



UNIWERSYTET
IM. ADAMA MICKIEWICZA
W POZNANIU

Zaawansowane języki programowania

Sylabus zajęć

Informacje podstawowe

Kierunek studiów Technologie komputerowe		Cykl dydaktyczny 2023/24	
Specjalność -		Kod zajęć 04TKOS.38N.04999.23	
Jednostka organizacyjna Wydział Fizyki		Języki wykładowe polski	
Poziom studiów studia inżynierskie pierwszego stopnia		Obligatoryjność Obowiązkowy	
Forma studiów studia stacjonarne		Blok zajęciowy Przedmioty nieprzypisane	
Profil studiów profil ogólnoakademicki			
Koordynator zajęć	Michał Mucha		
Prowadzący zajęcia	Michał Mucha		
Okres Semestr 4	Forma zajęć / liczba godzin / forma zaliczenia • Wykład: 30, Egzamin • Laboratorium: 30, Zaliczenie z oceną		Liczba punktów ECTS 4

Cele kształcenia dla zajęć

Kod	Cel
C1	Opanowanie przez słuchaczy umiejętności pisania kodu w języku C++ z wykorzystaniem podstawowych struktur języka. Przede wszystkim: operacji wejścia wyjścia, dynamicznej alokacji pamięci, pracy ze wskaźnikami, tworzenia obiektów (również dynamiczne), wykorzystanie dziedziczenia.
C2	W dalszym etapie opanowanie korzystania: z obsługi wyjątków, biblioteki STL, tworzenia i korzystania z dynamicznych bibliotek, umiejętności wywołania API systemu operacyjnego, tworzenia programów wielowątkowych.
C3	Opanowanie umiejętności programowania z wykorzystaniem GUI (Graphical User Interface) na przykładzie biblioteki wxWidgets

Wymagania wstępne

Znajomość podstaw programowania w języku C/C++.

Znajomość angielskiego na poziomie pozwalającym na czytanie dokumentacji.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	wie: Jakie są elementarne instrukcje wykonywane przez procesory. Co to są asemblery, języki: kompilowane, interpretowane i działające w oparciu o maszyny wirtualne. Zna podstawowe elementy występujące w językach programowania.	TKO_K3_W04_inz, TKO_K3_W06_inz	Egzamin ustny, Kolokwium ustne, Projekt
W2	zna: podstawowe słowa kluczowe języków C i C++. Zna składnię, wbudowane typy danych, instrukcje sterujące. Zna pojęcia klas. Zna koncepcję dziedziczenia, funkcji wirtualnych, klas abstrakcyjnych, polimorfizmu.	TKO_K3_W04_inz, TKO_K3_W05_inz	Egzamin ustny, Kolokwium ustne, Projekt
W3	zapoznaje się z bibliotekami runtime C/C++, pojęciami szablonów, obiektów funkcyjnych. Zapoznaje się z biblioteką STL - strukturami danych, iteratorami, algorytmami.	TKO_K3_W04_inz, TKO_K3_W05_inz	Egzamin ustny, Kolokwium ustne, Projekt
W4	zapoznaje się z zasadami tworzenia programów interakcyjnych. Zapoznaje się z koncepcją GUI (Graphical User Interface) na przykładzie biblioteki wxWidgets	TKO_K3_W04_inz, TKO_K3_W05_inz	Egzamin ustny, Kolokwium ustne, Projekt
W5	zapoznaje się z zasadami tworzenia programów wielowątkowych. Mutex'y Semaphores. Zapoznaje się z koncepcją smart pointers - klasy unique_ptr, shared_ptr, weak_ptr,	TKO_K3_W04_inz, TKO_K3_W05_inz	Egzamin ustny, Kolokwium ustne, Projekt
Umiejętności - Student/ka:			
U1	zapoznaje się ze środowiskiem IDE. Potrafi tworzyć "projekty", pisać i kompilować proste programy.	TKO_K3_U04_inz, TKO_K3_U07_inz	Kolokwium ustne, Projekt
U2	pisać programy używając różne typy danych, tablice, struktury, wskaźniki, dynamiczną alokację pamięci, operacje wejścia wyjścia.	TKO_K3_U04_inz, TKO_K3_U07_inz	Kolokwium ustne, Projekt

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
U3	potrafi definiować klasy. Używać zdefiniowane przez siebie typy danych. Używać bardziej złożone typy danych. Wykorzystywać dziedziczenie i funkcje wirtualne. Wykorzystywać wyjątki.	TKO_K3_U04_inz, TKO_K3_U07_inz	Kolokwium ustne, Projekt
U4	potrafi używać szablony klas i funkcji. Tworzyć programy z wykorzystaniem STL.	TKO_K3_U04_inz, TKO_K3_U07_inz	Kolokwium ustne, Projekt
U5	potrafi tworzyć programy z interface'em graficznym.	TKO_K3_U04_inz, TKO_K3_U07_inz	Kolokwium ustne, Projekt

