

KARTA PRZEDMIOTU

Nazwa przedmiotu: Podstawy programowania komputerów (InfAu>SI1-PPK-19)

Nazwa w języku polskim:

Nazwa w jęz. angielskim: Fundamentals of computer programming

Dane dotyczące przedmiotu:

Jednostka oferująca przedmiot: Wydział Automatyki, Elektroniki i Informatyki

Przedmiot dla jednostki: Politechnika Śląska

Domyślny typ protokołu dla przedmiotu:

EGZ

Język wykładowy:

polski

Strona WWW:

<https://platforma2.polsl.pl/rau2/course/view.php?id=44>

Skrócony opis:

Przedmiot ma na celu zapoznanie studentów z programowaniem strukturalnym w języku wysokiego poziomu. Kurs obejmuje wykład i laboratorium. Zajęcia poświęcone są podstawom programowania w C++ (organizacja programu, funkcje, lambda, strumienie, pliki, wskaźniki, struktury danych z biblioteki STL). W trakcie zajęć studenci zapoznają się także z dobrymi praktykami programowania, projektowaniem aplikacji, doбором struktur danych do zadania.

Opis:

ECTS: 4

Suma godzin: 120h (kontaktowa 60h / praca własna 60h)

* wykład 30h

* laboratorium 30h

Praca własna studenta: przygotowanie do laboratorium, implementacja zadania programistycznego.

Przedmiot kształtuje umiejętności praktyczne.

Przedmiotu kształtuje kompetencje inżynierskie.

Tematyka

* struktura programu

* typy, zmienne, operatory

* typ wyliczeniowy

* tablice

* pętle (while, do-while, for, for zakresowy)

* funkcje (deklarowanie, definiowanie, sposoby przekazywania argumentów, tablic, rekurencja)

* zasięg zmiennych, zmienne automatyczne

* struktury, unie

* kontenery STL: array, vector, string, map, unordered_map, pair, tuple, list, forward_list, queue, priority_queue

* operacje wejścia/wyjścia w C++

* wskaźniki (surowe i inteligentne), zmienne na sterpie, alokacja i zwalnianie pamięci w C++

* funkcje lambda

* przestrzenie nazw

* podział programu na pliki źródłowe i nagłówkowe

* operacje wejścia/wyjścia w C

* łańcuchy znakowe w C

* alokacja pamięci na sterpie w C

* preprocesor

tematyka wykładów i laboratoriów:

W01

* struktura programu

* typy, zmienne, operatory (także <=>)

* instrukcja warunkowa (if + ?:)

* operatory logiczne

W02

* pętle (while, do-while, for(;;))

* funkcja (prosta funkcja -- idea)

W03

* instrukcja switch

* tablice o stałym rozmiarze

* tablice dwuwymiarowe

* typ wyliczeniowy (enum, class enum)

W04

* bloki i zmienne lokalne, zakresy zmiennych, zmienne automatyczne

* funkcje (deklarowanie, sposoby przekazywania argumentów, tablic, rekurencja, inline, constexpr, constexpr, domyślne parametry funkcji, przeładowanie)

- W05
- * parametry main
 - * operacje wejścia/wyjścia (std::cout, std::cerr, std::cin, strumienie plikowe, strumień łańcuchowy)
- W06
- * manipulatory strumieni
 - * flagi błędów strumieni
 - * pliki nagłówkowe
 - * std::array
- W07
- * std::vector (tworzenie, push_back, [], size, capacity, lista inicjalizująca), for (::)
 - * std::string
 - * struktury
 - * unie
- W08
- * std::pair (make_pair, first, second)
 - * tablice asocjacyjne (map)
- W09
- * wskaźniki
 - * wskaźniki do zmiennej, stałej
 - * parametry wyjściowe funkcji (przez wskaźnik)
 - * tablice (podejście wskaźnikowe)
- W10
- * wskaźniki na funkcje
 - * typedef, using
 - * iteratory
 - * funkcje lambda
- W11
- * funkcje lambda (dokończenie)
 - * kontenery STL: (vector,) deque, (string, map), list, forward_list, set, priority_queue, stack, (pair,) tuple
- W12
- * kontenery STL (cd): list, forward_list, set, priority_queue, stack, (pair,) tuple
 - * std::format
- W13
- * alokacja pamięci, wycieki pamięci, wskaźniki inteligentne, alokacja tablicy (jedno- i wielowymiarowej)
- W14
- * ciąg dalszy inteligentnych wskaźników
 - * pliki binarne
 - * łańcuchy znakowe w C (idea, strlen, strcpy, strcmp)
- W15
- * operacje wejścia/wyjścia w C (printf, scanf, pliki)
 - * alokacja pamięci na sterce w C (malloc, calloc, realloc, free)
 - * preprocesor (makrodefinicje, kompilacja warunkowa, X macro)

Sylabus obowiązuje od roku akademickiego 2025/26 a jego zawartość nie podlega zmianom w trakcie trwania semestru.

