

# POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

## KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2014/2015

Wydział Inżynierii Elektrycznej i Komputerowej

Kierunek studiów: Elektrotechnika

Profil: Ogólnoakademicki

Forma studiów: niestacjonarne

Kod kierunku: Elek

Stopień studiów: II

Specjalności: Informatyczne systemy automatyki

### 1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Projektowanie i analiza algorytmów
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	Design and analysis of algorithms
KOD PRZEDMIOTU	WIEiK ELEKTROTECH oIIN PW15 14/15
KATEGORIA PRZEDMIOTU	Przedmioty specjalnościowe
LICZBA PUNKTÓW ECTS	4.00
SEMESTRY	3

### 2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁADY	ĆWICZENIA	LABORATORIA	LABORATORIA KOMPUTERO- WE	PROJEKTY	
3	15	0	0	10	5	0

### 3 CELE PRZEDMIOTU

**Cel 1** Poznanie podstawowych pojęć i zagadnień związanych ze złożonością obliczeniową algorytmów.

**Cel 2** Znajomość poznanego zbioru klasycznych metod programowania, podstaw wybranych zaawansowanych metod programowania.

**Cel 3** Napisanie i uruchomienie prostych programów dla wybranych metod programowania. Udoskonalenie umiejętności programowania w języku C/C++ oraz indywidualnej pracy programistycznej.

**Cel 4** Wykonanie wybranego zespołowego projektu programistycznego. Udoskonalenie umiejętności programowania w języku C/C++ oraz zespołowej pracy programistycznej.

## 4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1 Zaliczony przedmiot : Programowanie w C++.

## 5 EFEKTY KSZTAŁCENIA

**EK1 Wiedza** Znajomość wybranych zagadnień: klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).

**EK2 Wiedza** Znajomość poznanego zbioru klasycznych metod programowania i obszaru zastosowania każdej z nich.

**EK3 Umiejętności** Napisanie i uruchomienie programów w języku C/C++ dla wybranych klasycznych metod programowania.

**EK4 Kompetencje społeczne** Praca w małym zespole, podział zadań, efektywna współpraca w osiąganiu wyznaczonego celu, dzielenie się wiedzą, wywiązywanie się z przyjętych obowiązków, kierowanie pracą zespołu

## 6 TREŚCI PROGRAMOWE

WYKŁADY		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
<b>W1</b>	Wprowadzenie do algorytmów i ich złożoności obliczeniowej. Klasyfikacja problemów. Notacje asymptotyczne. Złożoność czasowa i pamięciowa. Klasyfikacja algorytmów. Hierarchia złożoności.	2
<b>W2</b>	Pojęcie rekursji. Rekursja liniowa i drzewiasta. Rekurencyjne wyznaczanie wartości ciągów: silnia, liczby Fibonacciego, współczynniki dwumienne. Inne algorytmy rekurencyjne: wieże Hanoi, sortowanie MergeSort i QuickSort, trawersowanie drzew binarnych. Analiza zalet i wad rekursji. Eliminacja rekursji: iteracja, zastosowanie stosu. Rozwiązywanie rekurencji: metoda podstawiania, metoda iteracyjna, metoda rekurencji uniwersalnej.	2
<b>W3</b>	Idea programowania dynamicznego. Wyznaczanie wartości ciągów metodą programowania dynamicznego: silnia, liczby Fibonacciego, współczynniki dwumienne, liczby Stirlinga. Inne algorytmy programowania dynamicznego: problem podziału zbioru, dyskretny problem plecakowy, problem iloczynu łańcucha macierzy. Wymagania czasowe i pamięciowe algorytmów programowania dynamicznego.	2
<b>W4</b>	Idea algorytmu zachłannego. Własność optymalnej podstruktury. Elementy teorii matroidów. Klasy matroidów: podziałowe, macierzowe, grafowe, transwersalne. Przykłady klasycznych algorytmów zachłannych: algorytm Gaussa, algorytm Kruskala, algorytm Huffmana. Wady i zalety algorytmów zachłannych.	2

