

Jednostka prowadząca: Wydział Techniczny

Kierunek studiów: Elektronika i telekomunikacja

Nazwa przedmiotu: Język programowania C++

Charakter przedmiotu: podstawowy, obowiązkowy

Typ studiów: inżynierskie I-go stopnia, stacjonarne/niestacjonarne

Formy dydaktyczne i terminarz:

Forma przedmiotu	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Rok studiów/Semestr	2/4	2/4	2/4		
Liczba godzin w semestrze	30/18	15/12	30/18		
Forma zaliczenia	Egzamin	zal.na ocenę	zal.na ocenę		
Liczba punktów ECTS	3/3	1/1	2/2		

WYKŁAD

Wymagania wstępne:

Wymagane zaliczenie wykładu z Informatyki.

Cele kształcenia:

Celem przedmiotu jest uzyskanie przez studenta wiedzy z zakresu podstaw, jak i bardziej zaawansowanych technik programowania w języku programowania C. W ramach zajęć zostaną omówione zasady konstruowania, zapisu i analizy algorytmów oraz podstawy programowania w języku strukturalnym (język C), w tym struktura programu, typy danych, współpraca z systemem plików, a także biblioteki funkcji oraz podstaw programowania obiektowego w C++.

Oczekuje się, że student będzie w stanie zapisywać wybrane algorytmy oraz pisać proste programy w języku C /C++ je realizujące.

Metody dydaktyczne:

Pokaz, prezentacja, demonstracja, wykład.

Zasady i kryteria zaliczenia:

Zaliczenie w postaci testu wielokrotnego wyboru, podstawą zaliczenia przedmiotu jest uzyskanie co najmniej 50 % maksymalnej liczby punktów.

Treści programowe:

1. Języki programowania (składnia, semantyka), wstęp do programowania strukturalnego i obiektowego krótki zarys historyczny języka C, dialekty języka C.
2. Program i jego elementy:
 - a. stałe,
 - b. zmienne,
 - c. instrukcje proste,
 - d. instrukcje strukturalne.
3. Preprocesor i dyrektywy kompilatora: #define, #include.

4. Pojęcie typu danych:
 - a. typy proste,
 - b. tablice,
 - c. łańcuch znaków,
 - d. unie.
5. Dynamiczne struktury danych.
6. Operatory arytmetyczne, logiczne, relacyjne, teoriomnogościowe.
7. Funkcje i podprogramy - main().
8. Wprowadzenie do programowania obiektowego:
 - klasy,
 - obiekty,
 - pola,
 - metody,
 - metody wirtualne,
 - interfejsy,
 - dziedziczenie,
 - polimorfizm,
 - enkapsulacja.
9. Zasady programowania obiektowego:
 - wielokrotne wykorzystanie klas,
 - kontrola dostępu,
 - polimorfizm w praktyce.
10. Kontenery obiektów, biblioteka STL.
11. Operacje wejścia wyjścia, strumienie w C++, obsługa wyjątków w C++.
12. Typy ogólne/szablony.
13. Wzorce projektowe, UML i wprowadzenie do metodologii tworzenia projektów na przykładzie programowania ekstremalnego (XP).

Literatura podstawowa:

1. Aho A. V., Hopcroft J. E., Ullman J. D., *Algorytmy i struktury danych*. Wydawnictwo HELION, Gliwice 2007.
2. Liberty J., *C++*. Wydawnictwo HELION, Gliwice 2002.
3. Stroustrup B., *Język C++*. WNT, Warszawa 1998.

Literatura uzupełniająca:

1. Adamiec-Wójcik I., Guerreiro P., *Elementy programowania obiektowego w C++*. Wydawnictwo Politechniki Łódzkiej, Filii w Bielsku-Białej, Bielsko Biała 1998.
2. Liberty J., *C++: księga eksperta*. Wydawnictwo HELION, Gliwice 1999.

Efekty kształcenia:

Umiejętność algorytmizacji problemów; implementacji algorytmów w języku C/C++ i wybranym środowisku programistycznym; tworzenia programów obiektowych; konstruowania dynamicznych struktur danych; wykonywania obliczeń numerycznych i przetwarzania danych.

Język wykładowy: polski.

ĆWICZENIA

Wymagania wstępne:

Wymagane zaliczenie wykładu z Informatyki.

Zasady i kryteria zaliczenia:

Uzyskanie co najmniej 50% punktów możliwych do zdobycia za rozwiązanie zadań z listy lub zaliczenie kolokwium w postaci testu.

Cele kształcenia:

Praktyczne zrozumienie i nabycie umiejętności posługiwania się pojęciami prezentowanymi na wykładzie poprzez rozwiązywanie prostych zadań. Zestawy zadań na ćwiczenia są proponowane przez prowadzącego.

