

Faculty of Electrical Engineering / ELECTRONICS, TELECOMMUNICATIONS AND COMPUTERS / OBJECT-ORIENTED SOFTWARE DESIGN

Course:	OBJECT-ORIENTED SOFTWARE DESIGN			
Course ID	Course status	Semester	ECTS credits	Lessons (Lessons+Exercises+Laboratory)
8641	Mandatory	2	6	3+1+0
Programs	ELECTRONICS, TELECOMMUNICATIONS AND COMPUTERS			
Prerequisites	No prerequisites required.			
Aims	Describing methodological problems regarding central concepts of object-oriented software design. Familiarizing students with techniques of object-oriented software design. Introduction to Java programming language.			
Learning outcomes	After passing this exam, student will be able to: 1. Differentiate between the structural and object-oriented programming, as well as to explain the basic principles of object-oriented programming. 2. Explain the concept of class as well as the associated concepts of inheritance and polymorphism. 3. Creates a program (project) in the Java programming language, which includes entry, processing and printing of data. 4. Creates a program (project) in the Java programming language, which involves working with multiple classes. 5. Creates a program (project) in the Java programming language, which includes working with graphic forms. 6. Explain the characteristics of collections in the Java programming language.			
Lecturer / Teaching assistant	Ph.D. Prof. dr Slobodan Đukanović – teacher M.Sc. Stefan Vujović – assistant			
Methodology	Lectures and exercises. Individual work on practical tasks, consultations.			
Plan and program of work				
Preparing week	Preparation and registration of the semester			
I week lectures	Introduction. About quality of the software. The need for object-oriented design and criteria for object orientation.			
I week exercises	About Eclipse software.			
II week lectures	Modularity. Reusable software.			
II week exercises	Creating class. Working with classes.			
III week lectures	Functional and object decomposition. Abstract data types.			
III week exercises	Working with strings. Collections in Java part I.			
IV week lectures	Classes and Objects.			
IV week exercises	Working with files. Regular expressions.			
V week lectures	Memory management. Genericity.			
V week exercises	Polymorphism. Inheritance. Creating interfaces.			
VI week lectures	First test.			
VI week exercises	First test.			
VII week lectures	Design by contract. Assertions. Exception handling.			
VII week exercises	Working with Forms. Graphical user interface - GUI.			
VIII week lectures	Inheritance. Polymorphism. Dynamic linking.			
VIII week exercises	Collections in Java part II.			
IX week lectures	Multiple inheritance. Techniques of inheritance.			
IX week exercises	Advanced options in Java. The connection to the server, upload and download.			
X week lectures	Introduction to Java programming language. Introduction to the programming environment.			
X week exercises	Creating of Android applications in Java. Plugins for Eclipse.			
XI week lectures	Second test.			
XI week exercises	Second test.			
XII week lectures	Implementation of OO concepts in Java.			
XII week exercises	Basic classes for creating Android applications. Activity.			

XIII week lectures	Working with collections in Java.					
XIII week exercises	Design of Android applications. XML layout.					
XIV week lectures	Working with forms in Java.					
XIV week exercises	Advanced options in the creation of Android applications. Databases. HTTP connection.					
XV week lectures	Test correction					
XV week exercises	Test correction					
Student workload	Per week: Working hours: 6 credits x 40/30 = 8 hours. Working hours structure: 3 hours for teaching 1 hour for exercises 4 hours for individual work, including consultations. Per semester: Teaching and the final exam: (8 hours) x 16 = 128 hours. Necessary preparation (before semester): 2 x (8 hours) = 16 hours. Total work hours for the course: 6 x 30 hours = 180 hours Additional hours for preparing correction of the final exam, including the exam taking: up to 36 hours. Work hours structure: 128 hours (lectures) + 16 hours (preparation) + 36 hours (additional work)					
Per week			Per semester			
6 credits x 40/30=8 hours and 0 minuts 3 sat(a) theoretical classes 0 sat(a) practical classes 1 excercises 4 hour(s) i 0 minuts of independent work, including consultations			Classes and final exam: 8 hour(s) i 0 minuts x 16 =128 hour(s) i 0 minuts Necessary preparation before the beginning of the semester (administration, registration, certification): 8 hour(s) i 0 minuts x 2 =16 hour(s) i 0 minuts Total workload for the subject: 6 x 30=180 hour(s) Additional work for exam preparation in the preparing exam period, including taking the remedial exam from 0 to 30 hours (remaining time from the first two items to the total load for the item) 36 hour(s) i 0 minuts Workload structure: 128 hour(s) i 0 minuts (cources), 16 hour(s) i 0 minuts (preparation), 36 hour(s) i 0 minuts (additional work)			
Student obligations			Lessons attendance is mandatory for students, as well as doing home exercises and test.			
Consultations			Every day, in agreement with teacher or assistant.			
Literature			Presentations for the lectures Bertrand Meyer, "Object-oriented software construction", Prentice Hall. Erich Gamma et al, "Design Patterns: Elements of Reusable Object-Oriented Software ", Addison-Wesley Professional, 1st edition. Paul Deitel and Harve			
Examination methods			The forms of knowledge testing and grading: - Two tests carry 2x35= 50 points. - Final exam carries 40 points. - Oral exam (optional). Student gets the passing grade by collecting 50 points at least.			
Special remarks						
Comment						
Grade:	F	E	D	C	B	A
Number of points	less than 50 points	greater than or equal to 50 points and less than 60 points	greater than or equal to 60 points and less than 70 points	greater than or equal to 70 points and less than 80 points	greater than or equal to 80 points and less than 90 points	greater than or equal to 90 points