**Nazwa przedmiotu:**

Object Oriented Programming

**Koordynator przedmiotu:**

Roman Podraza

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Computer Science

**Grupa przedmiotów:**

Technical Courses

**Kod przedmiotu:**

EOOP

**Semestr nominalny:**

2 / rok ak. 2015/2016

**Liczba punktów ECTS:**

6

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

- lectures attendance: 15 x 2 h = 30 h;
- preparation to lectures (reviewing slides, notes and a textbook): 30 h;
- preparation to tests: 20 h
- laboratory attendance: 13 x 2.33 h = 30 h ;
- preparation to laboratories (learning exemplary lab tasks, reviewing slides, notes and a textbook): 40 h;

Total: 30 h + 30 h+ 20 h + 30 h+ 40 h = 150 h.

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

2

**Język prowadzenia zajęć:**

angielski

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

3

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

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

**Wymagania wstępne:**

C Programming Language

**Limit liczby studentów:**

32

**Cel przedmiotu:**

The students should learn to design and implement object-oriented programs in C++ language. They should become familiar with object-oriented paradigm, the C++ language syntax, rules and concepts and should be able to use a good programming style.

**Treści kształcenia:**

Lectures:

1. Introduction (4h): the C/C++ language elements, basic data types, literals, references, casting, operator summary, statement summary.
2. Functions and Files (4h): linkage, header files, static and automatic variables, initialization of variables, default values of function parameters, function overloading, namespaces.
3.Classes (6h): classes and members, access control, interfaces and implementation, encapsulation, friends, name qualification, static members, constructors and destructors, initialization list, copy constructor, object-oriented programming.
4.Derived Classes (6h): inheritance, virtual functions, polymorphism, pure virtual functions, abstract classes, multiple inheritance, dynamic data structures, dynamic binding, run-time type inspection.
5.Operator Overloading (2h): operator functions, assignment and initialization, increment and decrement, friends and members.
6.Templates (2h): class template, function template.
7.Exception Handling (2h): error handling, throwing and catching exceptions, try block, catch blocks, discrimination of exceptions, naming of exceptions, uncaught exceptions.
8.Input/Output Functions (2h): streams, formatting, files.
9.Standard Libraries (2h): STL, containers, algorithms, iterators.

Laboratories:

The laboratory consists of:
• Five preliminary simple tasks
1. Pointers.
2. References.
3. Classes and objects: constructors and destructors.
4. Inheritance.
5. Polymorphism (virtual functions)
• A small project requiring design of simplified simulation of an institution (e.g. bank, university, hospital). Implementation of several cooperating classes is expected as well as testing scenarios.

**Metody oceny:**

During the lab exercises it is possible to score up to 40 points:
• 15 points for first five exercises (5\*3)
• 25 points for the project
 - 8 points for the Preliminary Project (PP)
 - 9 points for the Code in C++ (CC)
 - 8 points for the Final Project (FP)
Maximum score for the mid-term test is 20 points and for final test is 40 points. There are also up to 10 short tests (during lectures), each one for 1 point.
The final result is based on the following pattern:
• 5.0: 91-110 points
• 4.5: 81-90 points
• 4.0: 71-80 points
• 3.5: 61-70 points
• 3.0: 51-60 points
• 2.0: 0-50 points

**Egzamin:**

nie

**Literatura:**

1. H. Schildt, C++: The Complete Reference, Fourth Edition, McGraw-Hill, 2003.
2. B. Eckel, Thinking in C++: Introduction to Standard C++, Volume One (2nd Edition), Prentice Hall Inc., 2000.
3. B. Eckel, Ch. Allison, Thinking in C++, Volume 2: Practical Programming, Prentice Hall, 2003.
4. B. Stroustrup, The C++ Programming Language, 4th Edition, Addison-Wesley, 2013.
5. I. Pohl, C++ for C Programmers, Addison-Wesley, 1998.

**Witryna www przedmiotu:**

https://studia.elka.pw.edu.pl

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt EOOP\_W01:**

Student has the basic systematic knowledge on designing simple computer programs and decomposing functionality between classes

Weryfikacja:

Mid-term test evaluation Final test evaluation Lab. 6.-13. evaluation

**Powiązane efekty kierunkowe:** K\_W07, K\_W09

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04, T1A\_W07, T1A\_W03, T1A\_W04, T1A\_W07

**Efekt EOOP\_W02:**

Student knows basic elements of C++ Programming Language: class, object, overloading operators, templates, inheritance, polymorphism, handling dynamic memory resources, exception.

Weryfikacja:

Mid-term test evaluation Final test evaluation Lab. 1.-13. evaluation

**Powiązane efekty kierunkowe:** K\_W07, K\_W09

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04, T1A\_W07, T1A\_W03, T1A\_W04, T1A\_W07

**Efekt EOOP\_W03:**

Student knows basic elements of environment for object-oriented programming: libraries, editor, compiler, linker.

Weryfikacja:

Lab. 1.-5. evaluation

**Powiązane efekty kierunkowe:** K\_W07, K\_W09

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04, T1A\_W07, T1A\_W03, T1A\_W04, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt EOOP\_U01:**

Student is able to design object-oriented solution of a simple typical problem and implement it in C++ Programming Language

Weryfikacja:

Mid-term test evaluation Final test evaluation Lab. 1-13. evaluation

**Powiązane efekty kierunkowe:** K\_U12

**Powiązane efekty obszarowe:** T1A\_U15, T1A\_U16

**Efekt EOOP\_U02:**

Student is able to compile, run, test and debug a program in C++ Programming Language applying object-oriented methodology

Weryfikacja:

Mid-term test evaluation Final test evaluation Lab. 1-13. evaluation

**Powiązane efekty kierunkowe:** K\_U12

**Powiązane efekty obszarowe:** T1A\_U15, T1A\_U16

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt EOOP\_K01:**

Student realizes necessity to adjust himself/herself to standards and good programming practices

Weryfikacja:

Lab. 1-13. evaluation

**Powiązane efekty kierunkowe:** K\_K04

**Powiązane efekty obszarowe:** T1A\_K03, T1A\_K04

**Efekt EOOP\_K02:**

Student understands necessity to write code, which is portable and universal, decomposed into modules, enabling its reusability

Weryfikacja:

Lab. 1-13. evaluation

**Powiązane efekty kierunkowe:** K\_K04

**Powiązane efekty obszarowe:** T1A\_K03, T1A\_K04