Treści programowe:

1. Program i jego elementy:
 - a. stałe,
 - b. zmienne,
 - c. instrukcje proste,
 - d. instrukcje strukturalne.
2. Wyrażenia w języku C++.
3. Zmienne wskaźnikowe i referencje.
4. Programowanie obiektowe w C++:
 - klasy,
 - obiekty,
 - pola,
 - metody,
 - metody wirtualne,
 - interfejsy,
 - dziedziczenie,
 - polimorfizm,
 - enkapsulacja,
 - wielokrotne wykorzystanie klas,
 - kontrola dostępu,
 - polimorfizm w praktyce.
5. Kontenery obiektów, biblioteka STL.
6. Operacje wejścia wyjścia, strumienie w C++, obsługa wyjątków w C++.

Literatura podstawowa:

1. Hansen T.L., *C++: zadania i odpowiedzi*. WNT, Warszawa 1994.
2. Liberty J., *C++*. Wydawnictwo HELION, Gliwice 2002.
3. Stroustrup B., *Język C++*. WNT, Warszawa 1998.

Literatura uzupełniająca:

1. Adamiec-Wójcik I., Guerreiro P., *Elementy programowania obiektowego w C++*. Wydawnictwo Politechniki Łódzkiej Filii w Bielsku-Białej, Bielsko Biała 1998.
2. Liberty J., *C++: księga eksperta*. Wydawnictwo HELION, Gliwice 1999.
3. Vandevoorde D., *Język C++: ćwiczenia i rozwiązania*. WNT, Warszawa 2001.

Efekty kształcenia:

Po ukończeniu kursu student powinien umieć samodzielnie bez użycia kompilatora, znaleźć w kodzie programu C/C++ błędy składniowe, a także rozumieć w sposób głęboki treść programów stworzonych samodzielnie, jak i umieć „czytać kod” programów napisanych przez innych.

LABORATORIUM

Wymagania wstępne:

Wymagane zaliczenie laboratorium z Informatyki.

Cele kształcenia:

Zajęcia laboratoryjne mają na celu zdobycie przez studenta umiejętności pisania prostych programów w języku C/C++. Zakłada się że studenci będą w stanie pisać proste programy z wykorzystaniem języka C i stosować w praktyce podstawowe algorytmy i struktury danych, a także wykorzystywać cechy obiektowe języka C++.

Metody dydaktyczne:

Pokaz, prezentacja, ćwiczenia laboratoryjne.

Zasady i kryteria zaliczenia:

Pozytywna ocena z wszystkich ćwiczeń laboratoryjnych.

Treści programowe:

W ramach laboratorium student zdobywa umiejętność praktycznego wykorzystania wiedzy teoretycznej przedstawionej na wykładzie. Program obejmuje naukę pisania aplikacji począwszy od najprostszych programów aż do programów obliczeniowych o znacznym stopniu komplikacji. Oprócz zastosowania wiedzy zdobytej na wykładzie szczególnie nacisk jest położony na praktyczne aspekty programowania, a więc umiejętność posługiwania się kompilatorem i linkerem, umiejętność pisania „eleganckich” programów poprzez zastosowanie komentarzy i wcięć, radzenie sobie z typowymi błędami programistycznymi oraz optymalizacją kodu. W szczególności laboratorium dotyczyć będzie następujących treści:

1. Poprawne a niepoprawne deklarowanie typów i zmiennych w C.
2. Poprawne a niepoprawne zapisywanie instrukcji w języku C.
3. Typy danych.
4. Tablice i unie.
5. Współpraca z systemem plików.
6. Biblioteki funkcji.
7. Podstawowy tworzenia i używania klas.
8. Posługiwanie się dziedziczeniem i hierarchizacją.
9. Polimorfizm.
10. Strumienie.
11. Wstęp do programowania GUI. Programowanie interakcji z użytkownikiem.

Literatura podstawowa:

1. Meyers Scott, *Język C++ bardziej efektywny – 35 praktycznych sposobów ulepszenia programów*. WNT, Warszawa 1998.

Literatura uzupełniająca:

1. Hansen T.L., *C++: zadania i odpowiedzi*. WNT, Warszawa 1994.
2. Vandevoorde D., *Język C++: ćwiczenia i rozwiązania*. WNT, Warszawa 2001.

Efekty kształcenia:

Praktyczna umiejętność implementacji algorytmów w języku C/C++, tworzenie programów strukturalnych; konstruowanie dynamicznych struktur danych; tworzenie klas w C++.

Osoby prowadzące:

dr inż. Zbigniew Fjałkowski

mgr Beata Laszkiewicz

mgr inż. Paweł Hofman

mgr inż. Michał Jur

mgr Wojciech Liber

mgr inż. Konrad Żurawski

mgr inż. Bartosz Kwaśniewski