

SYLABUSDOTYCZY CYKLU KSZTAŁCENIA **2020-2024***(skrajne daty)*

Rok akademicki 2021/2022

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	Podstawy programowania
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	Kolegium Nauk Przyrodniczych
Nazwa jednostki realizującej przedmiot	Kolegium Nauk Przyrodniczych
Kierunek studiów	Inżynieria materiałowa
Poziom studiów	studia pierwszego stopnia
Profil	ogólnoakademicki
Forma studiów	stacjonarne
Rok i semestr/y studiów	II rok, 3 semestr
Rodzaj przedmiotu	kierunkowy
Język wykładowy	polski
Koordynator	dr hab. Paweł Jakubczyk, prof. UR
Imię i nazwisko osoby prowadzącej / osób prowadzących	dr hab. Paweł Jakubczyk, prof. UR mgr Paweł Śliż

* -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	30						15 (projekt)	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)

- Wykład – zaliczenie bez oceny
 Ćwiczenia – zaliczenie z oceną
 Zajęcia projektowe – zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Wiedza i umiejętności z matematyki (zgodnie z programem studiów), wiedza z zakresu podstaw informatyki.

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

3.1 Cele przedmiotu

C1	Zdobycie wiedzy na temat arytmetyki komputera, systemów liczbowych, kodu maszynowego i kodów źródłowych, genezy i podziału języków programowania
C2	Wykształcenie umiejętności konstrukcji algorytmów i ich zapisu w formie pseudokodu oraz w języku programowania wysokiego poziomu
C3	Wykształcenie umiejętności pisania programów w językach wysokiego poziomu

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu Student:	Odniesienie do efektów kierunkowych ¹
EK_01	Ma podstawową wiedzę w zakresie technik komputerowych, w tym metodyki tworzenia algorytmów i technik programowania.	K_Wo7
EK_02	Posiada umiejętność korzystania z anglojęzycznej wersji pomocy/podręcznika użytkownika aplikacji do pisania programów oraz dokumentacji języka programowania	K_Uo2
EK_03	Potrafi posługiwać się narzędziami informatycznymi do komputerowego rozwiązywania w wybranym języku programowania wybranych zagadnień inżynierii materiałowej.	K_Uo4
EK_04	Rozumie potrzebę podnoszenia swoich kwalifikacji, rozumie konieczność wzbogacania swojej wiedzy i umiejętności w związku ze zmianami zachodzącymi w technice i programowania.	K_Ko1

3.3 Treści programowe

A. Problematyka wykładu

Treści merytoryczne:
Komputer – podstawy działania, kod źródłowy, interpretacja kodu przez komputer. Arytmetyka maszyny - systemy liczbowe (binarny i szesnastkowy), konwersja liczb, reprezentacja liczb w maszynach cyfrowych, kody alfanumeryczne, formaty stałoprzecinkowe i zmiennoprzecinkowe, operacje arytmetyczne, błędy i dokładności obliczeń.
Program, instrukcja, programowanie, etapy tworzenia programu. Wprowadzenie do algorytmiki – podstawowe pojęcia, sposoby zapisu, schemat blokowy, cechy algorytmów.

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

Podstawowe pojęcia: język, translator, kompilator, interpreter, kod (źródłowy, maszynowy, wynikowy), kompilacja, konsolidacja, debugger. Generacje języków programowania.
Przygotowanie środowiska pracy dla programu Python dla Windows. Konsola i IDLE (Python's Integrated Development and Learning Environment), Visual Studio Code: instalacja i dostosowanie do języka Python.
Typy zmiennych i ich użycie w programie: int, float, str, listy, słowniki, zbiory, krotki.
Instrukcja If. Operatory logiczne: and, or, not.
Pętle: while, for, break i continue. Funkcje i ich argumenty.
Pakiety i moduły: instalacja i użycie na przykładzie: matplotlib, numpy. Środowisko Jupyter.
Przykłady użycia Pythona do rozwiązania problemów z zakresu Inżynierii materiałowej: analiza danych pomiarowych, dopasowanie danych pomiarowych, operacje na plikach.

