



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Systemy informatyczne`

### Przedmiot

Kierunek studiów

Edukacja Techniczno Informatyczna

Studia w zakresie (specjalność)

Poziom studiów

pierwszego stopnia

Forma studiów

stacjonarne

Rok/semestr

2/3

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 inż. Małgorzata A. Jankowska

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

tel. +48 61 665-20-69

Wydział Inżynierii Mechanicznej

ul. Piotrowo 3, 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

### Wymagania wstępne

Od studneta wymaga się posiadania podstawowej wiedzy z informatyki. Umiejętność biegłego programowania w języku C/C++, algorytmizacji zadań oraz logicznego i abstrakcyjnego myślenia. Zrozumienie potrzeby tworzenia systemów informatycznych w celu zwiększenia wydajności pracy, obliczeń, wizualizacji wyników oraz prezentacji informacji zgromadzonych w bazach danych.

### Cel przedmiotu

Głównym celem przedmiotu jest przekazanie wiedzy dotyczącej zaawansowanych elementów języka C/C++, których znajomość umożliwia tworzenie autorskich, złożonych systemów informatycznych.



## Przedmiotowe efekty uczenia się

### Wiedza

W02 zna i rozumie zaawansowane elementy języka C/C++, wykorzystywane do tworzenia aplikacji informatycznych, w tym programowanie obiektowe, przeciążanie operatorów, obsługę wyjątków, dynamiczne struktury danych, przestrzenie nazw. K1\_W08

W03 ma wiedzę niezbędną do tworzenia aplikacji okienkowych działających w systemie Windows. K1\_W08

### Umiejętności

U02 ma umiejętność tworzenia programów komputerowych z wykorzystaniem zaawansowanych elementów języka programowania C/C++. K1\_U02, K1\_U04, K1\_U11

U03 posiada umiejętność pozyskiwania informacji z literatury, baz danych oraz innych dostępnych źródeł wiedzy. K1\_U02, K1\_U04, K1\_U11

### Kompetencje społeczne

K01 potrafi dzielić się zdobytą wiedzą informatyczną z innymi w sposób zrozumiały. K1\_K01, K1\_K03, K1\_K05

K02 widzi znaczenie informatyzacji różnych dziedzin życia w odniesieniu do człowieka i społeczeństwa. K1\_K01, K1\_K03, K1\_K05

K03 rozumie potrzebę ciągłego doszkalania się w celu podnoszenia kompetencji zawodowych, zdobywania aktualnej wiedzy z zakresu ogólnie pojętej informatyki oraz języków programowania (np. poprzez czytanie czasopism informatycznych, uczestnictwie w kursach i studiach podyplomowych). K1\_K01, K1\_K03, K1\_K05

K04 wykazać się odpowiedzialnością za powierzone mu zadania, np. samodzielny projekt programistyczny. K1\_K01, K1\_K03, K1\_K05

K05 potrafi pracować nad wyznaczonym zadaniem samodzielnie oraz w zespole przyjmując w nim różne role. K1\_K01, K1\_K03, K1\_K05

## Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

W01 Sprawdziany pisemne oceniane na punkty, obejmujące analizę złożonych fragmentów programów. Ocena końcowa uzależniona od procentowej liczby punktów.

3	50.1%-70.0%
4	70.1%-90.0%
5	od 90.1%

U01 Praca indywidualna oceniana pod koniec zajęć laboratoryjnych. Zadania projektowe w formie programów.



3	50.1%-70.0%
4	70.1%-90.0%
5	od 90.1%

### Treści programowe

Zaawansowane elementy języka C++ wykorzystywane do tworzenia funkcji:

- tablice statyczne i dynamiczne, jedno- i dwuwymiarowe jako argumenty funkcji,
- rekurencja,
- modyfikator const,
- wskaźniki i referencje jako wynik funkcji,
- przeciążanie funkcji,
- szablony funkcji.

Programowanie obiektowe:

- klasy, obiekty klas,
- przeciążanie operatorów,
- obsługa wyjątków,
- dziedziczenie,
- polimorfizm.

Elementy programowania dla środowiska Windows z wykorzystaniem klas MFC.

### Metody dydaktyczne

Wykład: prezentacja multimedialna, rozwiązywanie przykładowych zadań na tablicy,

Ćwiczenia laboratoryjne: ćwiczenia praktyczne, wykonywanie eksperymentów, dyskusja, praca w zespole.

### 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, tom 1. 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	80	3,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	50	2,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu) <sup>1</sup>	30	1,0

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