



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie niskopoziomowe

Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

-

Poziom studiów

pierwszego stopnia

Forma studiów

stacjonarne

Rok/semestr

1/2

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

angielski

Wymagalność

obligatoryjny

Liczba godzin

Wykład

16

Laboratoria

16

Inne (np. online)

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

2

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Marcin Radom

Odpowiedzialny za przedmiot/wykładowca:

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu podstaw programowania i znać podstawowe metody stosowane przy rozwiązywaniu prostych zadań informatycznych. Powinien również posiadać umiejętność rozwiązywania podstawowych problemów, którymi zajmuje się informatyka oraz umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien również rozumieć konieczność poszerzania swoich kompetencji.

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.

Rozwijanie u studentów umiejętności rozwiązywania prostych problemów i konstruowania prostych algorytmów.

Przedmiotowe efekty uczenia się

Wiedza

W wyniku przeprowadzonych zajęć student:



1. Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów, języków i paradygmatów programowania oraz wiedzę szczegółową w zakresie wybranych zagadnień.
2. Ma podstawową wiedzę o cyklu życia systemów informatycznych, zarówno sprzętowych jak i programowych, a w szczególności o zachodzących w nich kluczowych procesach.
3. Zna podstawowe techniki, metody i narzędzia wykorzystywane w procesie rozwiązywania zadań informatycznych, głównie o charakterze inżynierskim, z zakresu kluczowych zagadnień informatyki.

Umiejętności

W wyniku przeprowadzonych zajęć student:

1. Potrafi – zgodnie z zadaną specyfikacją – zaprojektować oraz zrealizować szeroko rozumiany system informatyczny, dobierając język programowania odpowiedni do danego zadania programistycznego oraz używając właściwych metod, technik i narzędzi.
2. Ma umiejętność formułowania algorytmów i ich implementacji z użyciem przynajmniej jednego z popularnych narzędzi.

Kompetencje społeczne

Zaliczenie przedmiotu oznacza, że student:

1. Rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe.
2. Ma świadomość znaczenia wiedzy w rozwiązywaniu problemów inżynierskich oraz zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

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:

a) w zakresie wykładów:



- kolokwium zaliczeniowe przeprowadzane w ostatnich tygodniach zajęć lub w ustalonym ze studentami terminie w trakcie sesji. Zaliczenie składa się z 10 pytań, z których student może uzyskać łączną liczbę 10 punktów (0 - 4 pkt: 2.0; 5 pkt – 3.0; 6 pkt: 3.5; 7-8 pkt: 4.0; 9pkt: 4.5; 10pkt: 5).
- b) w zakresie laboratoriów sprawdzanie założonych efektów kształcenia realizowane jest przez:
- ciągłą ocenę wiedzy i umiejętności studentów na podstawie analizy sposobu praktycznego rozwiązywania zadanych na laboratorium problemów,
 - wyniki sprawdzianu polegającego na samodzielnym, praktycznym (przy komputerze) rozwiązaniu zadań programistycznych.
 - ocenę i „obronę” przez studenta zrealizowanego projektu

Treści programowe

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.

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.

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.

Metody dydaktyczne



1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami.
2. ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne.

Literatura

Podstawowa

1. Język ANSI C, B.W. Kernighan, D.M. Ritchie, WNT, Warszawa, 1998.
2. Symfonia C++, J. Grębosz, Oficyna Kallimach, Kraków, 2001

Programowanie w języku C++, J. Kniat , Nakom, Poznań, 2002.

Uzupełniająca

1. Wprowadzenie do programowania w języku C++, J. Kniat, WPP, Poznań, 1995
2. Pasja C++, J. Grębosz, Oficyna Kallimach, Kraków, 2001.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	50	2,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	32	1,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do kolokwiów, wykonanie projektu) ¹	18	1,0

¹ niepotrzebne skreślić lub dopisać inne czynności