

COURSE DESCRIPTION

1. Program identification information

1.1 Higher education institution	Polytechnic University of Bucharest
1.2 Faculty	Electronics, Telecommunications and Information Technology
1.3 Department	Telecommunications
1.4 Field of study	Electronics and Telecommunications
1.5 Study cycle	Undergrad License (full day)
1.6 Education / Qualification Program	Telecommunication technologies and systems

2. Course identification information

2.1 Name of the discipline		Object-oriented programming					
2.2 Course Lecturer		Dr. Eng. ILIAN Virgil Liviu Mircea					
2.3 Laboratory activities		As. Drd. Eng. VOICULESCU Valentin Gabriel					
2.4 Year of study	2	2.5 Semester	1	2.6 Type of assessment	Exam	2.7 The discipline regime	mandatory

3. Total estimated time (hours per semester for academic activities)

3.1 Number of hours per week	3	from which: 3.2 course	2	3.3 practical activities	1
3.4 Total hours of the curriculum	42	from which: 3.5 course	28	3.6 practical activities	14
Distribution of Time:					hours
Study after manual, course support, bibliography and notes					16
Additional documentation in the library on specialized electronic platforms					6
Training seminars / laboratories, themes, papers, portfolios and essays					10
Tutorial					1
Examinations					3
Other activities					0
3.7 Total hours of individual study					36
3.9 Total hours per semester					78
3.10 Number of credits					3

4. Preconditions (where applicable)

4.1 of the curriculum	<ul style="list-style-type: none"> • Computer programming • Data Structures and Algorithms
-----------------------	--

4.2 competencies	<ul style="list-style-type: none"> • Application knowledge of computer programming concepts related to, the use of a programming language and its programming techniques; • Solving practical practical problems that include elements of data structures and algorithms, programming and use of microprocessors or microcontrollers
------------------	--

5. Conditions (where applicable)

5.1 for running the course	Providing optical projector (video projector) with all associated accessories (power cables, data and video signals, remote control).
5.2 for running the applications	Compulsory attendance at laboratories (according to the regulations of the undergraduate studies in UPB).

6. Specific competences accumulated

Professional skills	C3: Apply basic knowledge, concepts and methods regarding the architecture of computer systems, microprocessors, microcontrollers, programming languages and techniques.
Transversal skills	CT1: look into the problems encountered in business, identifying items for which there are dedicated solutions, providing professional tasks; CT3: Adapting to new technologies, professional and personal development through training using printed documentation sources, specialized software and electronic resources in Romanian and at least one international language.

7. Course objectives (based on the specific skills grid accumulated)

7.1 General objective of the discipline	<ul style="list-style-type: none"> - Course: <ul style="list-style-type: none"> o Absorbing object-oriented programming concepts needed for computer modeling of various problems, applications and themes usually encountered in practice. Various practical situations are analyzed and algorithms are implemented in a high level language. Programs carried out at the laboratory helps build mental reflexes and helps shape the reality in the future work of engineer students. - Applications: <ul style="list-style-type: none"> o Develop programs using C++, based on real use cases. Identifying and practicing abstraction abilities needed for computer modeling of real-life situations. Using the typical object-oriented program design concepts and standardized C++, thus providing the ability modularization of programs and preparation stages of debugging and rewriting code.
7.2 Specific objectives	Developing the ability to develop programs in an object-oriented programming language, from specification of requirements to

	execution, debugging and interpretation of results.
--	---

8. Contents

8.1 Lectures	Teaching techniques	Remarks
Fundamental concepts of object oriented programming.	Interactivity with the students through the applicative part associated (specific examples using the C++ syntax). Intervals are reserved for presenting, analyzing and solving practical problems (modeling reality). The modeling part is often reduced to solving relatively simple problems but demanding immediate results. Solving such problems during course lectures provides the necessary support of understanding the object-oriented programming concepts. Dialogue during the lectures is also extended through consulting sessions. These are necessary in preparing students for the continuous assessment tests.	3 hours
C++ characterization (compared to ANSI C). Concrete examples using C++ specific syntax.		2 hours
Classes and Objects. Static-dynamic distinction in terms of the effects of a program. Description of object concepts on classes member functions, constructors, destructors, data hiding, abstraction.		2 hours
Member Functions, States (methods) functional details. Examples. Data Access Specificators.		2 hours
Constructors: details. Implied constructors with parameters, copy constructors. Examples. Destructors. Examples. Friend functions. Examples.		3 hours
C++ references. This pointer. Pointers to Classes. Static vs. dynamic in such a context. Examples.		2 hours
Static members of a class. Static functions.		2 hours
Inheritance: introduction. Specific concepts and rules.		3 hours
Inheritance: continued. Simple inheritance. Data access specifiers and methods. Inheritance of the public, protected or private type.		2 hours
Multiple inheritance. Polymorphism. Overloading functions. Early and late binding. Virtual methods.		3 hours
Interface inheritance vs. implementation inheritance. Overloading operators. Superchargeable operators.		2 hours
Review: examples of exercises and issues.		2 hours
Bibliography <ul style="list-style-type: none"> - S. B. Lippman, J. Lajoie, B. E. Moo- <i>C++ Primer</i>, Addison-Wesley, 2012 - Bjarne Stroustrup- <i>Programming: Principles and Practice Using C++</i> Addison-Wesley, 2008 - Andrew Koenig- <i>Accelerated C++: Practical Programming by Example</i> Addison-Wesley, 2000. - Danny KALEV, Michael TOBLER, Jan WALTER - <i>C++</i>, Waite Group, January 1999. 		

