



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Algorytmy i struktury danych 1 [S1Teleinf1>AiSD1]

### Przedmiot

Kierunek studiów  
Teleinformatyka

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

5,00

### Koordynatorzy

dr inż. Filip Idzikowski  
filip.idzikowski@put.poznan.pl  
prof. dr hab. inż. Jerzy Tyszer  
jerzy.tyszer@put.poznan.pl

### Wykładowcy

### Wymagania wstępne

Student powinien posiadać podstawową wiedzę z zakresu matematyki dyskretnej, kombinatoryki i rachunku prawdopodobieństwa. Powinien posiadać umiejętność wykonywania obliczeń za pomocą aparatu matematycznego z zakresu analizy matematycznej i rachunku prawdopodobieństwa oraz pozyskiwania informacji ze wskazanych źródeł.

### Cel przedmiotu

Celem przedmiotu jest przekazanie studentom podstawowej wiedzy dotyczącej podstawowych algorytmów matematyki dyskretnej i metod numerycznych, statycznych i dynamicznych struktur danych oraz zasad programowania obiektowego w języku C++.

### Przedmiotowe efekty uczenia się

Wiedza

Student ma podstawową wiedzę teoretyczną i praktyczną w zakresie programowania w językach C i

C++, z naciskiem na projektowanie programów poprawnie zbudowanych, projektowanie programów złożonych oraz wykorzystywanie oprogramowania bibliotecznego. Student ma także wiedzę o podstawowych algorytmach i strukturach danych wykorzystywanych w codziennej praktyce programisty.

#### Umiejętności

Przy projektowaniu oprogramowania student potrafi przeprowadzić analizę problemu z punktu widzenia postępowania algorytmicznego stosując kryteria złożoności obliczeniowej, skalowalności zastosowanych rozwiązań, oraz adekwatności przyjętych metod. Potrafi także krytycznie analizować dostępne oprogramowanie biblioteczne pod kątem zastosowania w realizowanym projekcie oraz zaproponować zasady współpracy w konfiguracji zbiorowego programisty

#### Kompetencje społeczne

Student ma świadomość możliwości i ograniczeń współczesnej informatyki przy jednoczesnym otwarciu na możliwość zastosowań w nowych dziedzinach życia codziennego, gospodarki, techniki i nauk

### Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza nabyta w ramach wykładu jest weryfikowana poprzez egzamin pisemny składający się z kilku zadań problemowych obejmujących treść wykładu i/lub rozwiązanie testu wyboru obejmującego około 15 pytań. Czas trwania egzaminu: 2h. Umiejętności nabyte w ramach ćwiczeń laboratoryjnych są weryfikowane na podstawie dwóch kolokwii pisemnych obejmujących zadania wykonywane w ramach zajęć. Ponadto umiejętności nabyte w ramach zajęć są na bieżąco oceniane na podstawie ćwiczeń projektowych i odpowiedzi ustnych. Ocenie podlega także aktywność na zajęciach.

### Treści programowe

Historia automatyzacji obliczeń, inżynieria oprogramowania, główne kategorie algorytmów. Struktura programu w C++, podstawowe typy danych oraz instrukcje. Reprezentacja binarna liczb całkowitych, systemy uzupełnieniowe, operatory bitowe. Złożoność obliczeniowa, problemy N i NP, modele obliczeń, maszyna Turinga. Funkcje, rola stosu, przekazywanie argumentów, przeciążanie funkcji, szablony funkcji, wyrażenia lambda. Rekurencja. Elementarne algorytmy sortowania. Kopiec binarny i kopcowanie. Szybkie sortowanie i sortowanie przez scalanie. Wyszukiwanie informacji. Kodowanie mieszające. Reprezentacja liczb zmiennoprzecinkowych. Błędy w obliczeniach numerycznych. Numeryczne rozwiązywanie równań. Znajdywanie ekstremum funkcji. Całkowanie numeryczne. Metoda eliminacji Gaussa. Porządek leksykograficzny, generowanie permutacji, transpozycje, algorytm Steinhaus-Johnsona-Trottera, podzbiory k-elementowe..

