



WYŻSZA SZKOŁA BANKOWA W POZNANIU
WYDZIAŁ FINANSÓW I BANKOWOŚCI
Rok akademicki 2018/2019

KARTA PRZEDMIOTU

Stopień studiów:

studia pierwszego stopnia (inżynierskie)

Kierunek studiów:

Informatyka inżynierska

Semestr: 2

Kod przedmiotu: P1111P22

Status przedmiotu: obligatoryjny, bez możliwości wyboru

Forma studiów: stacjonarny/niestacjonarny

Język wykładowy: polski

Forma zajęć	liczba godzin		Forma zaliczenia	Punkty ECTS (wpisuje odpowiedni dział Uczelni)
	S	N		

Nazwa przedmiotu:

Algorytmy i struktury danych

W:	30	16		Ogółem dla przedmiotu		5
Ć:				Punkty ECTS za zajęcia wymagające bezpośredniego udziału nauczyciela akademickiego	Punkty ECTS W	2/1,5
L:	30	16	zo		Punkty ECTS Ć, L, P, E	2
P:	10	10				
E:	5	5			Punkty ECTS za pracę własną studenta	1/1,5

Koordynator przedmiotu:

dr inż. Wojciech Complak

Prowadzący przedmiot:

dr inż. Wojciech Complak, dr inż. Maciej Piernik, dr Marek Gałązka, dr Mariusz Nogala, mgr Kamil Reczek, mgr inż. Jerzy Wiśniewski

1. Wymagania wstępne:

1.1. W zakresie wiedzy student:

- posiada wiedzę z matematyki w zakresie szkoły średniej na poziomie IV etapu edukacyjnego w zakresie podstawowym
- posiada umiejętność tworzenia programów w języku C#

1.2. W zakresie umiejętności student:

- posiada umiejętność efektywnego samokształcenia w dziedzinach związanych z informatyką.

1.3. W zakresie kompetencji społecznych student:

- prezentuje nabytą wiedzę
- ma świadomość konieczności poszerzania swoich kompetencji.

2. Założenia i cele kształcenia:

Celem przedmiotu jest zapoznanie studentów z zasadami projektowania, implementacji i wykorzystywania algorytmów. Studenci poznają teoretyczne i praktyczne aspekty podstawowych algorytmów i struktur danych wykorzystywanych w informatyce, wykorzystując imperatywny język programowania uczą się konstruowania, implementacji, testowania i analizy efektywności.

3. szczełowe efekty kształcenia przedmiotu:

Kod efektu	Nazwa efektu kształcenia	Odniesienie do efektów kształcenia programu studiów
------------	--------------------------	---

WIEDZA		
AiSD_W01	przywołuje pojęcie algorytmu, wybiera metodę zapisu algorytmu, wartościuje algorytm, buduje algorytm	K_W01, K_W03, K_W09, K_W12
AiSD_W02	wyjaśnia pojęcie rekurencji, klasyfikuje algorytm rekurencyjny i iteracyjny, wybiera metodę implementacji, ocenia implementację	K_W01, K_W03, K_W09, K_W12
AiSD_W03	definiuje problem sortowania, wartościuje algorytm, wybiera algorytm	K_W01, K_W03, K_W09, K_W12
AiSD_W04	definiuje rekurencyjne struktury danych, wybiera struktury danych	K_W01, K_W03, K_W09, K_W12
AiSD_W05	dobiera reprezentację numeryczną, ocenia zbieżność algorytmu	K_W01, K_W03, K_W09, K_W12
UMIEJĘTNOŚCI		
AiSD_U01	identyfikuje typ problemu, wyodrębnia problemy elementarne, proponuje właściwe algorytmy, buduje implementację	K_U04
AiSD_U02	wyodrębnia elementy problemu, adaptuje implementacje algorytmu, konstruuje rozwiązanie z wykorzystaniem przykładów, wyjaśnia cechy rozwiązania	K_U01, K_U04
KOMPETENCJE SPOŁECZNE		
AiSD_K01	identyfikuje rozwiązywany problem, czyta literaturę naukowo-dydaktyczną, kreatywnie rozwija przykłady, integruje informacje z różnych źródeł	K_K01, K_K02

4. Szczegółowy program zajęć:

Nr	Treści programowe	Efekty kształcenia przedmiotu (kod)	Forma zajęć/liczba godzin	
			S	NS
1	Wprowadzenie do algorytmiki: problem a algorytm, metody zapisu algorytmów, ocena wydajności, klasyfikacja złożoności problemów i algorytmów.	AiSD_W01 AiSD_U01 AiSD_K01	W-2 L-2 P-0 E-0	W-2 L-2 P-0 E-0
2	Pojęcie rekurencji, przykłady algorytmów rekurencyjnych, ocena i porównanie wydajności algorytmów iteracyjnych i rekurencyjnych rozwiązujących ten sam problem, przekształcanie do postaci iteracyjnej.	AiSD_W01 AiSD_W02 AiSD_U01	W-2 L-2 P-0 E-1	W-2 L-2 P-0 E-1
3	Rekurencyjne struktury danych: listy, kolejki, drzewa.	AiSD_W04 AiSD_U01	W-6 L-6 P-3 E-1	W-2 L-2 P-3 E-1
4	Algorytmy sortowania: definicja problemu sortowania, miary efektywności, metody proste, metody ulepszone, zasady doboru metod, ocena wydajności, sortowanie zewnętrzne.	AiSD_W01 AiSD_W03 AiSD_U01 AiSD_U02 AiSD_K01	W-4 L-4 P-3 E-1	W-2 L-2 P-3 E-1
5	Grafy i algorytmy grafowe, metody reprezentacji grafów, przeszukiwanie grafów, podstawowe problemy grafowe i ich znaczenie praktyczne.	AiSD_W01 AiSD_W04 AiSD_U01 AiSD_U02	W-6 L-6 P-4 E-1	W-2 L-2 P-4 E-1
6	Algorytmy wyszukiwania, przeszukiwanie tekstów.	AiSD_W01 AiSD_U01 AiSD_U02 AiSD_K01	W-2 L-2 P-0 E-0	W-2 L-2 P-0 E-0
7	Zaawansowane techniki programowania, algorytmy zachłanne, programowanie dynamiczne, kompresja i szyfrowanie danych.	AiSD_W01 AiSD_W04	W-6 L-6	W-2 L-2

