

15.	Przedmiot:	I/IM2012/12/15MP						
METODY PROGRAMOWANIA								
Semestr	Liczba tygodni w semestrze	Liczba godzin w tygodniu			Liczba godzin w semestrze			ECTS
		A	C	L	A	C	L	
II	15	2E		2	30		30	6

I. Cele kształcenia

Celem kształcenia jest przekazanie wiedzy z zakresu podstawowych struktur danych i technik programistycznych wykorzystywanych podczas projektowania algorytmów i pisania programów rozwiązujących proste problemy.

II. Wymagania wstępne

Zakres szkoły średniej.

III. Efekty kształcenia i szczegółowe treści kształcenia

Efekty kształcenia, jakie student osiągnie po ukończeniu przedmiotu opisane są w zakresie wiedzy, umiejętności i postaw.

Efekty kształcenia semestr II		Kierunkowe
EK1	Znać podstawowe struktury danych i techniki programistyczne wykorzystywane podczas pisania programów.	K_W14
EK2	Potrafić zaprojektować i napisać program realizujący wyznaczone zadanie, wykorzystujący poznane struktury danych i techniki programistyczne.	K_U15; K_U21 K_U22

Metody i kryteria oceny				
EK 1	Znać podstawowe struktury danych i techniki programistyczne wykorzystywane podczas pisania programów.			
Metody oceny	egzamin pisemny, egzamin ustny, zadanie domowe, zaliczenie ćwiczeń, laboratoriów/ symulatorów, sprawozdanie/ raport, projekt, prezentacja, sprawdziany i prace kontrolne w semestrze			
Kryteria/Ocena	2	3	3,5-4	4,5-5
Kryterium1 Struktury danych.	Nie posiada podstawowej wiedzy z zakresu struktur danych, nie jest w stanie przytoczyć przykładów nawet po uzyskaniu pomocy.	Posiada podstawową wiedzę o najważniejszych strukturach danych.	Posiada podstawową wiedzę o wszystkich przedstawionych w trakcie zajęć strukturach danych, potrafi je scharakteryzować.	Posiada wiedzę o przedstawionych w trakcie zajęć technikach programistycznych, potrafi je scharakteryzować oraz wskazać najlepsze do realizacji przedstawionych problemów.
Kryterium2 Techniki programistyczne.	Nie posiada podstawowej wiedzy o powszechnie stosowanych technikach programistycznych, nie jest w stanie przytoczyć przykładów nawet po uzyskaniu pomocy.	Posiada podstawową wiedzę o najważniejszych technikach programistycznych.	Posiada podstawową wiedzę o wszystkich przedstawionych w trakcie zajęć technikach programistycznych, potrafi je scharakteryzować	Posiada wiedzę o przedstawionych w trakcie zajęć technikach programistycznych, potrafi je scharakteryzować oraz wskazać najlepsze do realizacji przedstawionych problemów
EK 2	Potrafić zaprojektować i napisać program realizujący wyznaczone zadanie, wykorzystujący poznane struktury danych i techniki programistyczne.			
Metody oceny	egzamin pisemny, egzamin ustny, zadanie domowe, zaliczenie ćwiczeń, laboratoriów/ symulatorów, sprawozdanie/ raport, projekt, prezentacja, sprawdziany i prace kontrolne w semestrze			
Kryteria/Ocena	2	3	3,5-4	4,5-5
Kryterium1 Algorytmizacja.	Popelnia znaczne błędy przy tworzeniu najprostszych algorytmów, nawet z pomocą nie potrafi wskazać metod programistycznych potencjalnie użytecznych do rozwiązania przedstawionego problemu.	Potrafi tworzyć algorytmy wykorzystujące wskazane struktury danych i metody programistyczne, może popelniać drobne błędy.	Potrafi samodzielnie tworzyć poprawne algorytmy wykorzystujące poznane struktury danych i metody programistyczne, może popelniać drobne błędy.	Potrafi samodzielnie i swobodnie projektować i tworzyć algorytmy wykorzystujące dowolne użyteczne struktury danych i metody programistyczne, potrafi uzasadnić trafność wyboru.
Kryterium2	Nie potrafi wykorzy-	Potrafi pisać proste	Potrafi samodzielnie	Potrafi samodzielnie i



Programowanie	stać żadnego z dostępnych środowisk programistycznych do napisania najprostszego programu, nawet z pomocą prowadzącego zajęcia.	programy w jednym z dostępnych środowisk programistycznych, wykorzystujące wskazane algorytmy i struktury danych, może popęlniać drobne błędy.	pisać programy wykorzystujące poznane algorytmy, struktury danych i metody programistyczne, może popęlniać drobne błędy.	swobodnie projektować i tworzyć programy wykorzystujące dowolne użyteczne algorytmy, struktury danych i metody programistyczne, potrafi uzasadnić trafność wyboru.
---------------	---	--	--	--