### Tematyka zajęć

Wykład: Historia automatyzacji obliczeń, inżynieria oprogramowania, główne kategorie algorytmów. Struktura programu w C++, podstawowe typy danych. Wyrażenia arytmetyczne i logiczne, relacje, rzutowanie, blok instrukcji. Instrukcje warunkowe, wybór wielowariantowy, pętle, tablice. Reprezentacja binarna liczb całkowitych, system uzupełnieniowy do 2, operatory bitowe, sito Eratostenesa. Złożoność obliczeniowa, problemy N i NP, modele obliczeń, maszyna Turinga. Funkcje, rola stosu, przekazywanie argumentów, argumenty domniemane, przeciążanie funkcji, szablony funkcji, wyrażenia lambda. Rekurencja, algorytm Euklidesa. Elementarne algorytmy sortowania – wstawianie, wybieranie, sortowane bąbelkowe, metoda Shella. Kopiec binarny i kopcowanie. Szybkie sortowanie i sortowanie przez scalanie. Wyszukiwanie binarne i interpolacyjne. Kodowanie mieszające. Wybór funkcji mieszającej. Kolizje i ich unikanie. Reprezentacja liczb zmiennoprzecinkowych. Błędy w obliczeniach numerycznych. Numeryczne rozwiązywanie równań. Metoda bisekcji, metoda Newtona-Raphsona. Znajdywanie ekstremum funkcji. Całkowanie numeryczne. Metoda eliminacji Gaussa. Porządek leksykograficzny, generowanie permutacji, transpozycje, algorytm Steinhaus-Johnsona-Trottera, podzbiory k-elementowe, losowe podzbiory k-elementowe.

Laboratoria: Zapoznanie się z systemem programowania dostępnym w laboratorium. Proste operacje na plikach z wykorzystaniem dostępnego edytora tekstu. Proste programy w języku C++: operacje we/wy, wyszukiwanie minimum/maximum, operacje na macierzach. Podstawowe instrukcje sterujące. Funkcje – przekazywanie parametrów, szablony funkcji, przykłady zastosowań. Rekurencja – proste algorytmy

rekurencyjne. Proste algorytmy sortowania (wstawianie, selekcja, sortowanie bąbelkowe). Sortowanie przez kopcowanie. Metody szybkiego sortowania. Wyszukiwanie binarne i interpolacyjne. Kodowanie mieszające i rozwiązywanie kolizji. Metody numeryczne: bisekcja, metoda Newtona-Raphsona. Generowanie prostych obiektów kombinatorycznych.

## Metody dydaktyczne

Wykład: prezentacja multimedialna, wspomagana przykładami podawanymi na tablicy.

Laboratoria: rozwiązywanie zadań podanych przez prowadzącego, projektowanie prostych algorytmów, kodowanie algorytmów w języku C++.

## Literatura

1. R. Sedgewick, Algorytmy w C++, Oficyna Wydawnicza READ ME, Łódź, 1999
2. N. Wirth, Algorytmy + struktury danych = programy, WNT, Warszawa, 1980.
3. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Wprowadzenie do algorytmów, WNT, Warszawa, 2004
4. E.W. Dijkstra, Umiejętność programowania, WNT, Warszawa, 1985.
5. J. Grębosz, Symfonia C++, Oficyna Kallimach, Kraków 2008.
6. W. Lipski, Kombinatoryka dla programistów, WNT, Warszawa, 1982.

## Bilans nakładu pracy przeciętnego studenta

|                                                                                                                                                      | Godzin | ECTS |
|------------------------------------------------------------------------------------------------------------------------------------------------------|--------|------|
| Łączny nakład pracy                                                                                                                                  | 120    | 5,00 |
| Zajęcia wymagające bezpośredniego kontaktu z nauczycielem                                                                                            | 64     | 3,00 |
| Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu) | 56     | 2,00 |