**Nazwa przedmiotu:**

Języki i techniki programowania cz. 1

**Koordynator przedmiotu:**

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Geoinformatyka

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

1060-GI000-ISP-1007

**Semestr nominalny:**

1 / rok ak. 2023/2024

**Liczba punktów ECTS:**

3

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

Udział w zajęciach, wykłady: 15 godzin,
Udział w zajęciach, ćwiczenia: 30 godzin,
Sprawozdania, raporty z zajęć, prace domowe: 30 godzin,
Udział w konsultacjach: 5 godziny
W sumie: 80 godzin = 3 ECTS

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

Udział w zajęciach, wykłady: 15 godzin,
Udział w zajęciach, ćwiczenia: 30 godzin,
Udział w konsultacjach: 5 godziny
w sumie: 50 godzin = 2 ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

Udział w zajęciach, ćwiczenia: 30 godzin,
Udział w konsultacjach: 5 godziny
w sumie: 35 godzin = 1,4 ECTS

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 15h |
| Ćwiczenia: | 30h |
| Laboratorium: | 0h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

\_\_

**Limit liczby studentów:**

-

**Cel przedmiotu:**

Celem przedmiotu jest zapoznanie studentów z programowaniem strukturalnym oraz elementami warsztatu programisty

**Treści kształcenia:**

Wykłady:
1. Elementy programu: pliki nagłówkowe, funkcje, operatory, instrukcje. Kompilacja i konsolidacja programu. Sygnalizacja błędów kompilacji i konsolidacji.
2. Obiekty, zmienne, typy i wartości. Zakresy wartości i operacje dopuszczalne dla danego typu. Użycie operatorów do wyrażania obliczeń. Kontrola typów.
3. Wyrażenia i instrukcje. l-wartości i r-wartości. Instrukcja warunkowa i instrukcje pętli. Deklaracje i definicje funkcji. Parametry formalne i aktualne.
4. Sposoby sygnalizowania błędów. Wyjątki: generowanie i przechwytywanie. Uruchamianie i testowanie programu. Techniki pisania programów przyjaznych w uruchamianiu.
5. Deklaracje i definicje. Zasięgi widoczności, przesłanianie. Przekazywanie argumentów funkcji przez wartość, referencję i ustaloną referencję. Przestrzenie nazw.
6. Klasy. Interfejs i implementacja. Składowe publiczne i prywatne. Przeciążanie operatorów. Typy wyliczeniowe. Niezmienniki typów użytkownika. Struktury.
7. Strumienie wejściowe i wyjściowe. Współpraca z plikami. Sygnalizacja błędów wejścia/wyjścia.
8. Formatowanie wejścia/wyjścia dla danych liczbowych i łańcuchowych. Klasyfikowanie znaków. Praca z plikami binarnymi. Różne standardy kodowania znaków; kody liczbowe.
9. Dynamiczna alokacja i dealokacja pamięci. Operatory new i delete. Destruktory. Wycieki pamięci.
10. Semantyka kopiowania. Kopiowanie głębokie i płytkie. Konstruktor i podstawienie przenoszące. Współpraca z tablicami.
11. Kontenery. Zarządzanie zasobami. Zasada RAII. Zarządzanie wskazaniami. Wykorzystanie iteratorów do przechodzenia po elementach kolekcji.
12. Biblioteka standardowa C++. Porównanie różnych rodzajów kontenerów. Typowe operacje wykonywane na kontenerach. Predykaty.
13. Algorytmy zdefiniowane w bibliotece standardowej. Standardowe obiekty funkcyjne.
14. Metryki kodu źródłowego. Refaktoryzacja. Systematyczne testowanie. Zagadnienia bezpieczeństwa kodu.
15. Różnice między C++ i C. Klasyfikacja języków programowania. Kompilacja, konsolidacja, interpretacja; kompilacja JIT; środowiska z maszyną wirtualną; języki DSL.
Ćwiczenia:
Ćwiczenia obejmują projekty polegające na napisaniu programów o rosnącym stopniu złożoności:
1. Prosty program realizujący obliczenia z interfejsem tekstowym
2. Program wykorzystujący własne typy danych z zestawem testów.
3. Program wykorzystujący kolekcje i algorytmy z biblioteki standardowej w połączeniu z typem użytkownika.
4. Utworzenie własnej biblioteki i wykorzystanie jej w programie współpracującym ze strumieniami plikowymi.
5. Program wraz z zestawem testów jednostkowych