Literatura:

1. J. Grębosz: „Opus magnum C++”, 2023
2. Tomasz Jaśniewski: „C++ Zbiór zadań z rozwiązaniami”, Helion 2023
3. Robert C. Martin: „Czysty kod. Podręcznik dobrego programisty”, Helion 2023
4. Bjarne Stroustrup: „Język C++. Kompendium wiedzy”, Helion 2014
5. Standard C++ Library reference, <http://www.cplusplus.com/reference/>

Efekty uczenia się:

wiedza: zna i rozumie

EU_W1 Student zna konstrukcje językowe dla programowania strukturalnego (K1A_W09)

EU_W2. Student rozumie, czym jest funkcja w programowaniu i jej rola w modularności kodu . (K1A_W11, K1A_U07)

EU_W3. Student zna własności struktur danych (K1A_W12)

umiejętności: potrafi

EU_U1. Student potrafi dobrać adekwatne do problemu struktury danych. (K1A_U21)

EU_U2. Student potrafi zaimplementować w języku wysokiego poziomu program rozwiązujący zadany problem (K1A_U23)

EU_U3. Student potrafi zastosować dobre praktyki programistyczne (K1A_U23)

EU_U4. Student potrafi zweryfikować poprawność implementacji (K1A_U15)

EU_U5. Student potrafi sporządzić dokumentację techniczną dla stworzonego programu (K1A_U03)

kompetencje społeczne: jest gotów do

Metody i kryteria oceniania:

Egzamin składa się z dwóch części: A części praktycznej (zadanie programistycznego) i B części ustnej.

A. część praktyczna – zadanie programistyczne

1. Część praktyczna jest obowiązkowa.
2. Temat zadania jest zlecany przez prowadzącego zajęcia.
3. Zadanie jest realizowane przez jedną osobę.
4. Pełen kod źródłowy programu musi znajdować się w repozytorium git.
5. Kod programu musi być skomentowany w standardzie doxygen.
6. Dokumentacja zadania wygenerowana w doxygenie musi zostać umieszczona na stronie przedmiotu na platformie zdalnej edukacji.
7. Zadanie programistyczne składa się z ocenianych składowych.
8. Prowadzący laboratorium określa termin oddania każdej składowej.
9. Osobista prezentacja każdej składowej projektu jest warunkiem koniecznym do określenia stopnia wypełniania efektów kształcenia.
10. Prowadzący może zweryfikować autorstwo programu w sposób, który uzna za odpowiedni.
11. Ocena z zadania programistycznego jest średnią arytmetyczną składowych, przy czym każda ze składowych musi być oceniona pozytywnie.
12. W wypadku niezaliczenia części praktycznej egzaminu przystąpienie do kolejnego terminu części praktycznej wymaga zaimplementowania nowego zadania programistycznego zleconego przez prowadzącego laboratorium. Oznacza to, że ponownie muszą być ocenione wszystkie składowe zadania programistycznego.

B. część ustna

1. Część ustna jest nieobowiązkowa.
2. Warunkiem koniecznym do przystąpienia do części ustnej jest ocena pozytywna z części praktycznej.
3. Zaliczenie części praktycznej zwalnia z części ustnej z oceną 3.0.
4. Część ustna jest oceniana w skali od 2 do 5, przy czym ocenami pozytywnymi są oceny od 3 (dostateczna) do 5 (bardzo dobra).

Ocena końcowa z przedmiotu jest średnią arytmetyczną ocen z części praktycznej i ustnej, przy czym obie z nich muszą być pozytywne.

Sylabus obowiązuje od roku akademickiego 2025/2026, a jego zawartość nie podlega zmianom w trakcie trwania semestru.

Punkty przedmiotu w cyklach:

<bez przypisanego programu>			
Typ punktów	Liczba	Cykl pocz.	Cykl kon.
Europejski System Transferu Punktów (ECTS)	4	2020/2021-Z	