



## COURSE DESCRIPTION

### 1. Program Information

1.1 University	“Alexandru Ioan Cuza”, Iasi
1.2 Faculty	Computer Science
1.3 Department	Computer Science
1.4 Field of Study	Computer Science
1.5 Type of Degree	Bachelor
1.6 Study Program / Qualification	Licentiate in computer science

### 2. Course Information

2.1 Course Title	Object Oriented Programming						
2.2 Course Coordinator	Prof. dr. DOREL LUCANU Conf. dr. DRAGOȘ GAVRILUȚ						
2.3 Seminary Coordinator	Conf. dr. DRAGOȘ GAVRILUȚ						
2.4 Study Year		2.5 Semester		2.6 Evaluation	Mixed	2.7 Course Status*	

### 3. Total estimated hours (hours per semester and didactic activities)

3.1 Hours per week	4	in which: 3.2 course	2	3.3 seminar	2
3.4 Hours in curriculum	56	in which: 3.5 course	28	3.6 seminar	28
Time Distribution					hours
Individual of study using course support, bibliography, and others					14
Individual documentation in library, using electronic tools, field work					14
Preparation of seminars/laboratories, homeworks, reports, portfolios, and essays					28
Tutoring					-
Evaluation					4
Other activities					-
3.7 Total hours individual study					56
3.8 Total hours per semester					116
3.9 Number of credits					5

### 4. Preconditions (if necessary)

4.1 Curriculum	Introduction to Programming
4.2 Competences	Procedural and imperative programming in C++

### 5. Conditions (if necessary)

5.1 For Course Operation	-
5.2 For Seminary/Laboratory Operation	-mandatory attendance



## 6. Specific Skills Acquired

<b>Professional Skills</b>	Upon successful completion of this discipline, students will be able to: C1. Explains the syntax and semantics of the C ++ language instructions. C2. Describes through UML diagrams software systems of elementary/medium difficulty level. C3. Use POO concepts and OO patterns in designing software systems. C4. Analyze the requirements of software systems of elementary/medium difficulty level. C5. Write C ++ code starting from system specifications written in UML.
<b>Transversal Skills</b>	CT1. Combining OO modeling with OO programming. CT2. Developing the ability to model practical applications in various domains using OO concepts. CT3. The ability to use UML models as a means of communicating with customers in these domains.

## 7. Course Objectives (from the grid of specific skills acquired)

<b>7.1 General Objectives</b>	Assimilation of object-oriented programming at abstract level using UML, and at the concrete level using the C ++ programming language.
<b>7.2 Specific Objectives</b>	On successful completion of this course, students will be able to: understand, explain, analyze, use: O1. classes, objects, class hierarchies, polymorphism, abstract classes, interfaces, parameterized classes; O2. modeling in UML at introductory level; O3. C ++ language (ISO Standard), with emphasis on the representation of classes, objects, and relationships between them; O4. design patterns; O5. Standard Generic Type Library (STL).

## 8. General Description

8.1	Course	Teaching Methods	Observations (hours and bibliographic references)
1.	Introduction to C ++, history, compiler, grammar	Lecture	2
2.	Objects and classes in C++	Lecture	2
3.	Operators and methods	Lecture	2
4.	Class hierarchies in C++	Lecture	2
5.	Macros and templates	Lecture	2
6.	STL I	Lecture	2



7.	STL II	Lecture	2
8.	Partial evaluation	Programming test	6
9.	Exceptions in C++	Lecture	2
10.	OO modeling	Lecture	2
11.	OO Design I	Lecture	2
12.	OO Design II	Lecture	2
13.	OO Design III	Lecture	2
14.	MVC and GUI	Lecture	2

**BIBLIOGRAPHY****(SELECTIONS)**

H. Schildt: C++ manual complet, Teora, 2000

D. Kaler, M.J. Tobler, J. Valter: C++, Teora, 2000

Bjarne Stroustrup: The C++ Programming Language, Addison-Wesley, 3rd edition, 1997

Bruce Eckel : Thinking in C++, 2nd Edition

\*\*\* : Online C++ tutorial

Donald Bell. UML basics: An introduction to the Unified Modeling Language

SGI Standard Template Library Programmer's Guide

<b>8.2</b>	<b>Seminary / Laboratory</b>	<b>Teaching/Evaluation methods</b>	<b>Observations</b> (hours and bibliographic references)
1.	Recap C: pointers, files.	Review of topics presented in the course, Individual work on a given set of exercises, Interactive methods in the development of the solutions	
2.	Experimenting C++ objects and classes	Experiment, Individual work, Interactive methods	2
3.	Experimenting operators and methods	Idem	2
4.	Experimenting class hierarchies in C++	Idem	2
5.	Experimenting macros and templates	Idem	2
6.	Experimenting STL I	Idem	2
7.	Experimenting STL II	Idem	2
8.	Test	Programming test	2



9.	Experimenting exceptions in C++	Idem	2
10.	Experimenting OO modeling	Idem	2
11.	Experimenting OO Design I	Idem	2
12.	Experimenting OO Design II	Idem	2
13.	Experimenting OO Design III	Idem	2
14.	Test	Programming test	2

**Bibliography**

the same as that for the course

**9. Course content synchronization with the expectations of the community representatives, professional associations and employers from the program domain**

--

**10. Evaluation**

Activity Type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 The weight of each evaluation form (%)
10.4 Course	<ul style="list-style-type: none"><li>- the ability to model and design a simple system</li><li>- the ability to transform a UML diagram into C ++ code</li><li>- the ability to understand and explain a C ++ code</li><li>- the quality of the answers</li></ul>	Written test	60%
10.5 Seminary	<ul style="list-style-type: none"><li>- the ability to develop POO programs that solve problems of simple and medium level</li></ul>	Programming test (30%) Lab activity (10%)	40%



	- the ability to apply OO design patterns to finding solutions - quality of written programs		
--	---	--	--

**10.6 Minimal performance standards**

The score is calculated as follows:

- A first laboratory test at week 8 (35 pct)
- A second laboratory test at the end of the semester's teaching period (25 pts) - either the end of the 14th week or the 15th week
- A final test with the subject delivered at the course (in session, 28 pts)
- Every presence in the lab means 1 point, so that it can be accumulated here up to 12 pts (there are 12 labs + 2 lab tests)

The following are needed to graduate the discipline:

- At least 20 points from the two lab tests
- At least 8 points at the final test in the subject delivered at the course
- At least 10 points out of the present

The final marks are set according to ECTS using the obtained scores.

Date  
February 10, 2018

Course Coordinator  
Prof. dr. Dorel Lucanu  
Conf. dr. Dragoș Gavriliuț

Seminary/Laboratory Coordinator  
Conf. dr. Dragoș Gavriliuț

Department Date of Approval

Director of the Department

February 15, 2018

Prof. dr. Dorel Lucanu