

KARTA KURSU

Nazwa	Wprowadzenie do programowania	
Nazwa w j. ang.	Introduction to Programming	
Koordinator	dr hab. Piotr Czerski, prof. UKEN	Zespół dydaktyczny
		dr hab. Piotr Czerski, prof. UKEN
Punktacja ECTS*	3	

Opis kursu (cele kształcenia)

Celem kursu jest zapoznanie studentów z podstawami modelowania systemów informatycznych za pomocą technik programowania funkcjonalnego i obiektowego. przy użyciu języków C,C++. Przedmiot prowadzony jest w języku polskim z użyciem terminologii angielskiej.

Warunki wstępne

Wiedza	Student zna podstawy programowania na poziomie szkoły średniej.
Umiejętności	
Kursy	

Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Wiedza	W01 – Zna w zaawansowanym stopniu terminologię informatyczną w zakresie programowania.	K_W04
	W02 – Posiada podstawową wiedzę dotyczącą programowania.	K_W04

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Umiejętności	U01 – Posługuje się językiem specjalistycznym z zakresu programowania. Potrafi formułować i analizować problemy programistyczne oraz dobiera możliwe optymalne metody ich rozwiązywania.	K_U03
	U02 – Potrafi posługiwać się narzędziami programistycznymi.	K_U03

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Kompetencje społeczne	K01 – Samodzielnie i odpowiedzialnie podejmuje zadania w zakresie programowania.	K_K02

Organizacja											
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	15					15					

Opis metod prowadzenia zajęć

Wykład teoretyczny z podstaw programowania.
 Podczas pracy laboratoryjnej studenci będą rozwiązywać problemy zadane przez prowadzącego zajęcia programując je w języku wysokiego poziomu.
 Na ćwiczeniach na bieżąco weryfikowana będzie wiedza przekazywana podczas wykładów.

Formy sprawdzania efektów uczenia się

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Inne
W01	x				x				x			x	
W02	x				x				x			x	
U01	x				x							x	
U02	x				x							x	
K01	x				x				x			x	
K02													
...													

Kryteria oceny	Obecność na zajęciach, wykonywanie zadań programistycznych podczas ćwiczeń, kolokwium sprawdzające wiedzę teoretyczną.
----------------	--

Uwagi	
-------	--

Treści merytoryczne (wykaz tematów)

Systemy liczbowe. Podstawy dwuwartościowej algebry Boole’a. Architektura von Neumanna i maszyna Turinga. Programowanie równoległe. Podstawowe struktury danych. Analiza złożoności. Sortowanie. Rekurencja. Związek struktury danych z funkcjami. Dziedziczenie. Wskaźniki. Polimorfizm. C++ a programowanie obiektowe. Standardowa biblioteka szablonów. Funktory. Drzewa binarne. Notacja polska i drzewa wyrażeń. Inżynieria oprogramowania.

Wykaz literatury podstawowej

Wirh N.: Algorytmy + struktury danych = program., WNT, Warszawa 2004 (wyd. 7).
 Drozdek A.: C++, algorytmy i struktury danych, Helion 2004.
 Grębosz J.: Opus Magnum C++, Helion 2020

Wykaz literatury uzupełniającej

Eckel B.: Thinking in C++, t.1, Helion 2002
 Stroustrup B.: Programowanie. Teoria i praktyka z wykorzystaniem C++, Helion 2013
 Stroustrup B.: Język C++. Kompedium wiedzy, Helion 2014
 Morris Mano M., Kime C. R.: Podstawy projektowania układów logicznych i komputerów, WNT 2007

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta)

liczba godzin w kontakcie z prowadzącymi	Wykład	15
	Konwersatorium (ćwiczenia, laboratorium itd.)	15
	Pozostałe godziny kontaktu studenta z prowadzącym	15
liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	15
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	5
	Przygotowanie do egzaminu/zaliczenia	10
Ogółem bilans czasu pracy		75
Liczba punktów ECTS w zależności od przyjętego przelicznika		3