# SYLABUS

**DOTYCZY CYKLU KSZTAŁCENIA** *2023-2027*
Rok akademicki 2024/2025

## 1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

| | |
|---|---|
| Nazwa przedmiotu | *programowanie w języku Python* |
| Kod przedmiotu | |
| Nazwa jednostki prowadzącej kierunek | *Instytut Informatyki, Kolegium Nauk Przyrodniczych* |
| Nazwa jednostki realizującej przedmiot | *Instytut Informatyki, Kolegium Nauk Przyrodniczych* |
| Kierunek studiów | *informatyka* |
| Poziom studiów | *studia inżynierskie I-go stopnia* |
| Profil | *ogólnoakademicki* |
| Forma studiów | *stacjonarne* |
| Rok i semestr studiów | *rok II, semestry 3, 4* |
| Rodzaj przedmiotu | *przedmiot kierunkowy* |
| Język wykładowy | *polski* |
| Koordynator | *dr inż. Michał Kępski* |
| Imię i nazwisko osoby prowadzącej / osób prowadzących | *dr inż. Michał Kępski, mgr inż. Dawid Kosior, mgr inż. Marcin Mrukowicz* |

### 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 | | | 15 | | | | | 2 |
| 4 | | | | 30 | | | | | 2 |

### 1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

### 1.3 Forma zaliczenia przedmiotu (z toku)

semestr 3: zaliczenie z oceną
semestr 4: egzamin

## 2. WYMAGANIA WSTĘPNE

| |
|---|
| Znajomość zagadnień z przedmiotów algorytmy i struktury danych i programowanie obiektowe. |

## 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 i struktury danych języka Python oraz przeznaczenie wybranych bibliotek. Rozumie różnice między językami interpretowanymi a kompilowanymi. Zna mechanizmy programowania obiektowego w tym języku. | K_W05, K_W06 |
| EK_02 | Potrafi samodzielnie programować w języku Python, tworząc programy złożone z kilku funkcji, wykorzystujące obiekty różnych klas, edytujące pliki. Potrafi definiować własne klasy, korzystać z mechanizmów obiektowości, tworzyć kod obsługujący wystąpienia wyjątków. | K_U02 |
| EK_03 | Potrafi wykorzystywać wybrane zaawansowane zagadnienia języka Python. Potrafi wykorzystywać wirtualne środowiska i narzędzia do zarządzania pakietami. Potrafi napisać skrypty realizujące komunikację sieciową/działania w internecie. Potrafi korzystać z wybranych bibliotek do prezentacji danych i obliczeń. | K_U02 |

### 3.3 Treści programowe

A. Problematyka wykładu

| **Część 1 (semestr 3)** |
|---|
| 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. Przestrzenie 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: *conda, pip.* |
| 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*. Biblioteka *Scipy*. |
| 7. Zagadnienia zaawansowane. Dekoratory, zarządzanie atrybutami, generatory. Wyrażenia regularne. |

B. Problematyka ćwiczeń laboratoryjnych

| |
|---|
| **Część 1 (semestr 3)** |
| 1. Interpreter Pythona. Tworzenie skryptów. Typy danych i operatory. Parametry programu. |
| 2. Struktury danych. Listy, krotki, słowniki. Definiowanie funkcji. |
| 3. Definiowanie funkcji (ciąg dalszy). Moduły i pakiety. Obsługa wyjątków. |
| 4. Definiowanie własnych klas. |
| 5. Operacje na plikach. Biblioteka standardowa języka Python. |
| 6. Biblioteka standardowa języka Python – wybrane moduły. |
| 7. Testowanie kodu. Testy jednostkowe w Pythonie. |
| **Część 2 (semestr 4)** |
| 8. *Pip, conda, virtualenv*. Wirtualne środowiska i zarządzanie pakietami. |
| 9. Dekoratory, generatory. Zaawansowane konstrukcje języka Python. |
| 10. Zaawansowane zagadnienia programowania obiektowego w Pythonie. |
| 11. Serializacja i deserializacja obiektów. |
| 12. Komunikacja sieciowa z wykorzystaniem gniazd. |
| 13. Wybrane zadania w języku Pyhon. |
| 14. Podstawy graficznych interfejsów użytkownika w Pythonie. |
| 15. IPython i Jupyter notebook. |
| 16. *Numpy.* |
| 17. *Scipy* i *Matplotlib*. |
| 18. Biblioteka *Pandas* i *h5py*. |

### 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 zaliczeniowy, egzamin | wykład |
| EK_02 | Kolokwium przy komputerze, obserwacja w trakcie zajęć | laboratorium |
| EK_03 | Projekt programistyczny, obserwacja w trakcie zajęć | laboratorium |

### 4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

***Część 1***

***Wykład:***
Zaliczenie wykładu następuje na podstawie pozytywnie zweryfikowanego w trakcie kolokwium opanowania wiedzy (część kompetencji określonych w efekcie EK_01).

***Zajęcia laboratoryjne – efekty EK_01, EK_02:***

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 i potrafi się nią posługiwać w stopniu średnim. Zna i potrafi wykorzystać podstawowe typy danych i instrukcje sterujące. Student potrafi napisać prosty skrypt w języku Python i uruchomić go z linii poleceń. Zna i 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.

***Część 2***
***Wykład – efekt EK_01:***
Egzamin pisemny z dwóch semestrów: Aby otrzymać ocenę pozytywną, student musi odpowiedzieć na co najmniej 50% pytań dotyczących składni, struktur danych, konstrukcji programistycznych, w tym tych realizujących paradygmat programowania obiektowego, języka Python oraz wybranych bibliotek tego języka.
Skala ocen: 91 – 100% bardzo dobry (5.0); 81 – 90% plus dobry (4.5); 71 – 80% dobry (4.0); 61 – 70% plus dostateczny (3.5); 50 – 60% dostateczny (3.0); poniżej 50% niedostateczny (2.0).

***Zajęcia laboratoryjne – efekty EK_02, EK_03:***
Zaliczenie z oceną: na podstawie projektu programistycznego i aktywności podczas zajęć. Prowadzący na podstawie oceny aktywności podczas zajęć może podwyższyć lub obniżyć końcową ocenę o 0.5 stopnia.

Tematyka projektu jest podana przez prowadzącego zajęcia laboratoryjne i powiązana z zagadnieniami omawianymi na laboratoriach w części 2. przedmiotu.

Ocena projektu jest ustalana biorąc pod uwagę uzyskany zakres funkcjonalny, zgodność z ustaloną specyfikacją oraz jakość przygotowanej implementacji.

## 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 | 60 |
| Inne z udziałem nauczyciela<br>(udział w konsultacjach, egzaminie) | 3 |
| Godziny niekontaktowe – praca własna studenta<br>(przygotowanie do zajęć, egzaminu, napisanie referatu itp.) | 50 |
| SUMA GODZIN | 113 |
| **SUMARYCZNA LICZBA PUNKTÓW ECTS** | **4** |

## 6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

| | |
|---|---|
| wymiar godzinowy | - |
| zasady i formy odbywania praktyk | - |

## 7. LITERATURA

Literatura podstawowa:

1. Python 3: kompletne wprowadzenie do programowania / Mark Summerfield ; [tł. z ang. Robert Górczyński]. - Wyd. 2. - Gliwice: Helion, cop. 2010.

Literatura uzupełniająca:

1. Mark Lutz, David Ascher: Python. Wprowadzenie, 2002, Helion
2. Mark Lutz: Python. Wprowadzenie. Wydanie IV, 2010, Helion
3. Oficjalna dokumentacja języka Python: http://www.python.org/doc/