



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	E-I-EN-03-s2
Nazwa przedmiotu	Podstawy Programowania 2
Nazwa przedmiotu w języku angielskim	Fundamentals of Programming 2
Obowiązuje od roku akademickiego	2019/20

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	
Poziom kształcenia	
Profil studiów	
Forma i tryb prowadzenia studiów	
Zakres	
Jednostka prowadząca przedmiot	
Koordynator przedmiotu	dr inż. Arkadiusz Chrobot
Zatwierdził	Dziekan Wydziału Elektrotechniki Automatyki i Informatyki Dr hab. inż. Antoni Różowicz, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	
Status przedmiotu	
Język prowadzenia zajęć	
Usytuowanie modułu w planie studiów - semestr	
Wymagania wstępne	Podstawy Programowania 1
Egzamin (TAK/NIE)	
Liczba punktów ECTS	

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	Inne
-------------------------	--------	-----------	--------------	---------	------

Liczba godzin w semestrze	30		30	15	0
---------------------------	----	--	----	----	---

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Student zna zagadnienia związane ze wskaźnikami i dynamicznym przydziałem pamięci.	INF1_W06 INF1_W07
	W02	Student zna zagadnienia związane z prostymi abstrakcyjnymi strukturami danych, jak listy, drzewa binarne, grafy.	INF1_W06 INF1_W07 INF1_W11
	W03	Student zna zagadnienia związane z praktycznymi aspektami stosowania techniki rekurencji w programowaniu.	INF1_W06 INF1_W07 INF1_W11
Umiejętności	U01	Student potrafi zastosować dynamiczny przydział pamięci do budowy abstrakcyjnych struktur danych.	INF1_U01 INF1_U13 INF1_U14
	U02	Student potrafi tworzyć programy z zastosowaniem rekurencji.	INF1_U01 INF1_U13 INF1_U14
	U03	Student potrafi projektować i tworzyć programy komputerowe pracując w zespole.	INF1_U01 INF1_U13 INF1_U14 INF1_U32
Kompetencje społeczne	K01	Student potrafi efektywnie komunikować się z osobami, z którymi współpracuje w zespole.	INF1_K03

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	1. Wskaźniki, dynamiczny przydział pamięci
	2. Stos i jego zastosowania
	3. Rekurencja, technika dziel i zwyciężaj
	4. Kolejki i ich zastosowania
	5. Jednokierunkowa lista liniowa, jej zastosowania i związki z rekurencją
	6. Dwukierunkowa lista liniowa i jej zastosowania
	7. Dwukierunkowa lista cykliczna
	8. Drzewa BST i ich zastosowania
	9. Algorytm Quicksort i Heapsort
	10. Grafy, ich reprezentacje i zastosowania
	11. Podstawowe algorytmy grafowe (DFS i BFS)
	12. Algorytmy z nawrotami
laboratorium	1. Wskaźniki i dynamiczny przydział pamięci
	2. Stos i jego zastosowania
	3. Rekurencja i jej zastosowania
	4. Kolejki i ich zastosowania
	5. Jednokierunkowa lista liniowa i jej zastosowania
	6. Dwukierunkowa lista liniowa i jej zastosowania
	7. Dwukierunkowa lista cykliczna i jej zastosowania
	8. Drzewa BST i ich zastosowania
	9. Grafy i ich zastosowania
	10. Algorytmy z nawrotami

projekt	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. Studenci powinni wykazać się samodzielnością w realizacji projektu, a ponadto zaproponować sposób sprawdzenia poprawności przyjętych rozwiązań.
---------	--

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			x			x
W02			x			x
W03			x			x
U01			x			x
U02			x			x
U03			x			x
K01				x	x	

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład		Uzyskanie pozytywnej oceny z zajęć laboratoryjnych lub co najmniej 50% punktów z kolokwium.
laboratorium		Uzyskanie co najmniej 50% z kolokwiumów cząstkowych i realizowanych zadań.
projekt		Uzyskanie co najmniej 50% z wykonanego programu i jego dokumentacji, oraz obrony całości projektu.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	30		30	15		h
3.	Inne (konsultacje, egzamin)*	2		2	1		h
4.	Razem przy bezpośrednim udziale nauczyciela akademickiego	80					h
5.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	3,2					ECTS
6.	Liczba godzin samodzielnej pracy studenta	45					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	1,8					ECTS
8.	Nakład pracy związany z zajęciami o charakterze praktycznym	45					h

9.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	2,81	ECTS
10.	Sumaryczne godzinowe obciążenie pracą studenta	125	h
11.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	5	

LITERATURA

1. Brian W. Kernighan, Denis M. Ritchie, „Język ANSI C. Programowanie”, Wydanie 2, Helion, Gliwice 2010
2. Piotr Wróblewski, „Algorytmy, struktury danych i techniki programowania”, Helion, Gliwice 1997
3. Jon Bentley „Perełki oprogramowania”, WNT, Warszawa 1992
4. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, „Algorytmy i struktury danych”, Helion 2003
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest „Wprowadzenie do algorytmów”, WNT, Warszawa 1998
6. Donald E. Knuth „Sztuka programowania”, WNT, Warszawa 2002
7. Steven S. Skiena "The Algorithm Design Manual", Springer-Verlag, Londyn, 2008
8. Stephen Prata „Język C. Szkoła programowania”, Helion, Gliwice, 2016
9. Zed A. Shaw „Programowanie w C. Sprytne podejście do trudnych zagadnień, których wolałbyś unikać (takich jak język C)”, Helion, Gliwice, 2016