Szczegółowe treści kształcenia

SEMESTR II	METODY PROGRAMOWANIA	AUDYTORYJNE	30 GODZ.
------------	----------------------	-------------	----------

1. Rekurencja:
 - 1.1. rekurencyjne wyrażanie pojęć
 - 1.2. zastosowania i implementacja
 - 1.3. dowodzenie poprawności procedur rekurencyjnych
2. Programowanie z nawrotami:
 - 2.1. przeszukiwanie pełnej przestrzeni stanów
 - 2.2. ucinanie rekursji
3. Metoda *dziel i rządź*:
 - 3.1. metoda inkrementacyjna
 - 3.2. podział binarny
4. Dynamiczne struktury danych:
 - 4.1. typy wskaźnikowe
 - 4.2. wskaźnikowa realizacja list
 - 4.3. podstawowe operacje na listach
 - 4.4. listy jednokierunkowe, dwukierunkowe i cykliczne
 - 4.5. atrapy i strażnicy
5. Liniowe struktury danych: stopy i kolejki:
 - 5.1. implementacja tablicowa i listowa
 - 5.2. implementacja grafu za pomocą list sąsiedztwa
 - 5.3. algorytmy DFS i BFS
6. Drzewa:
 - 6.1. implementacja drzew dowolnego rzędu
 - 6.2. drzewa binarne
 - 6.3. obiegi drzew
 - 6.4. konwersja wyrażeń z postaci infiksowej na prefiksową i postfiksową (ONP)
7. Programowanie zachłanne:
 - 7.1. algorytm Huffmana
8. Metoda spamiętywania:
 - 8.1. programowanie dynamiczne
 - 8.2. problem plecakowy
 - 8.3. optymalne mnożenie wielu macierzy

SEMESTR II	METODY PROGRAMOWANIA	LABORATORYJNE	30 GODZ.
------------	----------------------	---------------	----------

1. Wykorzystanie rekurencji do rozwiązywania problemów algorytmicznych
2. Programowanie z nawrotami na przykładzie "problemu hetmanów"
3. Metoda *dziel i zwyciężaj* na przykładzie wyszukiwania przez podział binarny oraz algorytmu QuickSort
4. Praktyczna implementacja list jedno i dwukierunkowych oraz cyklicznych
5. Implementacja liniowych struktur danych: stopy i kolejki
6. Implementacja liniowych struktur danych: grafy
7. Implementacja i zastosowanie drzew dowolnego rzędu
8. Obiegi drzew i przykłady ich zastosowania
9. Programowanie zachłanne na przykładzie algorytmu Huffmana
10. Programowanie dynamiczne: problem plecakowy, optymalne mnożenie wielu macierzy

Bilans nakładu pracy studenta w semestrze II	Godziny	ECTS
Godziny zajęć z bezpośrednim udziałem nauczyciela: wykłady	30	
Godziny zajęć z bezpośrednim udziałem nauczyciela, o charakterze praktycznym: ćwiczenia, laboratoria, symulatory, zajęcia projektowe	30	



Godziny zajęć z bezpośrednim udziałem nauczyciela: udział w konsultacjach, zaliczeniach / egzaminach poza godz. zajęć dydaktycznych	4	
Własna praca studenta, w tym: przygotowanie do ćwiczeń, laboratoriów, symulatorów, w tym wykonanie sprawozdań, zadań	20	
Własna praca studenta: realizacja zadań projektowych	10	
Własna praca studenta: przygotowanie do zaliczenia, egzaminu	30	
Łączny nakład pracy	124	6
Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli:	64	3
Nakład pracy związany z zajęciami o charakterze praktycznym:	90	3

Zaliczenie przedmiotu

Wszystkie rodzaje zajęć z danego przedmiotu, odbywane w jednym semestrze, podlegają łącznemu zaliczeniu. Ocena z przedmiotu wynika z oceny poszczególnych zajęć, i oceny ewentualnego egzaminu i jest obliczana zgodnie z podanymi zasadami (średnia ważona): A/(E) 40%, C 30% L 30%; A/ (E) 40%, L 60%; A/(E) 40%, C 20%, L 20%, P 20%.

Ocena niedostateczna z zaliczenia którejkolwiek formy przedmiotu w semestrze powoduje niezaliczenie przedmiotu.

Zaliczenie przedmiotu w semestrze powoduje przyznanie studentowi liczby punktów ECTS przypisanej temu przedmiotowi.

IV. Literatura podstawowa

1. Cormen T.H., Leiserson C., Rivest R.L., *Wprowadzenie do algorytmiki*, Wydawnictwa Naukowo-Techniczne 2004.
2. Wirth N., *Algorytmy+Struktury danych=Programy*, Wydawnictwa Naukowo-Techniczne 2001.
3. <http://wazniak.mimuw.edu.pl>

V. Literatura uzupełniająca

1. Knuth D.E., *Sztuka programowania komputerów*, Wydawnictwa Naukowo-Techniczne 2002.
2. Spolsky J., *Sztuka pisania oprogramowania. Wybór i redakcja Joel Spolsky*. 2007.

VI. Prowadzący przedmiot

Koordynator przedmiotu		
dr inż. Paweł Banaś	p.banas@am.szczecin.pl	ZITM
Pozostałe osoby prowadzące zajęcia		