

Object-Oriented Programming

Iuliana Bocicor
iuliana@cs.ubbcluj.ro

Babes-Bolyai University

2017

Guiding Teachers

Object-
Oriented
Programming

Iuliana
Bocicor

- Lect. Dr. Iuliana BOCICOR
- Lect. Dr. Arthur MOLNAR
- Lect. Dr. Zsuzsanna MARIAN
- Assist. Dr. Gabriel Mircea

Activites

Object-
Oriented
Programming

Iuliana
Bocicor

- **Lecture:** 2 hours/week
- **Seminar:** 1 hour/week
- **Lab:** 2 hours/week

Course page: www.cs.ubbcluj/~iuliana/oop

Email: iuliana@cs.ubbcluj.ro. Please use your ubbcluj.ro email.

Objectives

Course Objectives:

- Object Oriented Paradigm
- C/C++ Programming Language

What you should learn from this course:

- to solve small/medium scale problems using OOP;
- to explain class structures as fundamental, modular building blocks;
- to understand the role of inheritance, polymorphism, dynamic binding and generic structures in building reusable code;
- to use classes written by other programmers;
- to use libraries (especially STL);
- to write small/medium scale C++ programs with simple graphical user interface.

Bibliography I

- ❶ B. Stroustrup. *The C++ Programming Language*, Addison Wesley, 1998.
- ❷ Bruce Eckel. *Thinking in C++*, Prentice Hall, 1995.
- ❸ A. Alexandrescu. *Programarea moderna in C++: Programare generica si modele de proiectare aplicate*, Editura Teora, 2002.
- ❹ S. Meyers. *Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd Edition)*, Addison-Wesley, 2005.

Bibliography II

- 5 S. Meyers. *More effective C++: 35 New Ways to Improve Your Programs and Designs*, Addison-Wesley, 1995.
- 6 B. Stroustrup. *A Tour of C++*, Addison Wesley, 2013.
- 7 E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Longman Publishing, 1995.
- 8 C++ reference (<http://en.cppreference.com/w/>).
- 9 Qt Documentation (<http://doc.qt.io/qt-5/>).

Course Content

Object-
Oriented
Programming

Iuliana
Bocicor

- ➊ C Programming Language Fundamentals, Functions
- ➋ Modular Programming and Memory Management
- ➌ Classes and Objects
- ➍ Templates and the C++ Standard Library
- ➎ Inheritance and Polymorphism
- ➏ I/O Streams and Exceptions
- ➐ Pointers and RAII
- ➑ Designing GUI with Qt
- ➒ Design Patterns

Rules I

- Attendance is compulsory for laboratory and seminar activities:
 - Laboratory: at least **12** out of a total of 14.
 - Seminar: at least **5** out of a total of 7.
- **Unless you have the required number of attendances, you are not allowed to take the examination this year!**

Rules II

- Another precondition to attend the examination in the regular session is to have the laboratory grade $L \geq 5$ (no rounding)! This grade is computed as shown on the next slide.
- If you do not have at least 5 for your laboratory activity, you may participate to the examination only during the re-take session. In this case, you must present all laboratory assignments and the maximum grade you can get for these is 5.

Grading

- Lab grade (**L**)
 - 50% Lab assignments
 - 50%: 3 practical tests during the labs (10%, 15%, 25%)
 - Additional work (**LA**): 0 - 0.5 (bonus)
- Written examination (**W**)
- Practical examination (**P**)
- Seminar activity (**SA**): 0 - 0.5 (bonus)

The final grade is computed as:

$$G = 0.3 \cdot L + 0.3 \cdot P + 0.4 \cdot W + SA + LA$$

To pass the examination all grades (L, W, P) must be ≥ 5 (no rounding).