



KARTA MODUŁU / KARTA PRZEDMIOTU

Kod modułu	
Nazwa modułu	Podstawy Programowania 2
Nazwa modułu w języku angielskim	Introduction to Programming 2
Obowiązuje od roku akademickiego	2015/2016

A. USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	I stopień (I stopień / II stopień)
Profil studiów	ogólnoakademicki (ogólnoakademicki / praktyczny)
Forma i tryb prowadzenia studiów	stacjonarne (stacjonarne / niestacjonarne)
Specjalność	bez specjalności
Jednostka prowadząca moduł	Zakład Informatyki
Koordinator modułu	Arkadiusz Chrobot
Zatwierdził:	

B. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	podstawowy (podstawowy / kierunkowy / inny HES)
Status modułu	obowiązkowy (obowiązkowy / nieobowiązkowy)
Język prowadzenia zajęć	polski
Usytuowanie modułu w planie studiów - semestr	II
Usytuowanie realizacji przedmiotu w roku akademickim	semestr letni (semestr zimowy / letni)
Wymagania wstępne	Podstawy programowania 1 (kody modułów / nazwy modułów)
Egzamin	nie (tak / nie)
Liczba punktów ECTS	5

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	inne
w semestrze	30		30	15	



C. EFEKTY KSZTAŁCENIA I METODY SPRAWDZANIA EFEKTÓW KSZTAŁCENIA

Cel modułu	Celem modułu jest zapoznanie studenta z dynamicznym zarządzaniem pamięcią, abstrakcyjnymi strukturami danych, rekurencją oraz praktycznymi aspektami ich wykorzystania w programowaniu. (3-4 linijki)
-------------------	--

Symbol efektu	Efekty kształcenia	Forma prowadzenia zajęć (w/c/l/p/inne)	odniesienie do efektów kierunkowych	odniesienie do efektów obszarowych
W_01	Student zna zagadnienia związane ze wskaźnikami i dynamicznym przydziałem pamięci.	W	KW_06, KW_07	T1A_W07
W_02	Student zna zagadnienia związane z prostymi abstrakcyjnymi strukturami danych, jak listy i drzewa binarne.	W	KW_06, KW_07	T1A_W07, T1A_W04
W_03	Student zna zagadnienia związane z praktycznymi aspektami stosowania techniki rekurencji w programowaniu.	W	KW_06, KW_07	T1A_W07, T1A_W04
U_01	Student potrafi zastosować dynamiczny przydział pamięci do budowy abstrakcyjnych struktur danych.	L,P	K_U12, K_U13	T1A_U09, T1A_U14, T1A_U16
U_02	Student potrafi tworzyć programy z zastosowaniem rekurencji.	L,P	K_U12, K_U13, K_U19	T1A_U09, T1A_U14, T1A_U16
K_01	Umiejętność pracy w zespole.	P	K_K03, K_U02	T1A_U02, T1A_K04

Treści kształcenia:

1. Treści kształcenia w zakresie wykładu

Nr wykładu	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1.	Wskaźniki, dynamiczny przydział pamięci	W_01
2.	Stos i jego zastosowania	W_01, W_02
3.	Rekurencja, technika dziel i zwyciężaj	W_03
4.	Kolejki i ich zastosowania	W_01, W_02
5. - 6.	Jednokierunkowa lista liniowa, jej zastosowania i związki z rekurencją	W_01, W_02, W_03
7.	Dwukierunkowa lista liniowa i jej zastosowania	W_01, W_02
8.	Dwukierunkowa lista cykliczna	W_01, W_02
9.-10.	Drzewa BST i ich zastosowania	W_01, W_02, W_03
11.	Algorytm Quicksort i Heapsort	W_03
12.	Grafy, ich reprezentacje i zastosowania	W_01, W_02
13.	Podstawowe algorytmy grafowe (DFS i BFS)	W_01, W_02,



14.	Algorytmy z nawrotami	W_03 W_01, W_02, W_03
-----	-----------------------	--------------------------------

2. Treści kształcenia w zakresie ćwiczeń

Nr zajęć ćwicz.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
-----------------	--------------------	---

3. Treści kształcenia w zakresie zadań laboratoryjnych

Nr zajęć lab.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1. - 2.	Wskaźniki i dynamiczny przydział pamięci	U_01
3.	Stos i jego zastosowania	U_01
4.	Rekurencja i jej zastosowania	U_02
5.	Kolejki i ich zastosowania	U_01
6.	Jednokierunkowa lista liniowa i jej zastosowania	U_01, U_02
7.	Dwukierunkowa lista liniowa i jej zastosowania	U_01
9.	Dwukierunkowa lista cykliczna i jej zastosowania	U_01
10 - 11.	Drzewa BST i ich zastosowania	U_01, U_02
12.	Algorytmy z nawrotami	U_01, U_02
13. - 14.	Grafy i ich zastosowania	U_01, U_02

