

KARTA PRZEDMIOTU (SYLABUS)

Opis przedmiotu

Kod przedmiotu		Nazwa przedmiotu	PODSTAWY PROGRAMOWANIA	
IT/P/I/NST/B _I -1			BASICS PROGRAMMING	
Język wykładowy		język polski		
Rok akademicki		2019/2020		
Kierunek		Informatyka techniczna		
w zakresie				
Poziom studiów		studia pierwszego stopnia		
Profil studiów		praktyczny		
Forma studiów		studia niestacjonarne		
Semestr / semestry		Semestr pierwszy -zimowy		
Przynależność do grupy zajęć		B – grupa przedmiotów kierunkowych		
Status przedmiotu		obowiązkowy		
Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS		Forma zajęć	Liczba godzin zajęć dydaktycznych	Liczba punktów ECTS
		Wykład	25 [h]	7 ECTS
		Ćwiczenia	30 [h]	
		
Powiązanie przedmiotu	z profilem studiów	kształtuje umiejętności praktyczne		3 ECTS
	z uprawnieniami	służy do zdobywania przez studenta kompetencji inżynierskich		7 ECTS
	z dyscypliną	informatyka techniczna i telekomunikacja		7 ECTS
Forma nauczania		tradycyjna - zajęcia zorganizowane w Uczelni		
Wymagania wstępne				
Jednostka prowadząca		Katedra Informatyki		
Koordynator		dr Artur Bartoszewski		
Osoby prowadzące		dr Artur Bartoszewski		
Adres strony internetowej pjo		www.wim.uniwersytetradom.pl		
Adres e-mail, telefon koordynatora		artur.bartoszewski@uthrad.pl		

EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE, REALIZACJA ZAJĘĆ DYDAKTYCZNYCH, WERYFIKACJA EFEKTÓW UCZENIA SIĘ

Cel kształcenia:	Poznanie możliwie szerokiego zakresu zagadnień podstawowych związanych z programowaniem komputerów, zwłaszcza konstrukcji językowych opisujących struktury danych i algorytmy oraz związanych z nimi technik tworzenia programów. Ćwiczenia o charakterze laboratoryjnym mają na celu opanowanie umiejętności formułowania problemu w języku C i C++ oraz tworzenia i uruchamiania programów.
Treści programowe:	<p>Wykład</p> <p>Pojęcia podstawowe: problem, algorytm, program, metody zapisu algorytmów. [1h] W1</p> <p>Reprezentacja danych w komputerze, dokładność obliczeń zmiennopozycyjnych. [1h]W1</p> <p>Proste modele maszyny cyfrowej i języka maszynowego, zapis operacji w języku maszynowym, dane, program, proces sekwencyjny. [1h]W1</p> <p>Pojęcie iteracji i rekurencji, [2h] W1</p> <p>Przegląd znanych algorytmów: (między innymi: algorytm Euklidesa i Hornera, Sito Erastotenesa, rozwiązywanie równań kwadratowych.). [4h]W1</p> <p>Języki programowania i ich klasyfikacja. Środowiska programistyczne, translatory, kompilacja i interpretacja. Uruchamianie i testowanie programu, problem poprawności i złożoności programu. [2h]W2</p> <p>Podstawowe konstrukcje języka C/C++: identyfikatory, typy standardowe, typy wyliczeniowe, zmienne i stałe, tablice i łańcuchy, wyrażenia, funkcje standardowe, Instrukcje warunkowe i pętle programowe. Rekurencja a iteracja. [6h]W2,</p> <p>Programowanie proceduralne: struktura programu, funkcje, moduły i biblioteki. Zasięg lokalny i globalny. Parametry formalne i aktualne, przekazywanie parametrów do funkcji, przeciążanie funkcji. [4h]W2</p> <p>Praca z tablicami jedno i wielowymiarowymi. Algorytmy sortowania.[2h] W2,</p> <p>Obsługa plików tekstowych [2h] W2</p> <p>Wskaźniki i struktury dynamiczne: listy jednokierunkowe i dwukierunkowe. [2h] W2</p> <p>Ćwiczenia laboratoryjne</p> <p>Studenci opanowują umiejętność doboru struktur danych i właściwych konstrukcji algorytmicznych do rozwiązywanego problemu oraz umiejętność analizy i budowy algorytmu. Studenci zapisują algorytmy w postaci schematów blokowych [4h] U2</p> <p>Studenci poznają wybrane środowiska programistyczne dla języków C i C++ [1h] U3</p> <p>Studenci wykonują programy służące do rozwiązywania problemów dotyczących następujących zagadnień:</p> <ul style="list-style-type: none"> • obliczenia numeryczne, ciągi liczbowe, zmiana reprezentacji liczb (system dwójkowy i szesnastkowy), [2h]U1, U2, • rozgałęzienia w programie, instrukcje wyboru oraz pętle programowe, iteracyjne rozwiązywanie problemów numerycznych (algorytmy działające na ciągach liczbowych, [4h] U1 • operacje na tablicach, tablicowanie funkcji, działania na znakach i łańcuchach znaków, wyznaczanie min/max n elementowego ciągu liczbowego, [4h] U1,U2 • podział programu na funkcje, przekazywanie parametrów do funkcji, przeciążanie funkcji [4h] U1 U2 • rekurencja (silnia, potęga, itp.) [2h]U1,U2 • złożone typy danych (struktury, unie, pola bitowe), sortowania (sortowanie bąbelkowe, łączenie dwóch ciągów uporządkowanych itp.), [2h]U2,U3 • operacje na plikach tekstowych [2h] U1, U2 • wskaźniki, proste listy jednokierunkowe (lista liczb, lista łańcuchów itp.) [2h]U2, U3
Metody dydaktyczne (kształcenia):	Metody podające - wykład informacyjny – W1, W2 Metody programowane z wykorzystaniem komputera – ćwiczenia laboratoryjne- U1,U2, U3, K1, K2 Wszystkie zastosowane metody umożliwiają rozpoznawanie i zaspokajanie indywidualnych potrzeb studentów, w tym studentów niepełnosprawnych oraz indywidualizację toku studiów.
Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej:	Warunkiem zaliczenia przedmiotu jest osiągnięcie wszystkich wymaganych efektów kształcenia określonych dla danego przedmiotu.

