



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie i język C

### Przedmiot

Kierunek studiów

Edukacja Techniczno Informatyczna

Studia w zakresie (specjalność)

Poziom studiów

pierwszego stopnia

Forma studiów

stacjonarne

Rok/semestr

1/2

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

15

Ćwiczenia

Laboratoria

30

Projekty/seminaria

Inne (np. online)

### Liczba punktów ECTS

3

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Małgorzata A. Jankowska

e-mail: malgorzata.jankowska@put.poznan.pl

tel. +48 61 665-20-69

Wydział Inżynierii Mechanicznej

ul. Jana Pawła II 24, 60-965 Poznań

tel.: +48 61 665-23-60

Odpowiedzialny za przedmiot/wykładowca:

### Wymagania wstępne

Podstawowa wiedza z informatyki oraz matematyki. Umiejętność programowania w dowolnym języku wysokiego poziomu na poziomie podstawowym, algorytmizacji zadań oraz logicznego i abstrakcyjnego myślenia. Zrozumienie potrzeby tworzenia programów komputerowych w celu zwiększenia wydajności pracy, obliczeń, wizualizacji wyników oraz prezentacji informacji zgromadzonych w bazach danych.

### Cel przedmiotu

Głównym celem przedmiotu jest zapoznanie studenta z zasadami programowania w języku C/C++, zdobycie przez niego umiejętności samodzielnego tworzenia aplikacji wykorzystujących podstawowe cechy tego języka oraz algorytmizacji zadań.



## Przedmiotowe efekty uczenia się

### Wiedza

W01 Znajomość pojęć i zagadnień informatycznych niezbędnych do zrozumienia filozofii programowania w językach wysokiego poziomu, w tym w języku C/C++. K1\_W01, K1\_W08

W02 Znajomość i rozumie elementów języka C/C++ niezbędnych do tworzenia rozbudowanych aplikacji informatycznych w zakresie obejmowanym przez treści programowe właściwe dla kierunku studiów. K1\_W01, K1\_W08

W03 Wiedza niezbędna do samodzielnego tworzenia algorytmów implementowanych w języku C/C++. K1\_W01, K1\_W08

### Umiejętności

U01 Umiejętność tworzenia programów komputerowych z wykorzystaniem języka programowania C/C++. K1\_U01, K1\_U02, K1\_U04, K1\_U11

U02 Zdolność wykorzystania zdobytych umiejętności w zakresie tworzenia algorytmów i efektywnej implementacji zadań obliczeniowych i problemów inżynierskich. K1\_U01, K1\_U02, K1\_U04, K1\_U11

U03 Umiejętność samokształcenia w zakresie poznawania zaawansowanych elementów języka C/C++ oraz innych języków wysokiego poziomu. K1\_U01, K1\_U02, K1\_U04, K1\_U11

U04 Umiejętność pozyskiwania informacji z literatury, baz danych oraz innych dostępnych źródeł wiedzy. K1\_U01, K1\_U02, K1\_U04, K1\_U11

### Kompetencje społeczne

K01 Odpowiedzialność za powierzone zadania, np. samodzielny projekt programistyczny. K1\_K01, K1\_K03, K1\_K05

K02 Rozumienie potrzeby ciągłego dokształcania się w celu podnoszenia kompetencji zawodowych, zdobywania aktualnej wiedzy z zakresu języków programowania oraz informatyki (np. poprzez czytanie czasopism informatycznych, uczestnictwie w kursach i studiach podyplomowych). K1\_K01, K1\_K03, K1\_K05

K03 Umiejętność pracy nad wyznaczonym zadaniem samodzielnie oraz w zespole przyjmując w nim różne role. K1\_K01, K1\_K03, K1\_K05

K04 Rozumienie znaczenia algorytmizacji zadań w procesie tworzenia efektywnych programów komputerowych. K1\_K01, K1\_K03, K1\_K05

## Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

### Wykłady

Pisemne kolokwia obejmujące materiał wykazany w treściach programowych.

### Laboratorium



Zadania projektowe wymagane w formie programów i oceniane pod koniec zajęć laboratoryjnych oraz na kolokwiach zaliczeniowych.

### Treści programowe

Podstawowe pojęcia i zagadnienia:

- języki programowania (języki niskiego i wysokiego poziomu oraz ich charakterystyka),
- historia języków programowania C/C++ oraz ich zalety,
- standard języka C++,
- wprowadzenie do pracy w zintegrowanym środowisku programistycznym Visual Studio,
- systemy liczbowe – binarny i heksadecymalny system liczbowy,
- standardy reprezentowania cyfr, liter i znaków specjalnych.

Typy danych i zmienne:

- definicja i podział na typy proste (skalarne) i we (strukturalne),
- charakterystyka typów prostych – całkowitoliczbowego, zmiennopozycyjnego, logicznego, znakowego.

Jednostki leksykalne języka C++ (słowa kluczowe, identyfikatory, literały, stałe).

Operatory:

- operatory arytmetyczne (addytywne, multiplikatywne, inkrementacji, dekrementacji),
- operatory logiczne (koniunkcji, alternatywy, negacji),
- operatory przypisania i porównania,
- operator warunkowy.

Wskaźniki, tablice i struktury:

- typy i zamienne wskaźnikowe – deklaracja, inicjacja i dereferencja,
- typ void – wskazania adresowe typu void,
- typy i zmienne referencyjne,
- tablice statyczne – deklaracja, inicjacja, dostęp do elementów tablicy,
- struktury – deklaracja i dostęp do elementów struktury,
- tablice dynamiczne.

Wyrażenia i instrukcje języka C++.



Arytmetyka wskaźników – związek między tablicami statycznymi i wskaźnikami.

Operacje wejścia/wyjścia na plikach.

Funkcje języka C++.

### Metody dydaktyczne

Wykład: prezentacja multimedialna ilustrowana przykładowymi programami napisanymi w języku C/C++.

Ćwiczenia laboratoryjne: tworzenie programów komputerowych w języku C/C++.

### Literatura

#### Podstawowa

1. H. M. Deitel, P. J. Deitel, Arkana C++ Programowanie, Wydawnictwo RM, Warszawa 1998.
2. S. Prata, Szkoła Programowania. Język C++, Wydawnictwo Helion, Gliwice 2006.
3. A. Zalewski, Programowanie w językach C i C++ z wykorzystaniem pakietu Borland C++, Wydawnictwo Nakom, Poznań 1996.
4. J. Grębosz, Symfonia C++. Programowanie w języku C++ orientowane obiektowo, Tom 1,2,3, Oficyna Kallimach, Kraków 1999.

#### Uzupełniająca

1. D. E. Knuth, Sztuka programowania. Tom1 Algorytmy podstawowe, Wydawnictwa Naukowo-Techniczne, Warszawa 2002.
2. N. Wirth, Algorytmy + struktury danych = programy, Wydawnictwa Naukowo-Techniczne, Warszawa 2004.

### Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	86	3,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	51	2,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu) <sup>1</sup>	50	2,0

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