

**SYLABUS****DOTYCZY CYKLU KSZTAŁCENIA 2023/2024 – 2026/2027***(skrajne daty)*

Rok akademicki 2023/2024

**1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE**

Nazwa przedmiotu	<b>Podstawy programowania</b>
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	Kolegium Nauk Przyrodniczych
Nazwa jednostki realizującej przedmiot	Kolegium Nauk Przyrodniczych Instytut Inżynierii Materiałowej
Kierunek studiów	Mechatronika
Poziom studiów	Studia I-go stopnia
Profil	praktyczny
Forma studiów	Studia niestacjonarne
Rok i semestr/y studiów	I rok, 2 semestr
Rodzaj przedmiotu	Przedmiot podstawowy
Język wykładowy	polski
Koordinator	prof. dr hab. inż. Lucyna Leniowska
Imię i nazwisko osoby prowadzącej / osób prowadzących	prof. dr hab. inż. Lucyna Leniowska dr hab. inż. prof. UR Krzysztof Szemela

\* -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
2	9	9		18					4

**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.

Laboratoria – zaliczenie z oceną.

**2. WYMAGANIA WSTĘPNE**

Znajomość zagadnień z podstaw informatyki, algorytmiki, analizy funkcjonalnej i logiki.

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

#### 3.1 Cele przedmiotu

C1	Zapoznanie ze środowiskiem programistycznym, składnią języka C/C++ i strukturą programu.
C2	Zapoznanie z programowaniem proceduralnym i podstawowymi algorytmami.

#### 3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych <sup>1</sup>
EK_01	Student zna i rozumie obsługę kompilatora języka C. Posiada wiedzę z zakresu: składni języka C, algorytmów, programowania strukturalnego. Definiuje i stosuje typy strukturalne statyczne. Definiuje i stosuje podstawowe elementy składni języka programowania C. Definiuje własne funkcje i procedury.	K_W09
EK_02	Student potrafi pozyskać wiedzę z dostępnych źródeł w celu stworzenia efektywnie działającego oprogramowania. Potrafi pozyskać informację z literatury anglojęzycznej jak również anglojęzycznych stron internetowych.	K_U01
EK_03	Student potrafi stworzyć algorytmy w celu rozwiązania postawionego zadania. Potrafi planować i przeprowadzać testy, w tym symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski. Potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych napisane programy.	K_U04
EK_04	Student potrafi napisać prosty program i przygotować niezbędną dokumentację.	K_U07
EK_05	Student potrafi posługiwać się językiem obcym na poziomie B2	K_U16
EK_06	Student potrafi zaplanować proces własnego uczenia się rozumiejąc potrzebę ciągłej aktualizacji wiedzy dotyczącej wiodących narzędzi programistycznych i algorytmów.	K_U19
EK_07	Student rozumie potrzebę i jest gotów do krytycznej oceny własnej wiedzy oraz wynikających z niej aspektów i skutków działalności inżyniera. Ma świadomość wpływu profesjonalnego przygotowania oprogramowania i jego dokumentacji na wyniki finansowe instytucji.	K_K01

