



Karta przedmiotu

Nazwa i kod przedmiotu	Basics of computer programming, PG_00045290						
Kierunek studiów	Inżynieria danych						
Data rozpoczęcia studiów	październik 2020 r.	Rok akademicki realizacji przedmiotu			2020/2021		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			angielski		
Semestr studiów	1	Liczba punktów ECTS			5.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Algorytmów i Modelowania Systemów						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	prof. dr hab. inż. Dariusz Dereniowski					
	Prowadzący zajęcia z przedmiotu	mgr inż. Robert Ostrowski dr inż. Tytus Pikies prof. dr hab. inż. Dariusz Dereniowski dr Marcin Jurkiewicz					
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	15.0	20.0	0.0	65
	W tym liczba godzin zajęć na odległość: 0.0 Adresy na platformie eNauczanie:						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach	Praca własna studenta	RAZEM		
	Liczba godzin pracy studenta	65	10.0	50.0	125		
Cel przedmiotu	Celem przedmiotu jest wprowadzenie do programowania komputerów, którego głównym celem jest nauczenie studenta rozwiązywania problemów programistycznych i pisania programów w języku C/C++.						
Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu			Sposób weryfikacji i oceny efektu		
	[K6_U01] programuje w językach programowania proceduralnego, obiektowego, funkcjonalnego i w logice, koduje programy na poziomie instrukcji procesora, uruchamia i testuje programy	Student programuje w języku programowania proceduralnego, uruchamia i testuje programy.			[SU1] Ocena realizacji zadania		
	[K6_W05] zna i rozumie modele programowania i ewolucję związanych z nimi języków. zna metody analizowania i projektowania systemów informatycznych i wykorzystywane w nich języki modelowania, a także podstawowe platformy programowania obiektowego	Student zna i rozumie wybrane modele programowania i ewolucję związanych z nimi języków. Student poznaje jedną z platform programowania obiektowego.			[SW1] Ocena wiedzy faktograficznej		

Treści przedmiotu	<p>WYKŁADY Wprowadzenie. Języki programowania, alfabet, syntaktyka i semantyka. Translacja. Klasyfikacja typów. Typy całkowite i zmiennoprzecinkowe. Operatory i wyrażenia arytmetyczne. Wybrane standardowe funkcje matematyczne. Typ znakowy. Rzutowanie typów. Typ logiczny. Operatory i wyrażenia logiczne. Podstawy obsługi wejścia/wyjścia. Instrukcje warunkowe (if, switch) oraz wyrażenie warunkowe. Instrukcje iteracyjne (for, while, do-while). Iteracje zagnieżdżone. Definiowanie typów. Stałe. Typ wyliczeniowy. Tablice jednowymiarowe i wielowymiarowe. Napisy. Zakres ważności i czas życia zmiennych. Funkcje. Efekty uboczne. Przesyłanie parametrów funkcji. Typ wskaźnikowy. Arytmetyka wskaźników. Wskaźniki w komunikacji między funkcjami. Dynamiczna alokacja pamięci. Podstawowe dynamiczne struktury danych. Struktury (rekordy). Struktury danych wykorzystujące rekordy i ich zastosowania. Zastosowania dynamicznych struktur danych (stosy, kolejki). Wprowadzenie do programowania obiektowego. Definiowanie klas. Zastosowania paradygmatu programowania obiektowego. Klasy strumieni wejścia/wyjścia. Formatowanie wejścia/wyjścia. Przetwarzanie plików. Zastosowania rekurencji (np. zasada dzieli i zwyciężaj, zachłanność, programowanie dynamiczne).</p> <p>LABORATORIUM Rozwiązywanie prostych zadań programistycznych w oparciu o wiedzę przekazaną na wykładzie na podstawie instrukcji laboratoryjnych.</p> <p>PROJEKT Samodzielne wykonanie zadań programistycznych. Student ma do dyspozycji konsultacje projektowe.</p>														
Wymagania wstępne i dodatkowe	Brak wymagań														
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	<table border="1"> <thead> <tr> <th>Sposób oceniania (składowe)</th> <th>Próg zaliczeniowy</th> <th>Składowa ocena końcowej</th> </tr> </thead> <tbody> <tr> <td>kolokwium pisemne</td> <td>50.0%</td> <td>40.0%</td> </tr> <tr> <td>laboratorium</td> <td>50.0%</td> <td>30.0%</td> </tr> <tr> <td>projekt</td> <td>50.0%</td> <td>30.0%</td> </tr> </tbody> </table>	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa ocena końcowej	kolokwium pisemne	50.0%	40.0%	laboratorium	50.0%	30.0%	projekt	50.0%	30.0%		
Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa ocena końcowej													
kolokwium pisemne	50.0%	40.0%													
laboratorium	50.0%	30.0%													
projekt	50.0%	30.0%													
Zalecana lista lektur	<p>Podstawowa lista lektur</p> <p>Uzupełniająca lista lektur</p> <p>Adresy eZasobów</p>	<p>1. Podstawy programowania - notatki do wykładu, 2013 (dostępne są na stronie WWW kursu).</p> <p>2. Materiały przygotowujące do laboratorium z Podstaw programowania (opracowanie zespołowe, 2013) (dostępne są na stronie WWW kursu).</p> <p>3. Grębosz Jerzy, Symfonia C++ Standard (tom 1 i 2), Wydanie 2000, Krakow 2008.</p> <p>-</p>													
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<p>Napisanie programu realizującego zadaną funkcjonalność.</p> <p>Przeanalizowanie sposobu działania zadanego programu.</p>														
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy														