczyński]. - Wyd. 2. - Gliwice: Helion, cop. 2010. 2. OFICJALNA DOKUMENTACJA JĘZYKA PYTHON: HTTP://WWW.PYTHON.ORG/DOC/
Literatura uzupełniająca: 1. Mark Lutz, David Ascher: Python. Wprowadzenie, 2002, Helion 2. Mark Lutz: Python. Wprowadzenie. Wydanie IV, 2010, Helion

Akceptacja Kierownika Jednostki lub osoby upoważnionej