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

### 3.3 Treści programowe

#### A. Problematyka wykładu (9 godz.)

Treści merytoryczne
1. Wprowadzenie do programowania. Pojęcie algorytmu i programu. Podstawowe zasady analizy algorytmów: poprawność i złożoność obliczeniowa algorytmu (pesymistyczna, oczekiwana) Kompilatory i języki programowania. Języki programowania i ich generacje. Paradygmaty.
2. Podstawowe elementy języka ANSI C. Programowanie proceduralne – ogólne założenia. Budowa programu, pliki nagłówkowe, operacje wejścia i wyjścia, semantyka zmiennej, zapis wyrażen arytmetycznych.
3. Przegląd konstrukcji języka C. Typy danych i ich opis. Specyfikatory i modyfikatory. Definiowanie typów. Instrukcje proste: przypisania, skoku, pusta. Przykłady.
4. Instrukcje strukturalne. Instrukcja warunkowa if...else pojedyncza i zagnieżdżona. Instrukcja wyboru switch. Przykłady.
5. Pętle. Organizowanie obliczeń cyklicznych o znanej ilości powtórzeń - pętla for dla zmiennych prostych. Warunek zakończenia działania pętli – instrukcje while i do while. Procedury continue i break. Obliczenia skończone i nieskończone. Przykłady algorytmów.
6. Funkcje. Definiowanie i wywoływanie funkcji. Zwracanie wartości, typ void. Przekazywanie parametrów, pojęcie wskaźnika. Parametry formalne i aktualne. Zakres widzialności. Rekurencja. Przykłady.
7. Typy strukturalne: tablice jednowymiarowe i wielowymiarowe. Deklaracja, wprowadzanie i wyprowadzanie danych, przykłady zastosowań.
8. Podsumowanie przedmiotu. Zaliczenie.

#### B. Problematyka ćwiczeń (9 godz.)

Treści merytoryczne
1. Zajęcia organizacyjne, zasady pracy, warunki zaliczenia. Wprowadzenie do języka ANSI C. Budowa programu. Zapis liczby, tworzenie identyfikatorów Operatory i ich priorytety. Tłumaczenie wyrażen arytmetycznych Zmienne i typy danych. Instrukcja przypisania. Podstawowe funkcje matematyczne z biblioteki <math.h>. Biblioteka <stdio.h> obsługa wejścia/wyjścia.
2. Instrukcje sterujące if oraz switch.
3. Organizowanie obliczeń cyklicznych o znanej ilości powtórzeń - pętla <b>for</b> ; Warunek zakończenia obliczeń pętli - instrukcje <b>while</b> i <b>do.. while</b> .
4. <b>Kolokwium nr 1</b>
5. Definicje i wywołanie funkcji. Typ <b>void</b> . Przekazywanie parametrów przez wartość i przez wskaźnik. Zmienne globalne/lokalne. Zmienne indeksowane. Wskaźniki – podstawy.
6. Tablice jedno i dwuwymiarowe – Algorytmy zliczania, porównywania, porządkowania w odniesieniu do zmiennych tablicowych. Zadania z zastosowania funkcji, wskaźników i tablic.
7. <b>Kolokwium nr 2</b>

#### C. Problematyka laboratoriów (18 godz.)

Treści merytoryczne
1. Zajęcia organizacyjne, warunki zaliczenia, regulamin pracowni. Ogólne zasady posługiwania się środowiskiem programowania DevC++. (2 godz. ).

2.	Struktura programu, zasady tworzenia identyfikatorów. Zmienne i typy danych. Operatory i ich priorytety. Obsługa funkcji wejścia wyjścia. (2 godz. ).
3.	Obsługa wejścia/wyjścia. Operacje na zmiennych prostych z zastosowaniem instrukcji warunkowej <b>if</b> . Podstawy obsługi błędów. (2 godz. ).
4.	Zastosowanie zagnieżdżonej instrukcji warunkowej <b>if</b> oraz instrukcji <b>switch</b> . (2 godz. ).
5.	Wykorzystanie standardowych funkcji matematycznych do zapisu wzorów matematycznych ( <b>sin, cos, tan, asin, acos, atan, log, exp, pow, sqrt, fabs</b> ).
6.	Zastosowanie pętli <b>for</b> , losowanie liczb, szeregi skończone.
7.	Pętle <b>for, while, do while</b> : tablicowanie funkcji, szeregi nieskończone.
8.	Funkcje i procedury, definicja i wywołanie, przekazywanie parametrów przez wartość lub zmienną, zmienne globalne/lokalne.).
9.	Zastosowanie funkcji/procedur. Rekurencja.
10.	Operacje na wskaźnikach.
11.	Tablice jednowymiarowe tworzenie, operacje na tablicach, inicjowanie i wyświetlanie, podstawowe operacje statystyczne.
12.	Tablice dwuwymiarowe: wartość maksymalna, minimalna, średnia macierzy, główna przekątna, operacje na macierzach.
13.	Tablice i struktury.
14.	Zaliczenie laboratorium.