B. Problematyka ćwiczeń laboratoryjnych

Treści merytoryczne:
Arytmetyka maszyny - system liczbowe (binarny i szesnastkowy), konwersja liczb, reprezentacja liczb w maszynach cyfrowych, kody alfanumeryczne, formaty stałoprzecinkowe i zmiennoprzecinkowe, operacje, błędy i dokładności obliczeń.
Wprowadzenie do algorytmiki – podstawowe pojęcia, sposoby zapisu, schemat blokowy, drzewo, cechy algorytmów.
Programowanie w języku Python: typy danych i operacje na nich (int, float, str, listy, słowniki, krotki i zbiory). Edytory wspomagające pisanie programów w języku Python.
Instrukcje sterujące: if, elif, obsługa wyjątków. Pętle „for” oraz „while”.
Funkcje, moduły i pakiety.
Wykorzystanie modułów i pakietów do operacji na wektorach, macierzach i sporządzaniu wykresów.
Operacje na plikach.

C. Problematyka zajęć projektowych

Treści merytoryczne
W ramach zajęć projektowych studenci realizują projekty przygotowane przez prowadzącego zajęcia lub zaproponowane przez siebie (po akceptacji pomysłu przez prowadzącego). Zakres projektów określa prowadzący zajęcia.

3.4 Metody dydaktyczne

Wykład: z prezentacją multimedialną

Ćwiczenia: ćwiczenia audytoryjne

Zajęcia projektowe: realizacja projektów.

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	Rozwiązywanie zadań na ćwiczeniach, Kolokwium, Obserwacja aktywności na wykładzie	W., Ćw.
EK_02	Obserwacja w trakcie zajęć	Ćw., Zaj proj.
EK_03	Ocena projektu	Zaj proj.
EK_04	Obserwacja w trakcie zajęć	W, Ćw., Zaj proj.

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Zaliczenie przedmiotu potwierdzi stopień osiągnięcia przez studenta zakładanych efektów uczenia się. Weryfikacja osiągniętych efektów uczenia się kontrolowana jest na bieżąco w trakcie realizacji zajęć. Ocena uzyskana z zaliczenia przedmiotu pozwoli ocenić stopień osiągniętych efektów.

Zaliczenie wykładu:

Zaliczenie na podstawie aktywności na wykładzie

Ocena z ćwiczeń:

Przygotowanie i realizacja zadań na oceny na ćwiczeniach. Ocena przygotowania do ćwiczeń.

Zaliczenie kolokwium pisemnych.

Ocena projektu:

Ocena napisanego projektu pod kątem efektywności, zastosowanej metody, poprawnego zapisu algorytmu do rozwiązania problemu.

Kryteria oceny dost. (51 - 60)% pkt, +dost. (61 - 70)% pkt, dobry (71 - 80)% pkt, +dobry (81 - 90)% pkt, bardzo dobry (91 - 100)% pkt.

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 akademickiego (udział w konsultacjach, egzaminie)	8
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	62
SUMA GODZIN	130
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	Nie dotyczy
zasady i formy odbywania praktyk	Nie dotyczy

7. LITERATURA

Literatura podstawowa:

1. W. Pirjanowicz, Podstawy programowania, Wydawnictwo Uniwersytetu Warmińsko-Mazurskiego, Olsztyn 2008.
2. A. Hodorowicz, ECDL: podstawy programowania w języku Python: sylabus v. 1.0 S10, PWN, Warszawa 2019.
3. P. Wróblewski, Algorytmy: struktury danych i techniki programowania, Helion, Gliwice 2010.
4. R. Kawa, J. Lembas, Wstęp do informatyki: wykłady z informatyki, PWN, Warszawa 2017.
5. M. Gorelock, I. Ozsvald, Python: programuj szybko i wydajnie, Helion, Gliwice 2015.
6. Z. Shaw, Python 3: proste wprowadzenie do fascynującego świata programowania, Helion, Gliwice 2018.
7. Dokumentacja dostępna na stronie: <https://www.python.org/>

Literatura uzupełniająca:

1. Dokumentacja dostępna na stronie: <https://matplotlib.org/>
2. Dokumentacja dostępna na stronie: <https://numpy.org/>

Akceptacja Kierownika Jednostki lub osoby upoważnionej