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
1.	Informacje ogólne o językach programowania. Kompilacja. Identyfikatory proste (wbudowane) typy danych oraz ich reprezentacja binarna. Zakres widoczności zmiennych. Podstawowe operatory.	W1, W2, U1, U2	Wykład, Laboratorium
2.	Instrukcje sterujące, pętle. Instrukcje warunkowe. Funkcje scanf i printf. Formatowanie wydruków (łańcuchy formatujące) w C. Operatory ++ i --. Wywoływanie funkcji z argumentami.	W1, W2, U1, U2	Wykład, Laboratorium
3.	Wskaźniki. Tablice. Struktury. Struktury anonimowe. Typy danych użytkownika w C. Dostęp do składowych struktur. Wywoływanie funkcji z wykorzystaniem struktur. Pola bitowe. Unie. Deklarowanie wskaźników. Arytmetyka na wskaźnikach. Konwersja wskaźników. Wskaźniki do funkcji. Stałe a wskaźniki. Inicjalizacja zmiennych. Operacje wejścia wyjścia w C. Przykłady. Ćwiczenia: mnożenie macierzy, Dynamiczna tabela łańcuchów. Alokacja tablic w „stylu Pascal’a”. Symulacja rzutów monetą. Biblioteczne funkcje do manipulacji łańcuchów. Listy. Stos. Kolejka. Operator wywołania funkcji. Modyfikatory deklaracji.	W1, W2, U1, U2	Wykład, Laboratorium
4.	Klasy. Obiekty. Deklaracje klas za pomocą: class, struct, union. Dziedziczenie. Dostęp do składowych klas. Dostęp do składowych klas bazowych i potomnych. Wirtualne klasy bazowe. Funkcje zaprzyjaźnione.	W1, W2, U1, U2, U3	Wykład, Laboratorium

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
5.	Operacje wejścia wyjścia w C++. Manipulatory. Klasy ofstream, ifstream. Operatory << >>. Właściwości konstruktorów i destruktorów. Konstruktory: domyślny, kopiujący. Przedefiniowanie konstruktorów. Kolejność wywołania konstruktorów. Inicjalizacja klas. Konstruktory jednoargumentowe. Destruktry. Wywołanie destruktorów. Destruktry a funkcja atexit i dyrektywa kompilatora #pragma exit. Destruktry a funkcja exit. Destruktry a funkcja abort. Destruktry wirtualne Słowo kluczowe this. Składowe statyczne. Funkcje inline. Przedefiniowywanie operatorów - binarnych, unarnych. Operatorów: =, (), [], ->. Definicje operatorów inkrementacji i dekrementacji. Zmiana definicji operatorów &&, i, . Funkcje wirtualne. Klasy abstrakcyjne. Klasy polimorficzne. Wyjątki. Pominięte wywołania destruktorów. Wyjątki w konstruktorach. Niejawna (implicit) konwersja typów danych. Operatory new i delete.	W1, W2, W3, U1, U2, U3	Wykład, Laboratorium
6.	Makra. Zagnieżdżone nawiasy. Przecinki. Wstawianie tokena za pomocą ##. Zamiana identyfikatora zmiennej na łańcuch za pomocą #. Kompilacja warunkowa.	W1, W2, W3, U1, U2, U3	Wykład, Laboratorium
7.	Szablony. Szablony funkcji. Przekrywanie szablonów funkcji. Jawne i niejawne szablony funkcji. Szablony klas. Argumenty szablonów. Biblioteka STL (SCL). Motywacja utworzenia biblioteki. STL a standard ANSI/ISO. Klasy nice. Obiekty funkcyjne.	W1, W2, W3, U1, U2, U3, U4	Wykład, Laboratorium
8.	Przegląd biblioteki STL. Kontenery. Array, Vector. List. Double ended queue. Set. Map. Hash tables. Iteratory. Iteratory wejścia/wyjścia. Iteratory jednokierunkowe. Iteratory dwukierunkowe. Iteratory swobodnego dostępu. Algorytmy i obiekty funkcyjne. Jak utworzyć algorytm generyczny? Inne tematy związane z STL. Adaptory. Stos. Kolejki. Adaptory iteratorów. Iteratory wsteczne. Iteratory pamięci. Binder'y. Alokatory. Adaptory funkcji. Kontenery skojarzeniowe	W1, W2, W3, U1, U2, U3, U4	Wykład, Laboratorium
9.	Mutex. Semafor. Inteligentne wskaźniki. Wątki. Wątki utworzone za pomocą API Windows. Priorytety wątków. Sekcja krytyczna. Synchronizacja wątków	W1, W2, W3, W4, W5, U1, U2, U3, U4, U5	Wykład, Laboratorium
10.	Przeszukiwanie dysku za pomocą funkcji shell'a (API). Wininet API. Biblioteka cURL. Biblioteka wxWidgets	W1, W2, W3, W4, W5, U1, U2, U3, U4, U5	Wykład, Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Wykład	Wykład z prezentacją multimedialną wybranych zagadnień
Laboratorium	Rozwiązywanie zadań (np.: obliczeniowych, artystycznych, praktycznych), Metoda warsztatowa, Metoda projektu

Forma zajęć	Warunki zaliczenia zajęć
Wykład	Warunkiem zaliczenia wykładu jest pozytywna ocena z egzaminu obejmującego znajomość składni, podstawowych struktur i koncepcji języków C/C++.
Laboratorium	Warunkiem zaliczenia zajęć jest pozytywna ocena z następujących elementów: - wykonanie zadań - utworzenie własnego projektu - prezentacja działającego programu.

