

# KARTA PRZEDMIOTU (SYLABUS)

## Opis przedmiotu

Kod przedmiotu		Nazwa przedmiotu	PODSTAWY PROGRAMOWANIA	
IT/P/I/ST/B <sub>1</sub> -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 stacjonarne		
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	30 [h]	7 ECTS
		Ćwiczenia laboratoryjne	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><b>Wykład</b></p> <p>Pojęcia podstawowe: problem, algorytm, program, metody zapisu algorytmów. [2h] 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. [1h]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. [5h]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.[4h] W2,</p> <p>Obsługa plików tekstowych [2h] W2</p> <p>Wskaźniki i struktury dynamiczne: listy jednokierunkowe i dwukierunkowe. [4h] W2</p> <p><b>Ćwiczenia laboratoryjne</b></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 [6h] 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"> <li>• obliczenia numeryczne, ciągi liczbowe, zmiana reprezentacji liczb (system dwójkowy i szesnastkowy), [4h]U1, U2,</li> <li>• rozgałęzienia w programie, instrukcje wyboru oraz pętle programowe, iteracyjne rozwiązywanie problemów numerycznych (algorytmy działające na ciągach liczbowych, [6h] U1</li> <li>• operacje na tablicach, tablicowanie funkcji, działania na znakach i łańcuchach znaków, wyznaczanie min/max n elementowego ciągu liczbowego, [8h] U1,U2</li> <li>• podział programu na funkcje, przekazywanie parametrów do funkcji, przeciążanie funkcji [4h] U1 U2</li> <li>• rekurencja (silnia, potęga, itp.) [4h]U1,U2</li> <li>• złożone typy danych (struktury, unie, pola bitowe), sortowania (sortowanie bąbelkowe, łączenie dwóch ciągów uporządkowanych itp.), [4h]U2,U3</li> <li>• operacje na plikach tekstowych [4h] U1, U2</li> <li>• wskaźniki, proste listy jednokierunkowe (lista liczb, lista łańcuchów itp.) [4h]U2, U3</li> </ul>
Metody dydaktyczne (kształcenia):	<p>Metody podające - wykład informacyjny – W1, W2</p> <p>Metody programowane z wykorzystaniem komputera – ćwiczenia laboratoryjne- U1,U2, U3, K1, K2</p> <p>Wszystkie zastosowane metody umożliwiają rozpoznawanie i zaspokajanie indywidualnych potrzeb studentów, w tym studentów niepełnosprawnych oraz indywidualizację toku studiów.</p>
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.

	<b>Wykład:</b> Ocena końcowa z egzaminu pisemnego <b>Ćwiczenia laboratoryjne:</b> 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: *np.*: 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 C i 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. Wirth N.: *Algorytmy + struktury danych = programy*, WNT, Warszawa 2004.
3. Simon L. D., Drozdek A.: *Struktury danych w języku C*, WNT, Warszawa 1996.

Autorskie materiały dydaktyczne zamieszczone na stronie [www.bartoszewski.uthrad.pl](http://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	30 [h]
Samodzielne studiowanie tematyki wykładów	X	40 [h]	X
Udział w ćwiczeniach laboratoryjnych	X	X	45 [h]
Samodzielne przygotowanie się do ćwiczeń	X	40 [h]	X
Udział w konsultacjach	6 [h]	X	X

Przygotowanie do <i>egzaminu</i>	X	<b>10 [h]</b>	X
Udział w <i>egzaminie</i>	<b>2 [h]</b>	X	X
Sumaryczne obciążenie pracą studenta	<b>8 [h]/ 0,5 ECTS</b>	<b>90 [h]/3,5 ECTS</b>	<b>75 [h]/ 3 ECTS</b>
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;