	Wykład: Ocena końcowa z egzaminu pisemnego Ćwiczenia laboratoryjne: Sprawdzian pisemny z zakresu algorytmiki oraz sprawdzian praktyczny umiejętności programowania i implementacji algorytmów w języku C++
--	---

Efekty uczenia się dla przedmiotu w odniesieniu do efektów kierunkowych i formy zajęć				Metody weryfikacji efektów uczenia się	
Numer efektu uczenia się	Opis efektów uczenia się dla przedmiotu (PEU) Student, który zaliczył przedmiot (W) zna i rozumie/ (U) potrafi / (K) jest gotów do:	Kierunkowy efekt uczenia się (KEU)	Forma zajęć	Forma weryfikacji (zaliczeń)	Metody sprawdzania i oceny
W1	Ma podstawową wiedzę na temat programowania komputerów, systemów operacyjnych, struktur danych i algorytmów oraz związanych z nimi technik tworzenia i uruchamiania programów	K_WG05	wykład, ćwiczenia laboratoryjne	egzamin, zaliczenie na ocenę	egzamin, kolokwium
W2	Zna składnię i polecenia języka C oraz elementy składni języka C++ (z wyłączeniem programowania obiektowego).	K_WG05	wykład	egzamin, zaliczenie na ocenę	egzamin, kolokwium
U1	Posiada umiejętność budowy, zapisu oraz implementacji algorytmów w językach programowania C i C++	K_UW04 K_UW12	wykład	egzamin, zaliczenie na ocenę	egzamin, kolokwium
U2	Umie dobrać właściwe struktury danych oraz konstrukcje algorytmiczne do rozwiązywanego problemu oraz umie zaprogramować algorytm w języku wysokiego poziomu	K_UW04 K_UW12	wykład, ćwiczenia laboratoryjne	egzamin, zaliczenie na ocenę	egzamin, kolokwium
U3	Ma umiejętność formułowania problemu w języku C oraz tworzenia i uruchamiania programów w wybranym środowisku programistycznym	K_UW04 K_UW12	wykład, ćwiczenia laboratoryjne	egzamin, zaliczenie na ocenę	egzamin, kolokwium
K1	Jest gotów poszerzać swoją wiedzę i doskonalić umiejętności potrzebne w wykonywaniu zawodu programisty	K_KK01	wykład	egzamin	egzamin
K2	Potrafi zaplanować pracę pod kątem zakładanych rezultatów, określić priorytetowe zadania w oparciu o zasady skutecznego działania.	K_KO05	ćwiczenia laboratoryjne	zaliczenie na ocenę	kolokwium
Stopień osiągnięcia kierunkowych efektów uczenia się: K_WG05+++ K_UW04+++ K_UW12++ K_KK01+++ K_KO05+					