		AiSD_U01 AiSD_U02 AiSD_K01	P-0 E-1	P-0 E-1
8	Algorytmy numeryczne, specyfika obliczeń numerycznych, typy danych, metody konstruowania algorytmów, optymalizacja, aproksymacja.	AiSD_W01 AiSD_W05 AiSD_U01 AiSD_K01	W-2 L-2 P-0 E-0	W-2 L-2 P-0 E-0
Liczba godzin łączna			75	47

5. Metody dydaktyczne:

Wykłady,
realizacja eksperymentów obliczeniowych,
analiza materiałów dydaktycznych przekazywanych na platformie Moodle.

6. Praca indywidualna studenta:

Student zapoznaje się z materiałami przedstawianymi w ramach jednostek wykładów oraz z przykładami udostępnianymi na platformie Moodle ilustrującymi typowe rozwiązania problemów napotykanym w konstrukcji aplikacji użytkowych. Ma rozwiązać zadania dotyczące syntezy i analizy programów, oceny poprawności i efektywności zaproponowanych implementacji. Łączne obciążenie powinno wynieść około 40 godzin.

7. Literatura obowiązkowa / podstawowa:

1. Wróblewski P., *Algorytmy, struktury danych i techniki programowania*. Wydanie IV., Helion, 2010
2. Wirth N., *Algorytmy + struktury danych = programy*. Wydawnictwa Naukowo-Techniczne, 2004

8. Literatura uzupełniająca (w tym inne pomoce naukowe, np. akty prawne, orzecznictwo sądowe, artykuły naukowe i itp.):

1. Aho A. V., Hopcroft J. E., Ullman J. D., *Algorytmy i struktury danych*, Helion, 2003
2. Heineman G., Pollice G., Selkow S., *Algorytmy. Almanach.*, Helion, 2012
3. Sedgewick R., Wayne K., *Algorytmy. Wydanie IV.*, Helion, 2012
4. Fortuna Z., Macukow B., Wasowski J., *Metody numeryczne*, WNT, 1995
5. Bjorck A., Dahlquist G., *Metody numeryczne*, Warszawa, PWN, 1987

9. Sposób zaliczenia:

W czasie semestru studenci zbierają punkty, które są podstawy zaliczenia. Maksymalnie można zdobyć 100 punktów, do uzyskania zaliczenia wymagane jest co najmniej 50 punktów. Punkty są przydzielane na podstawie:

- sprawdzianów wstępnych dotyczących obowiązującego materiału przedstawionego na poprzedzających wykładach; w trakcie semestru odbędzie się 5 sprawdzianów; zaliczenie (uzyskanie > 50% punktów) z co najmniej 3 sprawdzianów pozwala uzyskać 40 punktów,
- samodzielnie wykonywanych projektów programistycznych oddawanych w wyznaczonych terminach; opóźnienie oddania projektu wiąże się z obniżeniem maksymalnej liczby punktów jaką można otrzymać za projekt w wymiarze 10% punktów na każdy tydzień opóźnienia; w trakcie zaliczania projektu student ma obowiązek wykazania się wiedzą dotyczącą przedstawianego projektu; za wykonanie poszczególnych projektów uzyskać można:

I projekt: 0 ÷ 10 punktów,

II projekt: 0 ÷ 10 punktów,

III projekt: 0 ÷ 30 punktów,

- kolokwium końcowego (na platformie Moodle), którego zakres obejmuje teoretyczne podstawy przedstawianych w trakcie kursu zagadnień, z kolokwium można uzyskać 1 ÷ 10 punktów,
- ostateczna ocena wyliczana jest na podstawie sumy uzyskanych punktów:

punkty	ocena
< 50	2.0
50 ÷ 59	3.0
60 ÷ 69	3.5
70 ÷ 79	4.0
80 ÷ 89	4.5
> 89	5.0

- zaliczenie poprawkowe oraz w ramach ITS można uzyskać na podstawie rozliczenia projektów programistycznych i kolokwium (na platformie Moodle) w wyznaczonym przez Dziekana terminie.

10. Opis sposobu sprawdzenia osiągnięcia efektów kształcenia:

Efekt kształcenia	Metoda sprawdzenia
-------------------	--------------------

	Kolokwium zaliczeniowe	Projekty programistyczne
AiSD_W01, AiSD_W02, AiSD_W03, AiSD_W04, AiSD_W05, AiSD_U01, AiSD_U02, AiSD_K01	x	x