

**AGH**AGH UNIVERSITY OF SCIENCE  
AND TECHNOLOGY

Nazwa modułu: Programowanie obiektowe

Rok akademicki: 2015/2016      Kod: BIT-1-403-s      Punkty ECTS: 4

Wydział: Geologii, Geofizyki i Ochrony Środowiska

Kierunek: Informatyka Stosowana      Specjalność: —

Poziom studiów: Studia I stopnia      Forma i tryb studiów: Stacjonarne

Język wykładowy: Polski      Profil kształcenia: Ogólnoakademicki (A)      Semestr: 4

Strona www: —

Osoba odpowiedzialna: dr Oleksik Paweł (oleksik@agh.edu.pl)

Osoby prowadzące: dr Oleksik Paweł (oleksik@agh.edu.pl)

## Opis efektów kształcenia dla modułu zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Powiązania z EKK	Sposób weryfikacji efektów kształcenia (forma zaliczeń)
<b>Wiedza</b>			
M_W001	Rozumie paradygmat obiektowy i umie posługiwać się podstawową terminologią.	IT1A_W07	Kolokwium, Projekt
M_W002	Potrafi wskazać te elementy języków programowania, które są związane z podejściem obiektowym.	IT1A_W07	Kolokwium, Projekt
M_W003	Potrafi zaprojektować proste hierarchie klas.	IT1A_W07, IT1A_W09	Kolokwium, Projekt
M_W004	Zna zasady definiujące cykl życia obiektów, w tym reguły dotyczące konstruktorów i destruktorów.	IT1A_W07	Kolokwium, Projekt
<b>Umiejętności</b>			
M_U001	Umie zdefiniować hierarchię klas w języku C++.	IT1A_U16, IT1A_U13	Kolokwium, Projekt
M_U002	Umie w języku C++ zdefiniować obiekty zadanych typów oraz posługiwać się nimi.	IT1A_U16, IT1A_U13	Projekt, Wykonanie ćwiczeń laboratoryjnych
M_U003	Potrafi korzystać z nieobiektowych właściwości C++ (wyjątki, szablony).	IT1A_U16, IT1A_U13	Projekt, Wykonanie ćwiczeń laboratoryjnych
<b>Kompetencje społeczne</b>			

M_K001	Potrafi komunikować się z otoczeniem, jest zdolny do poszerzania wiedzy	IT1A_K06, IT1A_K01, IT1A_K04	Projekt, Wypracowania pisane na zajęciach
--------	---	------------------------------	---

## Matryca efektów kształcenia w odniesieniu do form zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Forma zajęć										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Inne	E-learning
Wiedza												
M_W001	Rozumie paradygmat obiektowy i umie posługiwać się podstawową terminologią.	+	-	-	-	-	-	-	-	-	-	-
M_W002	Potrafi wskazać te elementy języków programowania, które są związane z podejściem obiektowym.	+	-	+	-	-	-	-	-	-	-	-
M_W003	Potrafi zaprojektować proste hierarchie klas.	+	-	+	-	-	-	-	-	-	-	-
M_W004	Zna zasady definiujące cykl życia obiektów, w tym reguły dotyczące konstruktorów i destruktorów.	+	-	+	-	-	-	-	-	-	-	-
Umiejętności												
M_U001	Umie zdefiniować hierarchię klas w języku C++.	-	-	+	-	-	-	-	-	-	-	-
M_U002	Umie w języku C++ zdefiniować obiekty zadanych typów oraz posługiwać się nimi.	-	-	+	-	-	-	-	-	-	-	-
M_U003	Potrafi korzystać z nieobiektowych właściwości C++ (wyjątki, szablony).	+	-	+	-	-	-	-	-	-	-	-
Kompetencje społeczne												
M_K001	Potrafi komunikować się z otoczeniem, jest zdolny do poszerzania wiedzy	-	-	+	-	-	-	-	-	-	-	-

## Treść modułu zajęć (program wykładów i pozostałych zajęć)

### Wykład

#### Paradygmat obiektowy.

- Podstawowe pojęcia: abstrakcja, enkapsulacja, dziedziczenie i polimorfizm.
- Tworzenie hierarchii typów: uogólnienie vs. specjalizacja.
- Typy abstrakcyjne i interfejsy.

- Alternatywne podejścia: klasy jako obiekty, obiekty modyfikowane dynamicznie, prototypowanie.

#### Nieobiektywne rozszerzenia C++ względem C.

- Wzmocnienie kontroli typów; nowe typy danych, referencje.
- Przestrzenie nazw, przeładowywanie nazw funkcji.
- Parametry domyślne i parametry anonimowe.
- Dynamiczne zarządzanie pamięcią.
- Zmiany w bibliotece standardowej.

#### Podstawy programowania obiektowego w C++.

- Semantyka definicji typu (klasy).
- Zasady dostępu do składowych.
- Cykl życia instancji (obektu); metody specjalne: konstruktor i destruktor.
- Problemy z przekazywaniem obiektu przez stos: konstruktor kopiujący.
- Przeciążanie operatorów.

#### Dziedziczenie.

- Przesłanianie nazw; operator zakresu.
- Metody nie dziedziczone.
- Dziedziczenie wielobazowe.

#### Polimorfizm.

- Metody wirtualne.
- Klasy abstrakcyjne.

Obsługa wyjątków.

#### Programowanie generyczne.

- Szablony.
- Biblioteka STL.

#### Programowanie modularne w C++.

- Tworzenie bibliotek i plików nagłówkowych - zmiany względem C.

### **Ćwiczenia laboratoryjne**

#### Podstawy programowania obiektowego w C++.

- Klasy, metody, metody specjalne.
- Obsługa wyjątków.
- Dynamiczne zarządzanie pamięcią.

Testy jednostkowe. Programowanie sterowane testami.

#### Przeładowywanie operatorów.

- Podstawowe przypadki.
- Operatory specjalnie traktowane.

#### Dziedziczenie.

- Przeładowywanie i przesłanianie.
- Rzutowanie typów.
- Dziedziczenie operatorów.
- Dziedziczenie wielobazowe.

Polimorfizm. Klasy abstrakcyjne.

Elementy programowania generycznego.

### **Sposób obliczania oceny końcowej**

Ocena końcowa odpowiada ocenie z zaliczenia.

### **Wymagania wstępne i dodatkowe**

Umiejętność programowania proceduralnego.

### **Zalecana literatura i pomoce naukowe**

Bruce Eckel, „Thinking in C++”

### **Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu**

Nie podano dodatkowych publikacji

### **Informacje dodatkowe**

Podstawowym terminem uzyskania zaliczenia jest koniec zajęć w danym semestrze. Student może dwukrotnie przystąpić do poprawkowego zaliczenia.

udział „praktycznych” punktów ECTS: 3

udział „teoretycznych” punktów ECTS: 1

### **Nakład pracy studenta (bilans punktów ECTS)**

Forma aktywności studenta	Obciążenie studenta
Udział w wykładach	14 godz
Samodzielne studiowanie tematyki zajęć	17 godz
Udział w ćwiczeniach laboratoryjnych	42 godz
Przygotowanie do zajęć	47 godz
Sumaryczne obciążenie pracą studenta	120 godz
Punkty ECTS za moduł	4 ECTS