8.2 Practical applications	Teaching techniques	Remarks
Review the structure of a program. Testing of programming skills (short test). Comparison between procedural programming (structured) and object oriented programming.	The brief review of the theoretical notions specific to the language. Students attempt to write functional programs from the pseudo-code shown in laboratory platform. The language is ANSI C++ (-std = c++ 11). The programming environment and laboratory material platforms are available for the students in electronic format.	2 hours
Introduction to classes and objects		2 hours
Classes and Objects (cont.). Copy constructors. Friend functions.		2 hours
Copy Constructors (cont.) . Destructors. Functions with object argument and object references.		2 hours
Copy Constructors (cont.). Destructors (cont). Simple inheritance.		2 hours
Multiple inheritance. Polymorphism.		2 hours
Final verification laboratory		2 hours
Bibliografie - Brian OVERLAND - C++ - <i>A beginners guide without fear</i> , Prentice Hall/Pearson / <i>Ghid pentru incepatori</i> , Ed. Corint, 2006. - Herbert SCHILDT - C++ - <i>The complete reference</i> ;Mcgraw Hill/Osborne / <i>Manual complet</i> , Ed. Teora, 2001 - http://itlectures.ro/ro_RO/2017/09/01/object-oriented-programming-english/		

9. Bridging the course content with the expectations of the epistemic community representatives, professional associations and employers representatives for the domain of the program

Programming skills are widely sought on the labor market. Object-oriented programming is a central paradigm of modern programming, and the understanding and mastering of its principles provides not only an advantage for those who will focus on designing and developing software, but also, in a wider context, a deeper insight into understanding and design of complex systems in various fields.

It provides the graduates with the appropriate skills with the current qualification requirements and with a modern, high quality and competitive scientific and technical training that will allow them to be hired after graduation, being perfectly integrated into the politics of the Polytechnic University of Bucharest both in terms of content and structure, as well as in terms of international aptitudes and openness to students.

Appropriate use of evaluation criteria and methods in accordance with the European academic norms to which POLITEHNICA Bucharest is a part of the university allows students to self-assess continuously, based on the qualifications obtained and taking into account the observations and methodological indications that the course / laboratory offers.

10. Evaluation

Activity type	10.1 Evaluation Criteria	10.2 Evaluation methods	10.3 Weight in Final
---------------	--------------------------	-------------------------	----------------------

			mark
10.4 Lectures	<p>Correct identification of contexts theoretical and practical application of object-oriented programming concepts. Formation thinking as object-programming paradigm.</p> <p>C4.1: Define concepts, principles and methods used in the fields of computer programming, high-level languages and specific architecture computer systems, programmable electronic systems, graphics, software reconfigurable architectures.</p>	<p>Execution of individual written examination examination (total: 50p).</p> <p>We investigate both the students ability to formalize a problem as well as the ability to translate specific concepts into the program.</p> <p>The subjects cover both the theoretical part related to the definition of the concepts of objectual programming, as well as the practical part, from the perspective of the solving ability with the C ++ language of some programming problems.</p>	50%
10.5 Practical applications	<p>C4.5: Execute and pass a test on the architecture and functional principles of a functional software structure.</p> <p>C6.5: Executing a test on establishing and describing the operations required to build and test a specific algorithm, based on the object-oriented programming paradigm.</p>	<ul style="list-style-type: none"> - The laboratory activity is constantly checked, throughout the semester, by means of an individual sample to the computer: 15p - Two problems proposed to be solved in the form of a homework: 5p - The lab terminates with a final, individual check at the station. This is based on the implementation of a problem covering the whole semester: 30p. 	50%
10.6 Minimal performance standard:			
The accumulation abilities are checked and then the ability to identify the practical situations specific to the presented methods as well as their correct application in the electronic domain is verified.			

Promotion of subject implies the accumulation (without intermediate imposed thresholds) of at least 50p of the total of 100p.

Date of completion
25.09.2017

Signature of course and applications holders,
Dr. Eng. ILIAN Virgil Liviu Mircea

As. Drd. Eng. VOICULESCU Valentin Gabriel

Date of approval
25.09.2017

Signature Director of Department,
Conf. Dr. Eng. Marian VLĂDESCU