

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Programowanie niskopoziomowe		Kod 1010514331010510078
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 2 / 3
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 12 Ćwiczenia: - Laboratoria: 12 Projekty/seminaria: -		Liczba punktów 3
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 3 100%
Odpowiedzialny za przedmiot / wykładowca:		
Dr inż. Agnieszka Rybarczyk email: Anieszka.Rybarczyk@put.poznan.pl tel. tel. (0-61) 665-3029, fax: (0-61) 877 1525 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu podstaw programowania i znać podstawowe metody stosowane przy rozwiązywaniu prostych zadań informatycznych.
2	Umiejętności:	Powinien również posiadać umiejętność rozwiązywania podstawowych problemów, którymi zajmuje się informatyka oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
Cel przedmiotu:		
1. Przekazanie studentom podstawowej wiedzy z zakresu programowania systemowego na poziomie języka C oraz zapoznanie z teoretycznymi i praktycznymi problemami związanymi z metodologią programowania w tym języku. 2. Rozwijanie u studentów umiejętności rozwiązywania prostych problemów i konstruowania prostych algorytmów.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów, języków i paradygmatów programowania, - [K_W4] 2. ma szczegółową wiedzę nt. algorytmiki, - [K_W5] 3. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu analizy złożoności obliczeniowej algorytmów i problemów, implementacji języków programowania, - [K_W8]		
Umiejętności:		
1. potrafi wybrać język programowania odpowiedni do danego zadania programistycznego - [K_U20] 2. ma umiejętność formułowania algorytmów i ich programowania z użyciem przynajmniej jednego z popularnych narzędzi - [K_U22]		
Kompetencje społeczne:		
1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1]		
Sposoby sprawdzenia efektów kształcenia		

<p>Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób: Ocena formująca: a) w zakresie wykładów: - na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach. b) w zakresie laboratoriów: - na podstawie oceny bieżącego postępu realizacji zadań, Ocena podsumowująca: Sprawdzanie założonych efektów kształcenia realizowane jest przez: - ocenę przygotowania studenta do poszczególnych sesji zajęć laboratoryjnych (sprawdzian ?wejściowy</p>		
Treści programowe		
<p>Celem przedmiotu jest zapoznanie studenta z zagadnieniami programowania w języku C. Przedmiot obejmuje pełen wykład języka C ilustrowany licznymi przykładami. Oprócz wykładów będą prowadzone ćwiczenia laboratoryjne, w trakcie których studenci będą rozwiązywać zadania ilustrujące zagadnienia omawiane na wykładach.</p> <p>Pierwszy wykład będzie poświęcony wprowadzeniu do języka C i obejmować będzie rys historyczny, omówienie prostych programów oraz podstawowych jednostek leksykalnych języka C. Na kolejnym wykładzie zostaną przedstawione instrukcje proste, złożone, wyboru, warunkowe oraz pętli wraz z licznymi przykładami. Na trzecim wykładzie zaprezentowane zostaną dyrektywy preprocesora, deklaracje i definicje globalnych danych i struktur danych oraz deklaracje i definicje funkcji. Wykład czwarty zostanie poświęcony modularyzacji, tablicom jedno- i wielowymiarowym, funkcjom obliczającym oraz nie obliczającym wartości oraz podstawowym błędom popełnianym podczas ich definiowania. Na wykładzie piątym omówione zostaną zakresy i zasięgi deklaracji zmiennych, lokalizacja zmiennych oraz pliki dyskowe. Wykład szósty zostanie poświęcony alokacji pamięci. Na ostatnim wykładzie zostaną omówione podstawowe struktury dynamiczne.</p> <p>W ramach pierwszego laboratorium studenci zapoznają się z ogólnymi zasadami i elementami przedmiotu, zapoznają się ze środowiskiem programistycznym oraz rozpoczną pisanie prostych programów. Na następnym laboratorium studenci nauczą się pisać proste programy modułowe oraz wykorzystywać podstawowe operatory i instrukcje warunkowe. Na trzecim laboratorium studenci nauczą się stosować pętle, instrukcję wyboru, wskaźniki oraz tablice. Czwarte laboratorium będzie poświęcone definiowaniu funkcji, dynamicznemu przydziałowi pamięci, strukturom oraz operacjom dyskowym. Laboratorium piąte będzie przeznaczone na realizację projektu zaliczeniowego a szóste na jego przedstawienie i ocenę. Na laboratorium siódmym odbędzie się pisemny sprawdzian z zakresu materiału poruszanego na zajęciach.</p> <p>Metody dydaktyczne: 1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami, rozwiązywanie zadań. 2. ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne.</p>		
Literatura podstawowa:		
<p>1. Język ANSI C, B.W. Kernighan, D.M. Ritchie, WNT, Warszawa, 1998. 2. Symfonia C++, J. Grębosz, Oficyna Kallimach, Kraków, 2001 3. Programowanie w języku C++, J. Kniat, Nakom, Poznań, 2002.</p>		
Literatura uzupełniająca:		
<p>1. Wprowadzenie do programowania w języku C++, J. Kniat, WPP, Poznań, 1995 2. Pasja C++, J. Grębosz, Oficyna Kallimach, Kraków, 2001</p>		
Bilans nakładu pracy przeciętnego studenta		
Czynność		Czas (godz.)
1. udział w zajęciach laboratoryjnych: 15 x 1 godz.,		16
2. przygotowanie do ćwiczeń laboratoryjnych:		5
3. dokończenie (w ramach pracy własnej) sprawozdań z ćwiczeń laboratoryjnych:		5
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu		4
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)		6
6. przygotowanie do sprawdzianów / kolokwium		6
7. udział w wykładach		12
8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 150 stron		15
9. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym		6
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	75	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	32	1

Zajęcia o charakterze praktycznym	22	1
-----------------------------------	----	---