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 Testing and Quality Assurance**
Level:	**Bachelor**
Course Status:	**Elective**
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:	*This course provides you with knowledge and skills in the field of software testing and quality assurance. Through it, students become familiar with the testing cycle, quality verification methods, and black box testing techniques. It also discusses unit, integration, system, and acceptance testing of software. Lectures also include code analysis, white box testing techniques, and state transition testing. By the end of the course, students will have gained deep knowledge in the field of software testing and quality assurance.*
Objectives of the course:	*The aim of this course is to cover the fundamental knowledge and skills in software testing and ensuring its quality.*
Expected learning outcomes:	*Upon successful completion of the course, students will be able to:* • *Describe and apply the phases of the software testing cycle, including planning, specification, development, execution, evaluation, and reporting of testing.* • *Use quality verification methods and techniques to identify and correct defects in software, including static and dynamic code analysis, as well as unit and integration testing.* • *Understand and apply black box testing techniques to ensure that the software meets its functional and performance specifications, including constrained testing and non-replaceable testing.* • *Analyze software code and utilize white box testing techniques and state transition testing to identify and correct potential errors and risks.* • *Utilize various testing tools such as xUnit, NUnit, JUnit, PHPUnit, TestNG, etc., to automate and execute software testing.*

Prerequisites:	*Knowledge of programming fundamentals and software development, as well as understanding of database basics.*

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 with exercises and 30% laboratory work.*
Literature	
Basic Literature:	1. *Software Testing Foundations. Second Edition, Andreas Spillner, Tilo Linz, and Hans Schaefer. Rocky Nook, Inc. 2007. ISBN 9781 9339 5208 6.* 2. *Software Testing and Quality Assurance: Theory and Practice , Kshirasagar Naik & Priyadarshi Tripathy 2008. ISBN 978-0-471-78911-6*
Additional Literature:	1. *SOFTWARE TESTING Foundation Guide. Second Edition. Brian Hambling (Editor)*
Designed learning plan	
Week:	**Lectures and exercises to be held**
Week one:	*The fundamentals of software testing and basic concepts of Quality Assurance*
Week two:	*Software testing cycle and the quality assurance process*
Week three:	*Unit testing and unit quality verification*
Week four:	*Integration testing and assurance of integration quality*
Week five:	*System testing and assessment of system performance.*

Week six:	*Acceptance testing and quality control of software acceptance*
Week seven:	*Test 1*
Week eight:	*Testing software systems after addition of new modules and quality control of system updates.*
Week nine:	*Testing and static code analysis, and reviewing the quality of the code.*
Week ten:	*Black box testing techniques and their use to ensure high quality..*
Week eleven:	*White box testing techniques and their use to uncover potential defects.*
Week twelve:	*State transition testing and ensuring the quality of state changes in software.*
Week thirteen:	*Stress and overload testing and evaluating the quality of system resilience under challenging conditions*
Week fourteen:	*Test 2*
Week fifteen:	*Summary of the Topic: Reviewing important concepts and applying them in practice in the field of software testing and software quality assurance.*
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.	