### 3.4 Metody dydaktyczne

Wykład – wykład z prezentacją multimedialną realizowany w formie zdalnej z wykorzystaniem platformy Microsoft Teams.

Ćwiczenia – rozwiązywanie zadań, analiza przykładów, dyskusja.

Laboratoria – rozwiązywanie zadań, praca w grupach, analiza przykładów, dyskusja.

## 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	lab., ćw.
EK_02	Obserwacja w trakcie zajęć, sprawozdanie	lab., ćw.
EK_03	Kolokwium	lab., ćw.
EK_04	Kolokwium	lab., ćw.
EK_05	Kolokwium	lab., ćw.
EK_06	Obserwacja w trakcie zajęć,	lab., ćw.
EK_07	Obserwacja w trakcie zajęć	lab., ćw.

### 4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

#### Wykład:

Warunkiem zaliczenia wykładu jest uzyskanie pozytywnej oceny z zajęć ćwiczeniowych i laboratoryjnych. Uzyskanie pozytywnej oceny z zajęć ćwiczeniowych i laboratoryjnych uwzględnia sprawdzenie wiedzy wykładowej.

### Ćwiczenia

Efekty sprawdzane będą na 2 kolokwiach pisemnych podczas zajęć ćwiczeniowych.

Warunek uzyskania zaliczenia z przedmiotu: co najwyżej jedna nieobecność nieusprawiedliwiona oraz spełnienie każdego z niżej opisanych warunków:

1. uzyskanie co najmniej 50% punktów z każdego z kolokwiów,
2. uzyskanie pozytywnej oceny z odpowiedzi ustnych dotyczących zrealizowanych ćwiczeń.

Punktacja kolokwiów:

Ocena przedmiotu						
Przedział punktacji	0%-50%	51%-60%	61%-70%	71%-80%	81%-90%	91%-100%
Ocena	2,0	3,0	3,5	4,0	4,5	5,0

Ocena końcowa z ćwiczeń jest średnią arytmetyczną ocen uzyskanych z kolokwiów oraz odpowiedzi ustnych.

### Laboratoria

Efekty sprawdzane będą w trakcie zajęć laboratoryjnych (odpowiedź ustna lub pisemna).

Warunkiem zaliczenia jest opracowanie sprawozdań z realizacji ćwiczeń laboratoryjnych oraz uzyskanie ocen pozytywnych ze sprawozdań.

Ocena końcowa z laboratoriów jest średnią arytmetyczną ocen z odpowiedzi ustnych i sprawozdań.

## 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	36
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	14
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	50
SUMA GODZIN	100
<b>SUMARYCZNA LICZBA PUNKTÓW ECTS</b>	<b>4</b>

\* 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] Paul J. Deitel, Harvey Deitel: *Język C. Solidna wiedza w praktyce*. Wydanie VIII, Helion, 2020.
- [2] B. W. Kernighan, D. M. Ritchie: *„Język ANSI C: programowanie*. Helion, 2010.
- [3] K. N. King: *Język C: nowoczesne programowanie*. Helion, 2011.
- [4] C. L. Tondo, S. E. Gimpel: *Język ANSI C: programowanie, ćwiczenia*. Helion, 2010.
- [5] <http://upload.wikimedia.org/wikibooks/pl/6/6a/C.pdf> - książka dostępna na portalu wikibooks.

### Literatura uzupełniająca:

- [1] S. Prata: *Język C. Szkoła programowania*. Wydanie V, Helion 2008.
- [2] K. Wojtuszkiewicz: *Programowanie strukturalne i obiektowe*. T.I i T.II. PWN, 2009.
- [3] <http://4programmers.net/>  
„Programowanie w C.pdf” – Wikibooks  
<http://www.cplusplus.com/reference/clibrary/>

Akceptacja Kierownika Jednostki lub osoby upoważnionej