

Faculty of Electrical Engineering / APPLIED COMPUTER ENGINEERING / Programming language II

Course:	Programming language II			
Course ID	Course status	Semester	ECTS credits	Lessons (Lessons+Exercises+Laboratory)
10696	Mandatory	3	6	3+0+2
Programs	APPLIED COMPUTER ENGINEERING			
Prerequisites	No prerequisites required.			
Aims	To introduce students to basics of object-oriented programming, work with integrated environment, event-driven programming, C++ programming language syntax, visual components library (VCL), structures and classes, principles of object-oriented programming, database and network communication application creating.			
Learning outcomes	After the student passes this exam, he will be able to: understand the basic concepts of object oriented and event-driven programming, develop a simple object oriented application, develop a simple database and network communication application.			
Lecturer / Teaching assistant	Prof. Budimir Lutovac PhD, Boris Marković MSc			
Methodology	Lectures, exercises, studying and doing home exercises. Consultations.			
Plan and program of work				
Preparing week	Preparation and registration of the semester			
I week lectures	Introduction. Basics of object-oriented programming.			
I week exercises	Introduction. Basics of object-oriented programming.			
II week lectures	Work with integrated environments for object oriented and visual programming.			
II week exercises	Work with integrated environments for object oriented and visual programming.			
III week lectures	Event-driven programming.			
III week exercises	Event-driven programming.			
IV week lectures	C++ programming language syntax, variables, operators, Input/Output.			
IV week exercises	C++ programming language syntax, variables, operators, Input/Output.			
V week lectures	Control flow statements, functions, pointers, references.			
V week exercises	Control flow statements, functions, pointers, references.			
VI week lectures	Arrays, files, dynamical memory allocation. Visual components library (VCL)			
VI week exercises	Arrays, files, dynamical memory allocation. Visual components library (VCL)			
VII week lectures	Structures and pointers to structures, queues, stacks, adding functions to structures.			
VII week exercises	Structures and pointers to structures, queues, stacks, adding functions to structures.			
VIII week lectures	Principles of object-oriented programming - objects and classes.			
VIII week exercises	Principles of object-oriented programming - objects and classes.			
IX week lectures	Encapsulation and abstraction.			
IX week exercises	Encapsulation and abstraction.			
X week lectures	Constructors and destructors. Classes and friends.			
X week exercises	Constructors and destructors. Classes and friends.			
XI week lectures	Midterm exam.			
XI week exercises	Midterm exam.			
XII week lectures	Inheritance.			
XII week exercises	Inheritance.			
XIII week lectures	Virtual functions as class members, polymorphism.			
XIII week exercises	Virtual functions as class members, polymorphism.			
XIV week lectures	Creating a database application, Creating a network communication application (TCP/IP and UDP/IP)			

XIV week exercises	Creating a database application, Creating a network communication application (TCP/IP and UDP/IP)					
XV week lectures	Remedial midterm exam.					
XV week exercises	Remedial midterm exam.					
Student workload	3 hours of lectures, 2 hours of exercises, 3 hours of independent work, including homework and consultations.					
Per week			Per semester			
6 credits x 40/30=8 hours and 0 minuts 3 sat(a) theoretical classes 2 sat(a) practical classes 0 excercises 3 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			Regular attendance at classes, attending knowledge tests (colloquium and final exam).			
Consultations			After the lecture, and if necessary, by appointment.			
Literature			Bjarne Stroustrup, "The C++ Programming Language", and Lecture material available at a distance learning platform.			
Examination methods			The midterm exam 50 points. The final exam 50 points. A passing grade is obtained if at least 50 points are collected.			
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