

**SYLABUS**

DOTYCZY CYKLU KSZTAŁCENIA 2021/22-2024/25

(skrajne daty)

Rok akademicki 2021/2022

**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
Kierunek studiów	Mechatronika
Poziom studiów	Studia I stopnia
Profil	Praktyczny
Forma studiów	Stacjonarne
Rok i semestr/y studiów	Rok I, semestr 2
Rodzaj przedmiotu	Podstawowy
Język wykładowy	J. polski
Koordinator	dr hab. inż. Lucyna . LENIOWSKA, prof. UR
Imię i nazwisko osoby prowadzącej / osób prowadzących	dr hab. inż. Lucyna . LENIOWSKA, prof. UR, dr hab. inż. Krzysztof Szemela, prof. UR

\* -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	15	30		30					4

**1.2. Sposób realizacji zajęć**

X zajęcia w formie tradycyjnej

X 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 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 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	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	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	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	Potrafi napisać prosty program i przygotować niezbędną dokumentację.	K_U07
EK_05	Potrafi posługiwać się językiem obcym na poziomie B2	K_U16
EK_06	Potrafi zaplanować proces własnego uczenia się rozumiejąc potrzebę ciągłej aktualizacji wiedzy dotyczącej wiodących narzędzi programistycznych i algorytmów.	K_U19
EK_07	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

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 godz.)
2. Podstawowe elementy języka ANSI C. Programowanie proceduralne – ogólne założenia. Budowa programu, pliki nagłówkowe, operacje wejścia i wyjścia, semantyka zmiennej, zapis wyrażeń arytmetycznych. (2 godz.)
3. Przegląd konstrukcji języka C. Typy danych i ich opis. Specyfikatory i modyfikatory. Definiowanie typów. Instrukcje proste: przypisania, skoku, pusta. Przykłady. (2 godz.)
4. Instrukcje strukturalne. Instrukcja warunkowa if...else pojedyncza i zagnieżdżona. Instrukcja wyboru switch. Przykłady. (2 godz.)
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. (2 godz.)
6. Funkcje. Definiowanie i wywoływanie funkcji. Zwrocenie wartości, typ void. Przekazywanie parametrów, pojęcie wskaźnika. Parametry formalne i aktualne. Zakres widzialności. Rekurencja. Przykłady. (2 godz.)
7. Typy strukturalne: tablice jednowymiarowe i wielowymiarowe. Deklaracja, wprowadzanie i wyprowadzanie danych, przykłady zastosowań. (2 godz.)
8. Podsumowanie przedmiotu. Zaliczenie. (1 godz.)

#### B. Problematyka ćwiczeń audytoryjnych

Treści merytoryczne
1. Zajęcia organizacyjne, zasady pracy, warunki zaliczenia. Wprowadzenie do języka ANSI C. Budowa programu. Zapis liczby, tworzenie identyfikatorów. (2 godz.)
2. Zmienne i typy danych. Instrukcja przypisania. Operatory i ich priorytety. Tłumaczenie wyrażeń arytmetycznych. (2 godz.)
3. Podstawowe funkcje matematyczne z biblioteki <math.h>. Biblioteka <stdio.h> – obsługa wejścia/wyjścia. Kody formatujące i znaki specjalne dla funkcji wejścia/wyjścia. (2 godz.)
4. Instrukcje sterujące if oraz switch. (2 godz.)
5. <b>Kolokwium nr 1</b> (1 godz.)
6. Organizowanie obliczeń cyklicznych o znanej ilości powtórzeń - pętla <b>for</b> dla zmiennych prostych. (3 godz.)
7. Warunek zakończenia obliczeń pętli - instrukcje <b>while</b> i <b>do.. while</b> . Algorytmy wyboru ekstremum, sumowanie szeregu z zadaną dokładnością, tablicowanie funkcji, obliczenia z krokiem, itp. (2 godz.)
8. Definicje i wywołanie funkcji. Typ <b>void</b> . Przekazywanie parametrów przez wartość i przez wskaźnik. Zmienne globalne/lokalne. (4 godz.)
9. <b>Kolokwium nr 2</b> (1 godz.)
10. Zmienne indeksowane. Obsługa ciągów znakowych. Wskaźniki – podstawy. Znakowe i łańcuchowe wejście/wyjście. (3 godz.)
11. Operacje na wskaźnikach. (2 godz.)
12. Tablice jedno i dwuwymiarowe – Algorytmy zliczania, porównywania, porządkowania w odniesieniu do zmiennych tablicowych. (2 godz.)
13. Zadania z zastosowania funkcji, wskaźników i tablic. (2 godz.)
14. <b>Kolokwium nr 3</b> (1 godz.)
15. <b>Zaliczenie ćwiczeń</b> (1 godz.)

