



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

**Programowanie strukturalne i obiektowe**

### Przedmiot

Kierunek studiów

Automatyka i robotyka

Studia w zakresie (specjalność)

-

Poziom studiów

pierwszego stopnia

Forma studiów

niestacjonarne

Rok/semestr

2/3

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

18

Ćwiczenia

Laboratoria

18

Projekty/seminaria

Inne (np. online)

### Liczba punktów

5

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Piotr Kaczmarek

Odpowiedzialny za przedmiot/wykładowca:

### Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu sprzętu komputerowego i jego obsługi oraz z zakresu przedmiotów semestru I: Podstaw Informatyki oraz Technologii Informacyjnych.

### Cel przedmiotu

Cel modułu kształcenia:

1. Zapoznanie metodologią i zasadami programowania strukturalnego oraz obiektowego wykorzystując język programowania C++ w zakresie poszerzonym względem tego, przedstawionego w semestrze I oraz elementami języka Python.



2. Zapoznanie z dynamicznymi strukturami danych i ich implementacją w języku C++ i Python. Wykształcenie praktycznych umiejętności adekwatnego wykorzystania struktur zgodnie z wymaganiami
3. Umiejętność implementowania i adaptowania standardowych algorytmów do rozwiązywania różnorodnych problemów, oraz zagadnień związanych ze złożonością obliczeniową i optymalizacją
4. Znajomość podstawowych wzorców projektowych aplikacji i przykładu ich użycia

### Przedmiotowe efekty uczenia się

#### Wiedza

1. Ma uporządkowaną w zaawansowanym stopniu wiedzę w zakresie wybranych algorytmów i struktur danych oraz metodyki i technik programowania proceduralnego i obiektowego; -

[K1\_W8](P6S\_WG)

#### Umiejętności

1. potrafi opracować rozwiązanie prostego zadania inżynierskiego oraz zaimplementować, przetestować i uruchomić go w wybranym środowisku programistycznym na komputerze klasy PC dla wybranych systemów operacyjnych; - [K1\_U26(P6S\_UW)]

#### Kompetencje społeczne

1. jest gotów do krytycznej oceny posiadanej wiedzy; rozumie potrzebę i zna możliwości ciągłego doskonalenia się – podnoszenia kompetencji zawodowych, osobistych i społecznych, potrafi inspirować i organizować proces uczenia się innych osób; - [K1\_K1(P6S\_KR)]

### Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wykład: egzamin w formie pisemnej z zakresu wykładu

Laboratorium: sprawdzenie praktycznych umiejętności z zakresu algorytmów i struktur danych programowania obiektowego w języku C++ , oraz umiejętności wykorzystania bibliotek języka C++ STL. Ocena jest wypadkową ocen z 2 kolokwium, oceny pracy na zajęciach, zadań domowych.

### Treści programowe

Program wykładu i zajęć laboratoryjnych obejmuje następujące zagadnienia:

- dynamiczne struktury danych (tablica, lista, drzewo, tablica hashująca, stos, graf) budowa, implementacja w różnych językach programowania (C++, Python) oraz wydajność i zastosowania,
- algorytmy: złożoność algorytmów, podejście rekursywne i iterowane, sortowanie i przeszukiwanie, algorytmy dla struktur drzewiastych, algorytmy grafowe
- wzorce projektowe (m.in. Model Control View, Model View, Singleton, Dekorator, Strategy, Observer, Adapter)
- STL C++11,14 (kontenery i algorytmy, predykaty, wyrażenia regularne), programowanie generyczne (szablony).



## Metody dydaktyczne

1. Wykład: prezentacja multimedialna, ilustrowana przykładami podawanymi na tablicy, oraz programami stworzonymi w trakcie zajęć.
2. Ćwiczenia laboratoryjne: samodzielne ćwiczenie materiału wspomagane materiałami dydaktycznymi umieszczanymi na platformie e-learningowej

## Literatura

### Podstawowa

1. Opus Magnum C++11 : programowanie w języku C++. T. 1-3 / Jerzy Grębosz. Wydawnictwo Helion, cop. 2018.
2. materiały dydaktyczne udostępnione dla zajęć laboratoryjnych i wykładu:  
<https://moodle.put.poznan.pl>
3. Brad Miller and David Ranum "Problem Solving with Algorithms and Data Structures using Python"  
Luther College 2018 (dostępna online)

### Uzupełniająca

1. B. Eckel, Thinking In C++, Edycja polska, Wydawnictwo Helion
2. Podstawy programowania C i C++ - skrypt/P. Kaczmarek, D. Belter.

Wydawnictwo Politechniki Poznańskiej 2011

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	124	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	38	2
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu) <sup>1</sup>	86	3

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

