

DESCRIPTION OF A STUDY COURSE – SYLLABUS

Title of a course	Programming				
Head of course	MSc Jasminka Tomljanović, Senior Lecturer				
Study programme	Professional undergraduate study Information Science				
Status of a course	Obligatory				
Year of study	2	Semester	III	ECTS credits	6
Teaching plan (L + E + S+ Pr)	1+0+3+0				
Goals of a course					
The objective of the course is to make students capable to design, code, test, correct and document software solutions in an object-oriented C ++ programming language. The target level of program complexity is multi-class programs, with basic static and dynamic complex data structures, and file management.					
Conditions for enrolling course					
Passed College Programming Basics					
Learning outcomes on a level of a study programme which includes course					
Outcome 2: Apply business information system design methods. Outcome 4: Develop an application solution for the Internet and desktop environment. Outcome 5: Apply mathematical and statistical methods in information science. Outcome 12: Apply engineering methods and principles in information science.					
Expected learning outcomes on a level of a course					
<ol style="list-style-type: none"> Describe and interpret the operation of the recursive function Create object-oriented applications that include multiple classes, work with basic static and dynamic data structures, including linked lists, work with files Acquire an introduction to advanced programming techniques and object modelling with the UML technique Explain the basic concepts of object-oriented programming (encapsulation, data hiding, inheritance and polymorphism) Apply the basic concepts of object-oriented programming (encapsulation, data hiding, inheritance and polymorphism) 					
Content of a course					
<p>Introduction to C++ language. Modular programming and functional decomposition of a program. Lexical pre-processor. Characteristics of structured and object-oriented programming. Modelling and implementation of programmes in C++language: form, function and simple types of data. Class interface, abstraction and implementation. Operator super-ordination. Polymorphism and inheritance. Abstract class and generic class. Algorithms.</p> <p>Basic program structure. Working with databases. Life cycle of a program, methods of developing programming products.</p> <p>Encoding in C++, declaration, simple commands, mathematical operations, strings, matrixes, functions, uses of functions, processing, control instructions. Program structure. Developing a program with GUI Iteration with operating system. Process handling.</p>					
Teaching modes	<input checked="" type="checkbox"/> lectures <input type="checkbox"/> auditory exercises <input checked="" type="checkbox"/> seminars and workshops <input type="checkbox"/> distance learning <input type="checkbox"/> field classes		<input checked="" type="checkbox"/> individual assignments <input type="checkbox"/> multimedia and network <input type="checkbox"/> laboratory <input type="checkbox"/> supervisor's work <input type="checkbox"/> other _____		
Comments					
Students' obligations					
Successfully complete exercise and self-study tasks.					

Grading, evaluation and monitoring of students' work continuously during lectures and exams

Grading is based upon evaluation of of course's learning outcomes' adoption. Grading is performed continuously during lectures and/or during exam, in compliance with the provisions of Regulation on the assessment of students.

Continuous check-up:

Outcomes	Pre-exam I	Pre-exam 2	Assignment	Threshold	Max
Outcome 1	20%		2%	11 %	22%
Outcome 2	25%	20%	2%	23,5 %	47%
Outcome 3		10%	2%	6 %	12%
Outcome 4		10%	2%	6 %	12%
Outcome 5		5%	2%	3,5 %	7%
Percentage of ECTS	2,5%	2,5%	1%		
Total	45%	45%	10%	50 %	100 %

A student has passed the exam if he has acquired a percentage of credits for each learning outcome higher or equal to defined threshold.

Exam term:

Outcomes	Written exam	Oral exam	Max
Outcome 1	10%	10%	20%
Outcome 2	25%	15%	40%
Outcome 3	15%	5%	20%
Outcome 4	5%	5%	10%
Outcome 5	5%	5%	10%
Percentage of ECTS	4	2	
Total	60%	40%	100 %

A student has passed the exam if he has acquired a percentage of credits for each learning outcome higher or equal to defined threshold.

Grading:

A student has passed the exam if he has acquired at least 50% of anticipated credits of a specific learning outcome.

If a student has passed learning outcomes of all courses, the accomplished credits (percentages) of all passed learning outcomes are being added, while the final grade is defined upon following table:

Range of credits (percentages)	Numerical grade	ECTS grade
90,00 – 100,00	Excellent (5)	A
75,00 – 89,99	Very good (4)	B
60,00 – 74,99	Good (3)	C
50,00 – 59,99	Sufficient (2)	D
0,00 – 49,99	Insufficient (1)	F

Obligatory literature

1. Jasminka Tomljanović, Programiranje, Veleučilište u Rijeci
2. Julijan Šribar, Boris Motik: Demistificirani C++, peto dopunjeno izdanje. Element, Zagreb, 2018
3. Bruce Eckel, Thinking in C++, Volume 2, 2016

Additional literature

1. R. Sedgewick, Algorithmus in C++, Addison-Wesley Professional, 1993.
2. N. Wirth, Algorithmus+Data Structures=Programs, Prentice Hall, 1976
3. B. Stroustrup, The C++ Programming Language, Pearson Education Inc., Upper Saddle River, 2013

