



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Informatyka [S1MiKC1>INF]

### Przedmiot

Kierunek studiów

Mikroelektronika i komunikacja cyfrowa

Rok/Semestr

1/2

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

30

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

### Liczba punktów ECTS

4,00

### Koordynatorzy

dr inż. Michał Sybis

michal.sybis@put.poznan.pl

### Wykładowcy

### Wymagania wstępne

Podstawowe wiadomości z logiki matematycznej i kombinatoryki. Umiejętność formułowania prostych algorytmów. Potrafi pozyskiwać informacje z literatury oraz innych źródeł w języku polskim lub angielskim; potrafi integrować uzyskane informacje, dokonywać ich interpretacji i wyciągać wnioski. Zna ograniczenia własnej wiedzy i umiejętności, rozumie konieczność dalszego kształcenia się.

### Cel przedmiotu

Przedmiot wprowadza podstawowe zagadnienia programowania proceduralnego w języku C++. Studenci poznają kluczowe konstrukcje języka, takie jak typy danych, operatory, instrukcje warunkowe i pętle, a także uczą się wykorzystywać funkcje do strukturyzowania kodu. Szczególną uwagę poświęca się mechanizmom rekurencji, jej zastosowaniom oraz porównaniu z podejściem iteracyjnym. Kurs obejmuje również zagadnienia związane z obsługą wejścia i wyjścia, zarządzaniem pamięcią oraz organizacją kodu w sposób modułowy. Celem zajęć jest rozwinięcie umiejętności samodzielnego pisania poprawnych, czytelnych i efektywnych programów.

### Przedmiotowe efekty uczenia się

Wiedza:

1. Podstawowa wiedza teoretyczna i praktyczna w zakresie programowania w języku C i C++, ze szczególnym uwzględnieniem zasad konstruowania poprawnych programów, stosowania funkcji, pętli, instrukcji warunkowych oraz podstaw rekurencji.
2. Ogólna wiedza na temat projektowania prostych programów i wykorzystania podstawowych funkcji bibliotecznych w codziennej praktyce programistycznej.

Umiejętności:

1. Przy projektowaniu oprogramowania student potrafi przeprowadzić analizę problemu z punktu widzenia postępowania algorytmicznego stosując kryteria szybkości działania programu, skalowalności zastosowanych rozwiązań, oraz adekwatności przyjętych metod.
2. Przy projektowaniu oprogramowania student potrafi poprawnie podzielić problem na mniejsze części, dobrać odpowiednie podejścia algorytmiczne, zidentyfikować kluczowe zależności oraz skutecznie organizować kod przy użyciu funkcji i modułów.

Kompetencje społeczne:

1. Zrozumienie potrzeby szerszej popularyzacji wiedzy z zakresu nowoczesnych technik informatycznych.
2. Świadomość możliwości i ograniczeń współczesnej informatyki przy jednoczesnym otwarciu na możliwość zastosowań w nowych dziedzinach życia codziennego, gospodarki, techniki i nauki.
3. Umiejętność formułowania własnych opinii na temat aktualnie stosowanych i dostępnych technologii i rozwiązań w projektowaniu nowoczesnych systemów informatycznych

### Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza i umiejętności nabyte w trakcie wykładów są weryfikowane podczas egzaminu. Egzamin ma formę pisemną. Składa się 10-15 pytań otwartych, które nie muszą być równo punktowane. Próg zaliczeniowy dla egzaminu pisemnego to 50% możliwych do zdobycia punktów.

Umiejętności nabyte podczas realizacji zajęć laboratoryjnych są oceniane na podstawie: krótkich testów na początku zajęć (tzw. wejściówki), pracy studentów na zajęciach, oraz kolokwium końcowego (lub dwóch kolokwium). Próg zaliczeniowy to 50% możliwych do zdobycia punktów.

### Treści programowe

W ramach przedmiotu studenci zapoznają się z podstawami programowania w języku C++. Struktura przedmiotu w C++ obejmuje podstawowe elementy programu oraz organizację kodu źródłowego. Omówione zostaną podstawowe typy danych, reprezentacja liczb binarnych, operatory i wyrażenia, w tym operatory arytmetyczne, logiczne, porównania oraz przypisania. Zakres zajęć obejmuje także operacje bitowe, instrukcje sterujące, tablice, funkcje oraz rekurencję. Dodatkowo studenci zapoznają się z metodami debugowania i profilowania kodu, co pozwoli na efektywne diagnozowanie i optymalizację programów.

Zajęcia laboratoryjne stanowią praktyczne ujęcie zagadnień omawianych na wykładach i pozwalają studentom na samodzielne stosowanie zdobytej wiedzy w rzeczywistych implementacjach. W trakcie ćwiczeń studenci zapoznają się z podstawami programowania w języku C++, ucząc się tworzenia, kompilowania i uruchamiania programów.

Ogólnie przyjęta skala ocen:

- 2.0 dla <0%; 50%>
- 3.0 dla (50%; 60%>
- 3.5 dla (60%; 70%>
- 4.0 dla (70%; 80%>
- 4.5 dla (80%; 90%>
- 5.0 dla (90%; 100%>

### Tematyka zajęć

Zakres wykładu oraz zajęć laboratoryjnych obejmuje:

1. Struktura programu w języku C++.
2. Podstawowe typy danych, konwersja danych.
3. Operatory i wyrażenia, operacje bitowe.
4. Systemy liczbowe.
5. Instrukcje sterujące.

6. Tablice., tablice wielowymiarowe.
7. Funkcje, przekazywanie argumentów, przeciążanie funkcji, rekurencja.
8. Wybrane metody sortowania i wyszukiwania.
9. Debugowanie i profilowanie kodu.

## Metody dydaktyczne

Wykład: prezentacja multimedialna, ilustrowana przykładami podawanymi na tablicy.  
 Laboratoria: ćwiczenia praktyczne - realizacja zadań podanych przez prowadzącego

## Literatura

Podstawowa:

1. Jerzy Grębosz, Symfonia C++ : programowanie w języku C++ orientowane obiektowo. T. 1/2/3, 2000
2. Jerzy Grębosz, Pasja C++ : szablony, pojemniki i obsługa sytuacji wyjątkowych w języku C++. T. 1/2, 2004
3. Jerzy Grębosz, Opus Magnum C++11 : programowanie w języku C++. T. 1/2/3, 2018
4. Bruce Eckel, Thinking in C++. Edycja polska

Uzupełniająca:

1. Stephen Prata, Język C++.
2. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Wprowadzenie do algorytmów, WNT, Warszawa, 2004

## Bilans nakładu pracy przeciętnego studenta

|                                                                                                                                                      | Godzin | ECTS |
|------------------------------------------------------------------------------------------------------------------------------------------------------|--------|------|
| Łączny nakład pracy                                                                                                                                  | 120    | 4,00 |
| Zajęcia wymagające bezpośredniego kontaktu z nauczycielem                                                                                            | 60     | 2,00 |
| Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu) | 60     | 2,00 |