DESCRIPTION OF A STUDY COURSE – SYLLABUS

Title of a course	**Programming II**				
Head of course	**Associate Professor, PhD Alen Jakupović**				
Study programme	**Professional undergraduate study Telematics**				
Status of a course	Obligatory				
Year of study	2.	**Semester**	III	**ECTS credits**	6
Teaching plan (L + E + S+ Pr)	2+3+0+0				

Goals of a course

Acquire competencies for designing software systems using UML and implementing them in an object-oriented programming language.

Conditions for enrolling course

No conditions

Learning outcomes on a level of a study programme which includes course

Outcome 4: Use computer principles and methods related to the architecture and organization of computers and computer networks.
Outcome 5: Use computer principles and methods related to programming languages, databases, and operating systems.
Outcome 6: Design and implement desktop, web and mobile computer applications and computer programs for microcomputers and microcontrollers, with or without a database.
Outcome 10: Analyse and implement an information system in the field of telematics.
Outcome 15: Participate in teamwork and independently present professional content in written and spoken form in Croatian and English.

Expected learning outcomes on a level of a course

1. Describe basic concepts related to object-oriented design
2. Describe basic concepts related to object-oriented programming
3. Create the following UML diagrams: usage diagram, activity diagram, class diagram and sequence diagram
4. Implement classes in Java programming language using MVC architecture according to UML diagrams
5. Compile project documentation

Content of a course

Introduction into object-oriented languages (C++, C#, Java) and division into procedural and functional languages; Objects and classes (structure, syntax, encapsulation, principle of privacy); Object-oriented analysis (OOA); Object-oriented design (OOD); Object-oriented modelling with UML (aspects and diagrams, class diagram, diagram of the application case, component diagram, chart of division, diagram of state, sequence and collaboration, application of UML tools, GO and Poseidon); Attributes (constants, variables, parameters, accessibility, right of access, class and object attributes); Operations (call – by – value, call – by- reference, class methods in objects etc.); Object associations (aggregation, composition, cardinality, aspects of association implementation); Control structures (arrays, choice, multiple choice, iteration, structural programming); applications and applets; documentation and guidelines for making a source code, source code documentation.

Teaching modes	☒ lectures ☐ auditory exercises ☒ seminars and workshops ☐ distance learning ☐ field classes	☒ individual assignments ☐ multimedia and network ☐ laboratory ☐ supervisor's work ☐ other ____________________
Comments		

Students' obligations

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

Grading is based upon evaluation 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	Written test	Project defense	Practical assignments	Threshold	Max
Outcome 1	10 %			**5 %**	**10 %**
Outcome 2	10 %			**5 %**	**10 %**
Outcome 3	10 %	25 %		**12,5 %**	**35 %**
Outcome 4		10 %	25 %	**12,5 %**	**35 %**
Outcome 5		10 %		**5 %**	**10 %**
Percentage of ECTS	**1,8**	**2,7**	**1,5**	**-**	**-**
Total	**30 %**	**45 %**	**25 %**	**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 %**
Outcome 2	10 %		**10 %**
Outcome 3	10 %	25 %	**35 %**
Outcome 4		35 %	**35 %**
Outcome 5		10 %	**10 %**
Percentage of ECTS	**1,8**	**4,2**	**-**
Total	**30 %**	**70 %**	**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. Authorized Lectures

Additional literature

1. Britton, C., Doake, J.: A Student Guide To Object-Oriented Development, Elsevier, 2005.
2. O'Docherty, M.: Object-Oriented Analysis and Design – Understanding System Development with UML2.0, John Wiley & Sons, 2005.
3. Lethbridge, T. C., Langaniere, R.: Object-Oriented Software Engineering – Practical Software Development using UML and Java, The McGraw-Hill Education, 2005.
4. Eckel Bruce, Thinking in Java, 4th Edition, President, MindView, Inc., 2006. (highscore.googlecode.com/files/Thinking%20In%20Java%204th.pdf)