Literatura

Obowiązkowa

1. Kernigham Brain, Ritchie Dennis, „Język ANSI C” WNT Warszawa 1998
2. Stroustrup, Bjarne „Język C++” wydanie 4 WNT Warszawa 1998. Oryginał „The C++ programming language” -- 2nd ed. Addison-Wesley 1993
3. Eckel Bruce, „Thinking in C++. Edycja polska.” Helion. Gliwice. Na stronie internetowej <http://www.ibiblio.org/pub/docs/books/eckel/> dostępne są (za darmo) oryginalne wersje tej oraz innych książek autora „Thinking in C.”, „Thinking in Java”.
4. Vandevoorde David, Josuttis Nicolai M., „C++ Szablony.” Helion Gliwice 2003. Oryginał “C++ templates: The complete Guide.” Pearson Education Inc. część Addison Wesley 2003.
5. Grębosz Jerzy, „Opus Magnum C++11. Programowanie w języku C++” tomy 1 -- 3, Helion SA Gliwice 2018 Grębosz Jerzy, „Opus Magnum C++. Misja w nadprzestrzeń C++14/17” Helion SA, Gliwice 2020.

Dodatkowa

1. Meyers Scott, „Język C++ bardziej efektywny” Wydawnictwa Naukowo-Techniczne, Warszawa 1998. Oryginał „More Effective C++. 35 New Ways to Improve Your Programs and Designs” Addison-Wesley 1996.
2. Josuttis Nicolai M., „C++ biblioteka standardowa. Podręcznik programisty.” Helion Gliwice 2003. Oryginał „The C++ Standard Library: A Tutorial and Reference” Pearson Education Inc. część Addison-Wesley 1999.
3. Andrew Koenig, Barbara E. Moo, „Accelerated C++. Practical Programing by Example”, Pearson Education część Addison-Wesley Professional 2000. Polskie tłumaczenie „C++ potęga języka” Helion Gliwice 2004
4. Grębosz Jerzy, „Symfonia C++” tomy 1 -- 3, Oficyna Kallimach. Oraz inne książki tego autora poświęcone językom C i C++.
5. Hansen Tony, „C++ zadania i odpowiedzi” wydanie drugie WNT Warszawa 1994. Oryginał „The C++ Answer Book” Addison-Wesley 1990
6. Plauger P. J., „Biblioteka standardowa C++” WNT Warszawa 1997. Oryginał „The Standard C++ Library” Prentice Hall, Inc.
7. Lee Meng, Stepanov Alexander “The Standard Template Library” HP Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304 February 1995
8. Warzocha Bartosz, „Programowanie wieloplatformowe z C++ i wxWidgets 3” PWN Warszawa 2018
9. Jacek Galowicz “C++17 STL Receptury”, Helion Gliwice 2018. Oryginał “C++17 STL Cookbook” Packt Publishing 2017
10. <https://isocpp.org/home/> Strona komitetu standaryzacyjnego C++
11. <https://cplusplus.com/>
12. <https://en.cppreference.com/w/cpp/language>

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Wykład	30
Laboratorium	30
Przygotowanie do zajęć	30
Przygotowanie projektu	30
Łączny nakład pracy studenta	Liczba godzin 120
Liczba punktów ECTS	ECTS 4

* godzina (lekcyjna) oznacza 45 minut

Efekty uczenia się dla kierunku

Kod	Treść
TKO_K3_U04_inz	Absolwent/ka potrafi kodować, uruchamiać i testować programy na różnych platformach i w różnych środowiskach programistycznych
TKO_K3_U07_inz	Absolwent/ka potrafi ocenić użyteczność różnych narzędzi i metod informatycznych pod kątem typowych zadań, a następnie wybrać i użyć właściwe narzędzia i metody
TKO_K3_W04_inz	Absolwent/ka zna i rozumie fundamentalne pojęcia, struktury oraz procesy związane z językami programowania, inżynierią programowania i fizyką komputerową
TKO_K3_W05_inz	Absolwent/ka zna i rozumie fundamentalne pojęcia i zagadnienia związane z projektowaniem i weryfikacją algorytmów oraz struktur danych
TKO_K3_W06_inz	Absolwent/ka zna i rozumie fundamentalne pojęcia i zagadnienia związane z fizyczną i informatyczną architekturą sprzętowo-programową