

RTU Course "Object-Oriented Programming"

12306 Department of Applied Computer Science

General data

Code	DPI230
Course title	Object-Oriented Programming
Course status in the programme	Compulsory/Courses of Limited Choice
Course level	Undergraduate Studies
Course type	Academic
Field of study	Computer Science
Responsible instructor	Uldis Sukovskis
Academic staff	Pāvels Rusakovs Igoris Ščukins Gundars Alksnis
Volume of the course: parts and credits points	1 part, 3.0 Credit Points, 4.5 ECTS credits
Language of instruction	LV, EN
Annotation	This course covers object-oriented (OO) programming concepts and techniques using C++ language as an example. After completing the course students will understand the basic principles of OO programming and obtain OO programming skills. OO basic concepts - abstraction, encapsulation, inheritance and polymorphism are discussed as well as practical programming issues, including class hierarchy formation, an exception handling, templates, etc. Comparison of C++ with some other OO programming languages (eg C # or Java) is also given. During laboratory work students must demonstrate theoretical knowledge and acquire programming skills.
Goals and objectives of the course in terms of competences and skills	To understand and be able to explain the principles of object-oriented programming and technology, being able to apply these principles for the software development process using object-oriented programming language and visual programming environments.
Structure and tasks of independent studies	To use literature sources, to prepare and execute individual tasks in computer lab
Recommended literature	Bjarne Stroustrup. The C++ Programming Language: Special Edition. – Addison-Wesley Professional, 2000. ISBN-13: 978-0201700732, 1030 lpp. Booch, G., etc.: Object-Oriented Analysis and Design with Applications. 3rd Ed. Addison-Wesley, 2007. Ira Pohl. Object-Oriented Programming Using C++ (2nd Edition). – Addison-Wesley Professional, 1996. ISBN-13: 978-0201895506, 576 lpp. Sutter, H., Alexandrescu, A.: C++ coding standards: 101 rules, guidelines, and best practices. Addison-Wesley, 2007. Eckel, B.: Thinking in C++. 2nd Ed. Prentice Hall, 2000. Internets: http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html Soulie, J.: C++ Language Tutorial. 2009. Internets: http://www.cplusplus.com/doc/tutorial/
Course prerequisites	Knowledge in structural programming (preferably C).

Course outline

Theme	Hours
Software life cycle. Object-oriented method for analysis and programming. C++ program development.	2
Basic concepts of object-oriented programming: object, message, class, instance, method and features - abstraction, etc.	2
Abstraction. Classes and class members.	2
Constructors and destructors. Visibility scope.	2
Hierarchy. Derived classes. Multiple inheritance.	2
Virtual functions. Pure virtual functions and abstract classes.	2
Static class members. Class friends.	2
Polymorphism. Operation overloading.	2
Templates.	2
Exception handling.	2
Programming techniques using C++.	10
Introduction to other OOP language with comparison to C++.	2
Laboratory work.	16

Learning outcomes and assessment

Learning outcomes	Assessment methods
Is able to explain the OO concepts and demonstrate their implementation in C++.	Passed tests and examination.
Is able to independently implement the program in C++ using object oriented programming technology, to explain the program and statements used.	Successfully completed and defended laboratory works.
Is able to analyze the C++ programs for the structure and operation.	Passed tests and examination.
Is able to explain features of object-oriented and structural programming.	Passed tests and examination.

Study subject structure

Part	CP	Hours per Week			Tests		
		Lectures	Practical	Lab.	Test	Exam	Work
1.	3.0	2.0	0.0	1.0		*	