



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Języki programowania

### Przedmiot

Kierunek studiów

Mechatronika

Studia w zakresie (specjalność)

Poziom studiów

pierwszego stopnia

Forma studiów

stacjonarne

Rok/semestr

2/4

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

15

Laboratoria

15

Inne (np. online)

Ćwiczenia

0

Projekty/seminaria

0

### Liczba punktów ECTS

2

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Wojciech Pietrowski

email: wojciech.pietrowski@put.poznan.pl

Wydział Automatyki, Robotyki i Elektrotechniki

ul. Piotrowo 3A, 60-965 Poznań, pokój 651

Odpowiedzialny za przedmiot/wykładowca:

### Wymagania wstępne

Podstawowa wiedza zakresie informatyki, sprzętu komputerowego, systemów operacyjnych, algorytmów i struktur danych. Obsługa: komputera, systemu operacyjnego Windows oraz oprogramowania użytkowego.

### Cel przedmiotu

Przekazanie podstawowej wiedzy z zakresu języków programowania, opanowanie umiejętności opracowywania algorytmów i podstaw programowania strukturalnego w języku C/C++.

### Przedmiotowe efekty uczenia się

Wiedza

Student zna podstawowe pojęcia z zakresu języków programowania, algorytmów, struktur danych oraz opracowywania programów komputerowych.

Student potrafi zastosować podstawowe techniki programowania oraz posługiwać się wybranym środowiskiem programistycznym.



Student potrafi zastosować metody tworzenia algorytmów i programów komputerowych do rozwiązywania wybranych zadań i problemów technicznych.

#### Umiejętności

Student umie formułować i rozwiązywać zadania oraz problemy techniczne przy wykorzystaniu inżynierskich metod analitycznych i symulacyjnych.

Student umie dokonać algorytmizacji zadań oraz wybranych problemów inżynierskich i opracowywania odpowiadającym im programów komputerowych w języku C/C++.

Student umie testować opracowane programy oraz oceniać poprawność ich funkcjonowania.

#### Kompetencje społeczne

Student rozumie potrzebę uczenia się przez całe życie; potrafi inspirować i organizować proces uczenia się innych osób.

#### Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wykład: zaliczenie na podstawie kolokwium składającego się z pytań ogólnych i testowych. Skala ocen 51-60% pkt. - dst, 61-70% pkt dst+, 71-80% pkt. - db, 81-90% pkt. - db+, 91-100% pkt. - bdb.

Laboratorium: premiowanie praktycznej wiedzy zdobytej w trakcie poprzednich ćwiczeń laboratoryjnych, sprawdzenie praktycznych umiejętności programowania w języku C++ (kolokwium zaliczeniowe), ocena wiedzy i umiejętności związanych z realizacją indywidualnych i grupowych projektów programistycznych.

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za: umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium, wykorzystanie elementów i technik wykraczających poza materiał z zakresu prowadzonego wykładu i ćwiczeń laboratoryjnych, staranność estetyczną zrealizowanych projektów.

#### Treści programowe

Wykład:

Podział języków programowania. Kompilacja, interpretacja. Języki niskiego i wysokiego poziomu. Przegląd języków programowania: Asembler, Basic, Fortran, C, C++, C#, Pascal, JAVA, PHP. Środowiska programistyczne. Edytory i kompilatory programów źródłowych, Debugger. Programowanie strukturalne. Podstawy programowania w języku Pascal i Object Pascal, C/C++.

Laboratorium:

Opracowanie oprogramowania wykorzystującego instrukcje warunkowe, instrukcje iteracyjne. Wykorzystanie bibliotek z funkcjami. Opracowanie funkcji własnych oraz bibliotek.

#### Metody dydaktyczne



Wykład: prezentacja multimedialna oraz prezentacja pisania i wykonywania wybranych programów bezpośrednio w języku C/C++.

Laboratorium: ćwiczenia praktyczne dotyczące elementów języka C/C++, pisanie programów okienkowych w tym języku.

### Literatura

Podstawowa

Stroustrup Bjarne, Programowanie : teoria i praktyka z wykorzystaniem C++, Helion 2020

Grębosz Jerzy, Opus Magnum C++11 : programowanie w języku C++, Helion 2020

Zieliński Józef, Podstawy programowania w języku C++, Oficyna Wydawnicza "Impuls", 2013

Uzupełniająca

Grębosz Jerzy, Pasja C++ , Oficyna Kallimach 2004

### Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	50	2,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	32	1,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwίων/egzaminu, wykonanie projektu) <sup>1</sup>	18	1,0

<sup>1</sup> niepotrzebne skreślić lub dopisać inne czynności