I068	**Advanced programming techniques**	L	P	S	ECTS
		2	2	1	7

Course objectives. Introduce students with design patterns and how they can help in design object-oriented software. Students will design case study that will demonstrate how design patterns apply in practice. Special emphasis will be on practical implementation of such concepts in C++ language.

Prerequisites. Undergraduate university study programme of mathematics and/or computer science.

Course content.

1. Introduction. What is design pattern? Describing design patterns. How design patterns solve design problems? How to select a design pattern?
2. SOLID principles. Boost library.
3. Creational design patterns: abstract factory, builder, factory method, prototype, singleton.
4. Structural design patterns: adapter, bridge, composite, decorator, façade, flyweight, proxy.
5. Behavioral patterns: chain of responsibility, command, interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.

LEARNING OUTCOMES

No.	LEARNING OUTCOMES
1.	Understand concept of design pattern.
2.	Understand SOLID principles and being able to apply them in practice.
3.	Being able to create your own example of creational design pattern.
4.	Being able to create your own example of structural design pattern.
5.	Being able to create your own example of behavioral design pattern.

RELATING THE LEARNING OUTCOMES, ORGANIZATION OF THE EDUCATIONAL PROCESS AND ASSESSMENT OF THE LEARNING OUTCOMES

TEACHING ACTIVITY	ECTS	LEARNING OUTCOME **	STUDENT ACTIVITY*	EVALUATION METHOD	POINTS	
					min	max
Attending lectures and exercises	1	1-5	Lecture attendance, discussion, teams work, independent work on given tasks and short written exams	Attendance lists, tracking activities, closed form exercises	0	4
Homework assignments	1	1-5	Independent work on given problems	Evaluation	0	4
Written exam (Mid-terms)	2	1-5	Preparing for written exam	Evaluation	25	46
Final exam	2	1-5	Revision	Oral exam	25	46
TOTAL	6				50	100

Teaching methods and student assessment. Lectures and exercises are obligatory. The exam consists of a written and an oral part. Upon completion of the course, students can take the exam. Successful midterm exam scores replace the written exam. Exercises are both auditory and laboratory. Laboratory exercises include the usage of computers. Students can improve their grades by writing homework assignments and seminars. By the end of the class the practical project will be defined.

Can the course be taught in English: Yes

Basic literature:

1. E Gamma, R. Helm, R. Johnson, J. Vlissides, G. Booch, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.
2. A. Alexandrescu, Modern C++ Design: Generic Programming and Design Patterns Applied, Addison-Wesley, 2001.