### C. Problematyka zajęć laboratoryjnych.

Treści merytoryczne
1. Zajęcia organizacyjne, zasady pracy, warunki zaliczenia, regulamin pracowni. Ogólne zasady posługiwania się środowiskiem programowania <i>DevC++</i> . (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> ). (2 godz. )
6. Zastosowanie pętli <b>for</b> , losowanie liczb, szeregi skończone. (2 godz. )
7. Pętle <b>for, while, do while</b> : tablicowanie funkcji, szeregi nieskończone. (4 godz. )
8. Funkcje i procedury, definicja i wywołanie, przekazywanie parametrów przez wartość lub zmienną, zmienne globalne/lokalne. (4 godz. )
9. Zastosowanie funkcji/procedur. Rekurencja. (2 godz. )
10. Operacje na wskaźnikach. (2 godz. )
11. Tablice jednowymiarowe tworzenie, operacje na tablicach, inicjowanie i wyświetlanie, podstawowe operacje statystyczne. (2 godz. )
12. Tablice dwuwymiarowe: wartość maksymalna, minimalna, średnia macierzy, główna przekątna, operacje na macierzach. (2 godz. )
13. Zaliczenie laboratorium. (2 godz. )

### 3.4 Metody dydaktyczne

Wykład - prezentacja multimedialna,

Ćwiczenia – rozwiązywanie zadań, analiza przykładów, dyskusja,

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

## 3 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	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. METODY I KRYTERIA OCENY

Wykład - warunkiem zaliczenia jest zdobycie zaliczenia z ćwiczeń i laboratorium,

Ćwiczenia - zaliczenie 3 kolokwiów,

Laboratorium - wykonanie sprawozdań, zaliczenie ustne.

Efekty sprawdzane będą na 3 kolokwiach pisemnych podczas zajęć ćwiczeniowych oraz w trakcie zajęć laboratoryjnych (odpowiedź ustna). Warunek uzyskania zaliczenia przedmiotu: co najwyżej jedna nieobecność nieusprawiedliwiona oraz spełnienie każdego z niżej opisanych warunków:

1. uzyskanie co najmniej 50% punktów z każdego z dwóch kolokwiów
2. uzyskanie pozytywnej z odpowiedzi ustnych dotyczących realizowanych 6 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.

Poprawy:

Jednorazowa poprawa każdego ćwiczenia w trakcie zajęć/konsultacji w semestrze.

#### 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	75
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	5
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	20
<b>SUMA GODZIN</b>	<b>100</b>
<b>SUMARYCZNA LICZBA PUNKTÓW ECTS</b>	<b>4</b>

#### 6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	
zasady i formy odbywania praktyk	

#### 7. LITERATURA

Literatura podstawowa:

B. W. Kernighan, D. M. Ritchie, „*Język ANSI C : programowanie*”, Helion, 2010

K. N. King, „*Język C : nowoczesne programowanie*”, Helion, 2011

C. L. Tondo, S. E. Gimpel, „*Język ANSI C : programowanie, ćwiczenia*”, Helion, 2010

<http://upload.wikimedia.org/wikibooks/pl/6/6a/C.pdf> - książka dostępna na portalu wikibooks

[http://www.acm.uiuc.edu/webmonkeys/book/c\\_guide/](http://www.acm.uiuc.edu/webmonkeys/book/c_guide/)

#### **Literatura uzupełniająca:**

S. Prata, „*Język C. Szkoła programowania. Wydanie V*”, Helion 2008

B. Borowik, W. Borowik, B. Borowik, „*Meandry języka C/C++*”, Mikom 2006

P. Stańczyk, „*Algorytmika praktyczna*”, PWM, 2009

K. Wojtuszkiewicz, „*Programowanie strukturalne i obiektowe T. I i T.II*”, PWM, 2009

<http://4programmers.net/>

<http://computer.howstuffworks.com/c1.htm>

„*Programowanie w C.pdf*” – Wikibooks

<http://www.cplusplus.com/reference/clibrary/>

M. Grochowina, L. Leniowska The new method of the selection of features for the k-NN classifier in the arteriovenous fistula state estimation, Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds). ACSIS, Vol. 8, pages 281–285 (2016)

<http://dx.doi.org/10.15439/2016F244>

<https://fedcsis.org/proceedings/2016/drp/244.html>

L. Leniowska (red.) „*Nauka dla Gospodarki - Mechatronika*”, s.179, Wydawnictwo Uniwersytet Rzeszowski – Inprona, Rzeszów, 2011r., ISBN 978-83-63151-00-2

Akceptacja Kierownika Jednostki lub osoby upoważnionej