Basic data of the subject	
University:	**University of Applied Sciences in Ferizaj**
Academic unit:	**Faculty of Engineering and Informatics**
Program:	**Applied Informatics**
Title of the subject:	**Software Engineering**
Level:	**Bachelor**
Course Status:	**Obligatory**
Year of studies:	**III, Semester V**
Number of hours per week:	**3**
Value of Credits - ECTS:	**5**
Time / location:	
Course lecturer:	
Contact details:	
Course Description:	*The course provides students with fundamental knowledge of program engineering methods, tools and their application in the design and development process of engineering systems. Students are taught to design a system from the projects vision and requirement documentation to full implementation and support, i.e. during this module project proposal, requirements document, system architecture document are being prepared, system testing and evaluation are done.*
Objectives of the course:	*The aim of the course – introduce the student to the methods and measures of Program Engineering and its application in the creation of program products.*
Expected learning outcomes:	*Upon successful completion of this course, student will be able to:* • *Use the main concepts, standards of programs engineering and definition of software process.* • *Assimilation of the creation stages of software and systems, lifetime cycles and its models, perform the network graphic analysis, applying the methods of requirements specification and validation during the development of program systems.* • *Use effectively the CASE programming measures of automatic construction and documentation.* • *Understand the processes of requirements specification and analysis, ability to model dynamics and static structure of the subject field, assimilation of object oriented UML modelling language.* • *Communicate with other people in the working group, organize its own work and perform the planned activities in time gaining the ability to work in a team.* • *Apply the methods of requirements specification and validation during the development of program systems.*

Prerequisites:	*Basic knowledge in the field of computer engineering and programming.*

Contribution to the student load (which must correspond with learning outcomes)			
Activity	**Hour**	**Day/Week**	**In total**
Lectures with numerical exercises	3	15	45
Internship			
Contacts with teacher / consultations			
Field exercises			
Midterm, seminars and projects.	3	2	6
Homework			
Self-learning time student (at the library or at home)	3	15	45
Final preparation for the exam	7	2	14
Time spent on evaluation (tests, quiz and final exam)			
Projects and presentations.	3	5	15
Total			**125**

Teaching methodology:	*The course takes 15 weeks with 1.5 hours of lectures and 1.5 hours weekly individual and group exercises.* *Exercises will be held in the form of individual and group work in which concrete examples will be discussed.* *Active participation is extremely important so students are encouraged to attend lectures and exercises regularly and contribute to the discussions that take place in lectures. Lectures, exercise, individual work, discussions and group work.*
Assessment methods:	*The student can choose to be assessed one of the two forms of assessment, given below:* *1. Form 1: Evaluation with colloquiums and project* *2. Form 2: Evaluation with the final exam.* ***Form 1:*** *In the first form of assessment "Assessment with colloquiums and project" the student is assessed in four activities that are carried out during the lectures:* *1. Colloquium 1 (35%), individual assessment* *2. Colloquium 2 (35%), individual assessment* *3. Class activity (10%), individual assessment* *4. Project (20%), group assessment.* *If the student is not satisfied with the assessment achieved according to form 1, then he can undergo the assessment according to form 2 to obtain a higher assessment.* ***Form 2:***

	Through the final exam, the student can achieve a maximum of 70% of the points from the total of 100 points. *The rest of the 20% points must be completed by group work in the Project, an activity carried out during the lectures.* *In Colloquium 1, Colloquium 2 and the final exam, the evaluation of the students will be done through an evaluation form, which must be completed individually by the student. The evaluation form will contain 5 tasks through which the student's learning outcomes will be evaluated.* *Activity in the class means the student's engagement in dealing with the issues discussed in the class, during the lectures.* *Project (20%), group assessment: it is an activity in which students apply the acquired knowledge in a concrete project. It is carried out in groups of 3 or 4 students who are obliged to carry out the activity, document and present it to the subject professor.* ***Rating:*** *91-100 points – graded 10 (ten)* *81-90 points – graded 9 (nine)* *71-80 points – grade 8 (eight)* *61-70 points – grade 7 (seven)* *51-60 points – grade 6 (six)* *0-50 points – The student repeats the exam*
The ratio of theory and practice:	*70% theory and 30% practice.*
Literature	
Basic Literature:	1. *PRESSMAN, Roger S.; and MAXIM, Bruce R. (2014). Software Engineering: A Practitioner's Approach. Boston, Mass: McGraw-Hill, 8th edition, 976 p. ISBN 978-0078022128.*
Additional Literature:	2. *SOMMERVILLE, Ian (2010). Software Engineering. Harlow: Pearson Education, 9th edition, 792 p. ISBN 978-0137035151.*
Designed learning plan	
Week:	**Lectures and exercises to be held**
Week one:	*Introduction to the Course.*
Week two:	*Definition of program engineering (PE). Management of software projects. Analysis and design stages of a programming product: processes, methods and means.*
Week three:	*Object-oriented modelling. Object-oriented design – UML language: use case diagram. Purpose of requirements engineering.*

Week four:	*Conversation with the customer: dialogue, interview, etc. CASE tools.*
Week five:	*Object-oriented design – UML language: classes' diagram.*
Week six:	*Design of program architecture.*
Week seven:	*Test 1*
Week eight:	*Modelling of UML interface: sequences and communication diagrams. Object-oriented design – UML language: realization diagrams: components and composition models.*
Week nine:	*Stage of programming realizations (of coding and code generation) and works integration.*
Week ten:	*Techniques of software testing and evaluation.*
Week eleven:	*The concept of project management.*
Week twelve:	*Agile methods in Software Engineering.*
Week thirteen:	*Improvement of a project.*
Week fourteen:	*New technologies of program engineering.*
Week fifteen:	*Test 2*
Academic policies and rules of conduct	
Regular attendance of lectures and exercises is necessary, as well as active participation with discussion and solution of tasks. Not impeding the progress required for learning using mobile phones turned off or in silent mode.	