4. Charakterystyka zadań projektowych

Zadania projektowe polegają na zaprojektowaniu, stworzeniu i udokumentowaniu programu komputerowego. Tematyka problemu rozwiązywanego przez program jest dowolna, ale zadanie jest tak sformułowane, aby studenci w trakcie jego realizacji korzystali z technik programowania i konstrukcji języka wysokiego poziomu prezentowanych na wykładzie. Zadania są realizowane w grupach dwuosobowych. Studenci powinni wykazać się samodzielnością w realizacji projektu, a ponadto zaproponować sposób sprawdzenia poprawności przyjętych rozwiązań, oraz udokumentować kod źródłowy programu.

5. Charakterystyka zadań w ramach innych typów zajęć dydaktycznych

Metody sprawdzania efektów kształcenia

Symbol efektu	Metody sprawdzania efektów kształcenia (sposób sprawdzenia, w tym dla umiejętności – odwołanie do konkretnych zadań projektowych, laboratoryjnych, itp.)
W_01	Kolokwium pisemne
W_02	Kolokwium pisemne
W_03	Kolokwium pisemne
W_04	Kolokwium pisemne
U_01	Zadania laboratoryjne, kolokwium laboratoryjne (przewidziane na zajęcia nr 8 i nr 15), zadanie projektowe.
U_02	Zadania laboratoryjne, kolokwium laboratoryjne (przewidziane na zajęcia nr 8 i nr 15), zadanie projektowe.
U_03	Zadania laboratoryjne, kolokwium laboratoryjne (przewidziane na zajęcia nr 8 i nr 15),



	zadanie projektowe.
K_01	Zdanie projektowe.

D. NAKŁAD PRACY STUDENTA

Bilans punktów ECTS		
	Rodzaj aktywności	obciążenie studenta
1	Udział w wykładach	30
2	Udział w ćwiczeniach	
3	Udział w laboratoriach	30
4	Udział w konsultacjach (2-3 razy w semestrze)	2
5	Udział w zajęciach projektowych	
6	Konsultacje projektowe	15
7	Udział w egzaminie	
8		
9	Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela akademickiego	77 <i>(suma)</i>
10	Liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczyciela akademickiego <i>(1 punkt ECTS=25-30 godzin obciążenia studenta)</i>	3
11	Samodzielne studiowanie tematyki wykładów	10
12	Samodzielne przygotowanie się do ćwiczeń	
13	Samodzielne przygotowanie się do kolokwium	10
14	Samodzielne przygotowanie się do laboratoriów	10
15	Wykonanie sprawozdań	
15	Przygotowanie do kolokwium końcowego z laboratorium	10
17	Wykonanie projektu lub dokumentacji	15
18	Przygotowanie do egzaminu	
19		
20	Liczba godzin samodzielnej pracy studenta	55 <i>(suma)</i>
21	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy <i>(1 punkt ECTS=25-30 godzin obciążenia studenta)</i>	2
22	Sumaryczne obciążenie pracą studenta	132
23	Punkty ECTS za moduł <i>1 punkt ECTS=25-30 godzin obciążenia studenta</i>	5
24	Nakład pracy związany z zajęciami o charakterze praktycznym <i>Suma godzin związanych z zajęciami praktycznymi</i>	80
25	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym <i>1 punkt ECTS=25-30 godzin obciążenia studenta</i>	3



E. LITERATURA

Wykaz literatury	<ol style="list-style-type: none">1. Brian W. Kernighan, Denis M. Ritchie, „Język ANSI C. Programowanie”, Wydanie 2, Helion, Gliwice 20102. Piotr Wróblewski, „Algorytmy, struktury danych i techniki programowania”, Helion, Gliwice 19973. Jon Bentley „Perełki oprogramowania”, WNT, Warszawa 19924. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, „Algorytmy i struktury danych”, Helion 20035. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest „Wprowadzenie do algorytmów”, WNT, Warszawa 19986. Donald E. Knuth „Sztuka programowania”, WNT, Warszawa 20027. Steven S. Skiena "The Algorithm Design Manual", Springer-Verlag, Londyn, 20088. Stephen Prata „Język C. Szkoła programowania”, Helion, Gliwice, 20169. Zed A. Shaw „Programowanie w C. Sprytne podejście do trudnych zagadnień, których wolałbyś unikać (takich jak język C)”, Helion, Gliwice, 2016
Witryna WWW modułu/przedmiotu	(strona dostępna w semestrach letnich) http://achilles.tu.kielce.pl/Members/achrobot/pp.html