**Metody oceny:**

W trakcie zajęć student może zgromadzić maksymalnie 100 punktów podzielonych na: 50 punktów z zadań programistycznych ocenianych w skali 0-5, 40 punktów za dwa kolokwia przeprowadzane w środku i na koniec semestru w formie pisemnej lub komputerowej na ćwiczeniach oraz 10 punktów za realizację dodatkowych zadań domowych. Do zaliczenia przedmiotu trzeba uzyskać minimum 50% punktów ze sprawdzianów oraz zadań programistycznych. Punkty są przeliczane na oceny wg następujących zakresów: 0-50 punktów – ocena 2, 51-60 punktów – ocena 3.0, 61-70 punktów – ocena 3.5, 71-80 punktów - ocena 4.0, 81-90 punktów – ocena 4.5, 91-100 punktów – ocena 5.0

**Egzamin:**

nie

**Literatura:**

\_\_

**Witryna www przedmiotu:**

-

**Uwagi:**

Oprogramowanie wykorzystywane na ćwiczeniach: zestaw kompilatorów i narzędzi GCC (MinGW), zintegrowane środowisko Eclipse CDT

## Charakterystyki przedmiotowe

### Profil praktyczny - wiedza

**Charakterystyka GI.ISP-1007\_W01:**

Zna podstawowe konstrukcje języka C++

Weryfikacja:

Kolokwium

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka GI.ISP-1007\_W02:**

Zna zasady programowania obiektowego

Weryfikacja:

Kolokwium

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka GI.ISP-1007\_W03:**

Ma podstawową wiedzę na temat sposobów oceny jakości kodu źródłowego, jego testowania i poprawiania

Weryfikacja:

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka GI.ISP-1007\_W04:**

Ma podstawową wiedzę na temat rodzajów języków komputerowych i ich przeznaczenia

Weryfikacja:

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

### Profil praktyczny - umiejętności

**Charakterystyka GI.ISP-1007\_U01:**

Potrafi zdefiniować interfejs oraz implementację klasy

Weryfikacja:

**Powiązane charakterystyki kierunkowe:** K\_U06, K\_U15

**Powiązane charakterystyki obszarowe:** I.P6S\_UU, I.P6S\_UW

**Charakterystyka GI.ISP-1007\_U02:**

Potrafi napisać program komputerowy wykorzystując techniki programowania obiektowego

Weryfikacja:

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U02, K\_U05, K\_U15

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UO, I.P6S\_UK

**Charakterystyka GI.ISP-1007\_U03:**

Potrafi skorzystać z bibliotek i ich dokumentacji

Weryfikacja:

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U02, K\_U06, K\_U15

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UO, I.P6S\_UU

**Charakterystyka GI.ISP-1007\_U04:**

Potrafi napisać program w sposób umożliwiający ponowne wykorzystanie kodu źródłowego

Weryfikacja:

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U06, K\_U15

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UU

**Charakterystyka GI.ISP-1007\_U05:**

Potrafi przetestować napisany przez siebie program

Weryfikacja:

**Powiązane charakterystyki kierunkowe:** K\_U03, K\_U04, K\_U15

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UK

### Profil praktyczny - kompetencje społeczne

**Charakterystyka GI.ISP-1007\_K01:**

Potrafi współpracować w zespole rozwiązując przydzielone problemy.

Weryfikacja:

**Powiązane charakterystyki kierunkowe:** K\_K04, K\_K01

**Powiązane charakterystyki obszarowe:** I.P6S\_KO, I.P6S\_KR, I.P6S\_KK