Literatura podstawowa, literatura uzupełniająca, pomoce naukowe

Literatura podstawowa:

1. Grębosz J. : Symfonia C++, Programowanie w języku C++ orientowane obiektowo, Wydawnictwo Edition 2000.
2. Jakubczyk K.: Turbo Pascal i Borland C++. Przykłady, Helion, Gliwice 2002,
3. Sokół R. : Microsoft Visual Studio 2012 Programowanie w Ci C++, Helion 2014.
4. Kernighan B. W., Ritchie D. M.: język ANSI C, Wydawnictwo Naukowo Techniczne, Warszawa 1994.

Literatura uzupełniająca:

1. Sysło M.M.: Piramidy, szyszki i inne konstrukcje algorytmiczne, WSiP, Warszawa 1998.
2. Kubiak M.J.: *Programuję w językach Turbo Pascal i C/C++*, Mikom, Warszawa 2001.
3. Wirth N.: *Algorytmy + struktury danych = programy*, WNT, Warszawa 2004.
4. Simon L. D., Drozdek A.: *Struktury danych w języku C*, WNT, Warszawa 1996.

Autorskie materiały dydaktyczne zamieszczone na stronie www.bartoszewski.uthrad.pl

Nakład pracy studenta potrzebny do osiągnięcia zakładanych efektów uczenia się – bilans punktów ECTS			
Udział w zajęciach, aktywność	Obciążenie studenta [h]		
	Inne godz. kontaktowe (IGK)	Zajęcia bez nauczyciela-praca własna studenta (ZBN)	Zajęcia dydaktyczne
Udział w wykładach	X	X	25 [h]
Samodzielne studiowanie tematyki wykładów	X	50 [h]	X
Udział w ćwiczeniach laboratoryjnych	X	X	30[h]
Samodzielne przygotowanie się do ćwiczeń	X	50 [h]	X

Udział w konsultacjach	6 [h]	X	X
Przygotowanie do <i>egzaminu</i>	X	10 [h]	X
Udział w <i>egzaminie</i>	2 [h]	X	X
Sumaryczne obciążenie pracą studenta	8[h]/ 0,5 ECTS	110 [h]/4,5ECTS	55[h]/ 2 ECTS
Punkty ECTS za przedmiot	7 ECTS		

Informacje dodatkowe, uwagi

Studentowi przysługuje jeden termin podstawowy i jeden termin poprawkowy zaliczenia dla każdej formy zajęć. Obecność na zajęciach laboratoryjnych jest obowiązkowa. Obecność na wykładach jest zalecana i może być premiowana. W przypadku zaległości powstałych wskutek nieobecności studenta na zajęciach projektowych student jest zobowiązany do uczestnictwa w zajęciach innej grupy (tzw. odrobienie zajęć) lub wykonania (w przypadku braku możliwości odrobienia) i zaliczenia dodatkowego projektu.

Zgodnie z Regulaminem Studiów UTHRad podstawowym terminem uzyskania zaliczenia jest ostatni dzień zajęć w danym semestrze. Termin zaliczenia poprawkowego (tryb i warunki ustala prowadzący moduł na zajęciach początkowych) nie może być późniejszy niż ostatni termin egzaminu w sesji poprawkowej (dla przedmiotów kończących się egzaminem) lub ostatni dzień trwania semestru (dla przedmiotów niekończących się egzaminem).

Terminy odbywania zajęć: semestr zimowy, zgodnie z rozkładem zajęć

Miejsce odbywania zajęć: ul. Malczewskiego 20A, Radom;