Załącznik nr 1.5 do Zarządzenia Rektora UR nr 7/2023

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA *2024-2028*
(skrajne daty)
Rok akademicki 2025/2026

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	Podstawy programowania
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	Wydział Nauk Ścisłych i Technicznych
Nazwa jednostki realizującej przedmiot	Wydział Nauk Ścisłych i Technicznych
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 Krzysztof Kucab
Imię i nazwisko osoby prowadzącej / osób prowadzących	dr Krzysztof Kucab, 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
Laboratorium - 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
C2	Wykształcenie umiejętności konstrukcji algorytmów i ich zapisu 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	Odniesienie do efektów kierunkowych [1]
EK_01	Ma podstawową wiedzę w zakresie technik komputerowych, w tym metodyki tworzenia algorytmów i technik programowania	K_W07
EK_02	Posiada umiejętność korzystanie z anglojęzycznej wersji pomocy/podręcznika użytkownika wybranego języka programowania	K_U02
EK_03	Potrafi posługiwać się narzędziami informatycznymi do komputerowego rozwiązywania w wybranym języku programowana wybranych zagadnień typowych dla inżynierii materiałowej	K_U04
EK_04	Rozumie potrzebę podnoszenia swoich kwalifikacji, rozumie konieczność wzbogacania swojej wiedzy i umiejętności do zmian zachodzących w technice i programowania	K_K01

3.3 Treści programowe

A. Problematyka wykładu

Treści merytoryczne
Podstawy działania komputera: - podstawy budowy komputera; - 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 (IEEE 754), operacje arytmetyczne, błędy i dokładności obliczeń.
Algorytmy i programowanie: - pojęcie programu; - instrukcje; - etapy tworzenia programu; - algorytmy – podstawowe pojęcia, sposoby zapisu, 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.

- schemat blokowy.
Podstawy języków programowania: - język, translator, kompilator, interpreter; - kod (źródłowy, maszynowy, wynikowy); - kompilacja, konsolidacja.
Praktyczne środowisko programistyczne: - przygotowanie środowiska pracy na przykładzie Microsoft Visual Studio; - instalacja i dostosowanie do języków C# oraz Python.
Programowanie proceduralne i obiektowe: - typy zmiennych; - instrukcje warunkowe; - operatory logiczne; - pętle - funkcje i argumenty; - klasy i obiekty.
Przykłady praktyczne: - zastosowanie języków wysokiego poziomu do rozwiązania problemów inżynierii materiałowej.

B. Problematyka laboratoriów

Treści merytoryczne
Arytmetyka maszyny - system liczbowe (binarny i szesnastkowy), konwersja liczb, reprezentacja liczb w maszynach cyfrowych, kody alfanumeryczne, formaty stałoprzecinkowe i zmienno-przecinkowe, operacje, błędy i dokładności obliczeń.
Wprowadzenie do algorytmiki – podstawowe pojęcia, sposoby zapisu, schemat blokowy, drzewo, cechy algorytmów (skończoność, jednoznaczność, efektywność).
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 sporządzania wykresów.
Operacje na plikach.

C. Problematyka projektu

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 – wykład z prezentacją multimedialną

Laboratorium - wykonanie doświadczeń i zadań praktycznych
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, Kolokwium, Obserwacja aktywności na wykładzie	Wykład, Laboratorium
EK_02	Obserwacja w trakcie zajęć	Projekt, Laboratorium
EK_03	Ocena projektu	Projekt
EK_04	Obserwacja w trakcie zajęć	Wykład, laboratorium, projekt

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Zaliczenie przedmiotu potwierdzi stopień osiągnięcia przez studenta zakładanych efektów uczenia się. Weryfikacja osiąganych 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 i testu końcowego w formie pytań zamkniętych.

Ocena z laboratorium: Ocena przygotowania do zajęć i realizacji zaplanowanych zadań. Zaliczenie kolokwiów.

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 z harmonogramu studiów	60
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	8
Godziny niekontaktowe – praca własna studenta	57

(przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	
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	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: syllabus 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. P. Barry, Python : przewodnik dla uczących się podstaw Pythona, , Helion, Gliwice 2024.
7. Z. Shaw, Python 3: proste wprowadzenie do fascynującego świata programowania, Helion, Gliwice 2018.
8. Dokumentacja dostępna na stronie: https://www.python.org/

Literatura uzupełniająca:

1. J. Albahari, „C# 12 w pigułce. Kompendium programisty", Helion 2024.
2. Dokumentacja dostępna na stronie: https://matplotlib.org/
3. Dokumentacja dostępna na stronie: https://numpy.org/
4. Dokumentacja języka C# dostępna pod adresem: https://learn.microsoft.com/pl-pl/dotnet/csharp/

Akceptacja Kierownika Jednostki lub osoby upoważnionej