WYKŁADY		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
<b>W5</b>	Wyszukiwanie wyczerpujące w przestrzeni rozwiązań a idea algorytmu wyszukiwania z nawrotami. Przykład : problem komiwojażera (TSP). Przykłady klasycznych algorytmów z nawrotami: problem skoczka szachowego, problem ośmiu hetmanów. Gry i ich reprezentacje w postaci drzew gier. Ocena rozwiązań w węzłach. Technika odcinania alfa-beta gałęzi drzewa gry. Wady i zalety algorytmów z nawrotami.	2
<b>W6</b>	Koncepcja wyszukiwania lokalnego. Przykład: problem spełnialności (SAT). Wyszukiwanie lokalne w optymalizacji. Przykłady: pokrycie wierzchołkowe grafu (VCP), minimalne drzewo rozpinające (MST), problem komiwojażera (TSP). Przykładowe operacje i definicje sąsiedztwa. Ulepszenia i modyfikacje wyszukiwania lokalnego. Algorytm wyszukiwania z listą tabu (TS). Przykład : problem MST z ograniczeniami. Zalety i wady wyszukiwania lokalnego.	2
<b>W7</b>	Idea algorytmów iteracyjnych. Podstawowe typy algorytmów iteracyjnych. Wyszukiwanie z tabu. Symulowane wyżarzanie. Algorytmy ewolucyjne. Schematy zrównoleglenia programów iteracyjnych.	1
<b>W8</b>	Programowanie równoległe. Charakterystyka systemów równoległych. Organizacja obliczeń równoległych. Przykład obliczeń równoległych sito Eratostenesa. Podstawowe miary jakości obliczeń równoległych. Zależności pomiędzy miarami jakości. Przykład oceny jakości systemu równoległego. Bariera: inherentna sekwencyjność problemów.	2

LABORATORIA KOMPUTEROWE		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
<b>K1</b>	Generacja grafów w reprezentacji tablicowej o zadanych własnościach.	2
<b>K2</b>	Programowanie dynamiczne : problem plecakowy.	3
<b>K3</b>	Algorytmy zachłanne : kodowanie Huffmana.	3
<b>K4</b>	Wyszukiwanie lokalne : algorytmy GSAT i WALKSAT dla problemu Spełnialności.	2

PROJEKTY		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
<b>P1</b>	Projekt zespołowy implementacji wybranego algorytmu dla problemu z klasy NP (TSP, kolorowanie grafu).	5

## 7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Praca w grupach

## 8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
<b>Godziny kontaktowe z nauczycielem akademickim, w tym:</b>	
Godziny wynikające z planu studiów	30
Konsultacje przedmiotowe	10
Egzaminy i zaliczenia w sesji	10
<b>Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:</b>	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	25
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	15
programowanie, testowanie programów	30
<b>SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA</b>	<b>120</b>
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	4.00

## 9 SPOSOBY OCENY

### OCENA FORMUJĄCA

F1 Projekt zespołowy

F2 Sprawozdanie z ćwiczenia laboratoryjnego

### OCENA PODSUMOWUJĄCA

P1 Średnia ważona ocen formujących

P2 Zaliczenie pisemne

### WARUNKI ZALICZENIA PRZEDMIOTU

W1 Obecności na wykładach i laboratoriach komputerowych

### KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 2.0	Brak znajomości przynajmniej 3 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
NA OCENĘ 3.0	Znajomość 3 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).symptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
NA OCENĘ 3.5	Znajomość 4 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
NA OCENĘ 4.0	Znajomość 5 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).symptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
NA OCENĘ 4.5	Znajomość 6 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).symptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
NA OCENĘ 5.0	Znajomość 6 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).symptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne) oraz wiedza wykraczająca poza program wykładu.
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 2.0	Brak znajomości teoretycznej przynajmniej 3 spośród 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne).
NA OCENĘ 3.0	Znajomość teoretyczna 3 spośród 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań.
NA OCENĘ 3.5	Znajomość teoretyczna 4 spośród 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań 3 z nich.
NA OCENĘ 4.0	Znajomość teoretyczna 4 spośród 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań.

