

SYLABUS**DOTYCZY CYKLU KSZTAŁCENIA 2024-2028***(skrajne daty)*

Rok akademicki 2024/2025

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	podstawy programowania w języku C
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	Instytut Informatyki
Nazwa jednostki realizującej przedmiot	Instytut Informatyki
Kierunek studiów	Informatyka i ekonometria
Poziom studiów	studia pierwszego stopnia
Profil	praktyczny
Forma studiów	Studia stacjonarne
Rok i semestr/y studiów	rok I, sem. I
Rodzaj przedmiotu	przedmiot kierunkowy
Język wykładowy	polski
Koordynator	dr inż. Michał Kępski
Imię i nazwisko osoby prowadzącej / osób prowadzących	dr inż. Michał Kępski, dr inż. Bogusław Twaróg

* -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
1	15			15					2

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)

ZALICZENIE Z OCENĄ

2. WYMAGANIA WSTĘPNE

Umiejętność programowania na poziomie szkoły średniej.

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

3.1 Cele przedmiotu

C1	Usystematyzowanie wiadomości ze szkoły średniej na temat programowania komputerów.
C2	Zapoznanie studentów z podstawowymi zasadami i narzędziami tworzenia programów oraz możliwościami współczesnych języków programowania wysokiego poziomu na przykładzie języka C.
C3	Wykształcenie / poprawienie umiejętności programowania w języku C.

3.2 Efekty uczenia się dla przedmiotu

K_Wo3, Uo1, Uo6,

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych ¹
EK_o1	Zna podstawowe pojęcia związane z programowaniem komputerów i konstrukcje programistyczne dostępne w języku C.	K_Wo3
EK_o2	Potrafi wykorzystać odpowiednie typy danych oraz konstrukcje programistyczne dostępne w języku C przy tworzeniu programów rozwiązujących zadane problemy.	K_Uo1
EK_o3	Potrafi tworzyć programy z wykorzystaniem standardowych bibliotek programistycznych dostępnych w języku C. Potrafi tworzyć programy wykorzystujące wskaźniki, struktury i unie. Potrafi tworzyć programy dynamicznie alokujące pamięć.	K_Uo6

3.3 Treści programowe

A. Problematyka wykładu

1. Podstawowe pojęcia związane z programowaniem komputerów. Programowanie strukturalne. Ogólne cechy języka C.
2. Struktura programu w języku C. Zmienne i stałe. Typy danych, literały, operatory i wyrażenia. Operacje wejścia i wyjścia.
3. Instrukcje sterujące: <code>if</code> , <code>if-else</code> , <code>switch</code> , <code>while</code> , <code>do-while</code> , <code>for</code> . Zagnieżdżanie instrukcji. Instrukcje przerwań <code>break</code> i <code>continue</code> .
4. Wskaźniki. Tablice jedno- i wielowymiarowe. Łańcuchy znaków.
5. Funkcje. Przekazywanie parametrów do funkcji. Zmienne globalne i lokalne. Rekurencja. Wybrane funkcje biblioteczne. Struktury i unie. Dynamiczny przydział pamięci.
6. Operacje na plikach. Tryby dostępu do pliku. Odczyt, zapis, dostęp swobodny.
7. Narzędzia deweloperskie języka C (kompilatory, debugery, <code>make</code> , <code>cmake</code>)

B. Problematyka ćwiczeń, konwersatoriów, laboratoriów, zajęć praktycznych

1. Środowisko programistyczne – konfiguracja. Tworzenie prostych programów w języku C. Funkcje <code>scanf()</code> i <code>printf()</code> do obsługi wejścia/wyjścia standardowego.
2. Tworzenie programów wykorzystujących instrukcje sterujące: <code>if</code> , <code>if-else</code> , <code>switch</code> .

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

Zagnieżdżanie instrukcji.
3. Tworzenie programów wykorzystujących pętle <code>while</code> , <code>do-while</code> , <code>for</code> . Instrukcje przerwań <code>break</code> i <code>continue</code> .
4. Definiowanie własnych funkcji. Parametry funkcji i wartości zwracane.
5. Tablice w języku C. Łańcuchy znaków, tablice wielowymiarowe.
6. Rekurencja w języku C.
7. Wskaźniki w języku C. Arytmetyka wskaźników.
8. Operacje na plikach. Odczyt, zapis, dostęp swobodny. Strumienie.
9. Struktury i unie.
10. Dynamiczna alokacja pamięci.