NA OCENĘ 4.5	Znajomość teoretyczna 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań 4 z nich.
NA OCENĘ 5.0	Znajomość teoretyczna 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań.
EFEKT KSZTAŁCENIA 3	
NA OCENĘ 2.0	Brak napisania i uruchomienia co najmniej 1 programów w języku C/C++ dla zbioru podstawowych metod programowania.
NA OCENĘ 3.0	Napisanie i uruchomienie co najmniej 1 w pełni funkcjonalnego programu w języku C/C++ dla zbioru podstawowych metod programowania.
NA OCENĘ 3.5	Napisanie i uruchomienie co najmniej 2 programów w języku C/C++ dla zbioru podstawowych metod programowania (w tym 1 w pełni funkcjonalnego).
NA OCENĘ 4.0	Napisanie i uruchomienie 2 w pełni funkcjonalnych programów w języku C/C++ dla zbioru podstawowych metod programowania.
NA OCENĘ 4.5	Napisanie i uruchomienie co najmniej 3 programów w języku C/C++ dla zbioru podstawowych metod programowania (w tym 2 w pełni funkcjonalnych).
NA OCENĘ 5.0	Napisanie i uruchomienie 3 w pełni funkcjonalnych programów w języku C/C++ dla zbioru podstawowych metod programowania.
EFEKT KSZTAŁCENIA 4	
NA OCENĘ 2.0	Bierność lub niewywiązywanie się z przyjętych obowiązków lub destrukcyjny wpływ na pracę zespołu.
NA OCENĘ 3.0	Umiarkowana aktywność lub niewywiązanie się z części przyjętych obowiązków lub brak kreatywności lub brak współpracy w zespole.
NA OCENĘ 3.5	Zadowolająca aktywność, wywiązanie się z przyjętych obowiązków, przejawy kreatywności, poprawna współpraca w zespole w roli wykonawcy (łącznie)
NA OCENĘ 4.0	Dobra aktywność, wywiązanie się z przyjętych obowiązków, kreatywność, efektywna współpraca w zespole (łącznie)
NA OCENĘ 4.5	Dobra aktywność, wywiązanie się z przyjętych obowiązków, kreatywność, efektywna współpraca w zespole, transfer wiedzy do pozostałych członków zespołu (łącznie)
NA OCENĘ 5.0	Wyróżniająca się aktywność, wywiązanie się z przyjętych obowiązków, kreatywność, efektywna współpraca w zespole, transfer wiedzy do pozostałych członków zespołu, wykazanie się umiejętnościami kierowniczymi (łącznie)

## 10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK1		Cel 1	W1	N1	P2
EK2		Cel 2	W2 W3 W4 W5 W6 W7 W8	N1 N2	F2 P1 P2
EK3		Cel 3	K1 K2 K3 K4	N2	F2 P1
EK4		Cel 4	P1	N3	F1 P1

## 11 WYKAZ LITERATURY

### LITERATURA PODSTAWOWA

- [1 ] Cormen T.H., Leiserson C.E., Rivest R.L., C. Stein — *Wprowadzenie do algorytmów*, Warszawa, 2007, WN-T
- [2 ] Kernighan B.W., Pike R. — *Lekcja programowania*, Warszawa, 2002, WN-T
- [3 ] Lipski W. — *Kombinatoryka dla programistów*, Warszawa, 2004, WN-T

### LITERATURA UZUPEŁNIAJĄCA

- [1 ] Sedgewick R. — *Algorytmy w C++. Sortowanie i wyszukiwanie*, Warszawa, 1999, Wydawnictwo RM
- [2 ] Sedgewick R. — *Algorytmy w C++. Algorytmy grafowe*, Warszawa, 2003, Wydawnictwo RM

## 12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

### OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr hab. inż. Prof PK Zbigniew Kokosiński (kontakt: zk@pk.edu.pl)

### OSOBY PROWADZĄCE PRZEDMIOT

1 dr inż. Zbigniew Kokosiński (kontakt: Zbigniew.Kokosinski@pk.edu.pl)

## 13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejsowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....