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną.

Laboratorium: wykonywanie praktycznych zadań programistycznych – praca przy komputerze.

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	Pisemne zaliczenie wykładu	Wykład
EK_02	Kolokwium	Laboratorium
EK_03	Kolokwium	Laboratorium
EK_02, EK_03	Obserwacja na zajęciach	Laboratorium

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Wykład – test

Efekt EK_01:

Zaliczenie: Student potrafi poprawnie odpowiedzieć na co najmniej 50% pytań związanych pojęciami związanymi z programowaniem komputerów i konstrukcjami programistycznymi dostępnymi w języku C.

Laboratorium – kolokwium praktyczne przy komputerze

Efekt EK_02:

DST: Student potrafi poprawnie przygotować programy rozwiązujące proste problemy, wykorzystując do tego celu niektóre z poznanych typów danych oraz konstrukcje programistyczne dostępne w języku C.

DB: Student potrafi poprawnie przygotować programy rozwiązujące średniozaawansowane problemy, wykorzystując do tego celu niektóre z poznanych typów danych oraz konstrukcje programistyczne dostępne w języku C.

BDB: Student potrafi poprawnie przygotować programy rozwiązujące zaawansowane problemy, wykorzystując do tego celu niektóre z poznanych typów danych oraz konstrukcje programistyczne dostępne w języku C.

Efekt EK_03:

DST: Student potrafi poprawnie przygotować programy, rozwiązujące proste problemy wykorzystując do tego celu możliwości standardowych bibliotek programistycznych dostępnych w języku C używanych na zajęciach. Potrafi poprawnie przygotować programy operujące na wskaźnikach, prostych strukturach i uniach.

DB: Student potrafi poprawnie przygotować programy, rozwiązujące średniozaawansowane problemy, wykorzystując możliwości standardowych bibliotek programistycznych dostępnych w języku C używanych na zajęciach. Potrafi poprawnie przygotować programy operujące na wskaźnikach (także wskaźnikach do funkcji), strukturach i uniach. Definiuje własne struktury i unie.

BDB: Student potrafi poprawnie przygotować programy, rozwiązujące zaawansowane problemy, wykorzystując możliwości standardowych bibliotek programistycznych dostępnych w języku C, także takich, które student poznawał samodzielnie. Potrafi poprawnie przygotować programy operujące na wskaźnikach (także wskaźnikach do funkcji), złożonych strukturach i uniach i ich tablicach. Definiuje własne struktury i unie, w tym także takie, które wymagają dynamicznego przydziału pamięci.

Laboratorium – obserwacja na zajęciach

Ocena z laboratorium jest średnią ocen uzyskanych za efekt EK_02 i EK_03. Rezultatem obserwacji na zajęciach może być podniesienie lub obniżenie oceny o pół stopnia w zależności od stopnia aktywności studenta.

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	30
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	2
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	28
SUMA GODZIN	60
SUMARYCZNA LICZBA PUNKTÓW ECTS	2

** Należy uwzględnić, że 1 pkt ECTS odpowiada 25-30 godzin całkowitego nakładu pracy studenta.*

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa:

1. Kernighan B.W., Ritchie D.M.: Język ANSI C: Programowanie. wyd. Helion, Gliwice, 2010.
2. Prata S.: Język C. Wyd. Helion, Gliwice, 2006.
3. Tondo C.L., Gimpel S.E.: Język ANSI C: Programowanie, ćwiczenia. wyd. Helion, Gliwice, 2010.
4. FREE SOFTWARE FOUNDATION. *The GNU C Reference Manual* [online]. [b.r.]. Dostępny w Internecie: <https://www.gnu.org/software/gnu-c-manual/>
5. Slajdy do wykładu: <https://github.com/majk3l/C-Introduction/>

Literatura uzupełniająca:

1. Shaw Z.A: Programowanie w C: sprytnie podejście do trudnych zagadnień, których wolałbyś unikać. Helion, Gliwice, 2014.
2. Reese R.: Wskaźniki w języku C: Przewodnik. Wyd. Helion, Gliwice, 2014.

Akceptacja Kierownika Jednostki lub osoby upoważnionej