SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babes-Bolyai University**
1.2 Faculty	**Faculty of Mathematics and Computer Science**
1.3 Department	**Department of Mathematics**
1.4 Field of study	**Mathematics**
1.5 Study cycle	**Bachelor**
1.6 Study programme / Qualification	**Mathematics and Computer Science (in Romanian)**

2. Information regarding the discipline

2.1 Name of the discipline	**Test Design Techniques**						
2.2 Course coordinator	**Lecturer PhD Camelia Chisăliţă-Creţu**						
2.3 Seminar coordinator	**Lecturer PhD Camelia Chisăliţă-Creţu**						
2.4. Year of study	**3**	2.5 Semester	**6**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**Optional**
2.8 Discipline Code	**MLE5110**						

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	4	Of which: 3.2 course	2	3.3 seminar/laboratory	1 lab + 1 project
3.4 Total hours in the curriculum	48	Of which: 3.5 course	24	3.6 seminar/laboratory	24

Time allotment:	Hours
Learning using manual, course support, bibliography, course notes	15
Additional documentation (in libraries, on electronic platforms, field documentation)	20
Preparation for seminars/labs, homework, papers, portfolios and essays	15
Tutorship	5
Evaluations	22
Other activities:	-

3.7 Total individual study hours	77
3.8 Total hours per semester	125
3.9 Number of ECTS credits	5

4. Prerequisites (if necessary)

4.1. curriculum	• OOP, Programming Fundamentals, Advanced Programming Methods
4.2. competencies	• Good programming skills in at least one of the programming languages Java, C#

5. Conditions (if necessary)

5.1. for the course	• Course hall with projector
5.2. for the seminar /lab activities	• Laboratory: computers and use of a programming language environment

6. Specific competencies acquired

Professional competencies	• **C2.1** Identify adequate software systems development methodologies • **C1.2** Identify and explain specific test design techniques that correspond to a testing level. • **C1.3** Source code and goal oriented test elaboration in a well-known programming language. • **C4.3** Identify models and methods adequate to real life problem solving.
Transversal competencies	• **CT1** Apply rules to organized and efficient work, responsibilities of didactical and • scientific activities and creative capitalization of own potential, while respecting principles and rules for professional ethics. • **CT3** Use efficient methods and techniques for learning, knowledge gaining, and research and develop capabilities for capitalization of knowledge, accommodation to society requirements and communication in English.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• Enhance the students understanding of testing and test design techniques. • Provide the students with an environment in which they can explore the usage and usefulness of software testing and test design concepts in various business scenarios. • Induce a realistic and industry driven view of software testing concepts and their inherent benefits.
7.2 Specific objective of the discipline	• Give students the ability to explore various test design techniques applied to different levels of testing. • Improve the students' abilities to tackle on goal driven testing. • Enhance the students understanding of test design techniques value in business. • Students will be able to use various tools for the testing process (i.e., test management, test running, test reporting and bug reporting). • Students will be able to design test cases according to an established testing goal and using specific test design technique in order to investigate the software.

8. Content

8.1 Course	Teaching methods	Remarks
1. Software Testing. Test Design Techniques 1.1. Software Testing. Goals. Scope 1.2. Test Design Technique. Attributes 1.3. Taxonomy of Test Design Techniques	• Interactive exposure • Explanation. Conversation • Didactical demonstration	
2. Coverage-based Techniques I 2.1. Focus. Objectives 2.2. Tours	• Interactive exposure • Explanation. Conversation • Didactical demonstration	
3. Coverage-based Techniques II 3.1. Specification-based Testing; 3.2. Requirements-based Testing;	• Interactive exposure • Explanation. Conversation • Didactical demonstration	
4. Tester-based Techniques I 4.1. Focus. Objectives 4.2. User Testing. Alpha Testing. Beta Testing	• Interactive exposure • Explanation. Conversation • Didactical demonstration	
5. Tester-based Techniques II 5.1. Bug Bashes. Paired Testing.	• Interactive exposure • Explanation. Conversation • Didactical demonstration	

6. Activity-based Techniques 6.1. Focus. Objectives 6.2. Guerilla Testing. All-pairs Testing 6.3. Use Cases Testing. Scenario Testing	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
7. Evaluation-based Techniques I 7.1. Focus. Objectives 7.2. Function Equivalence Testing. Self-verifying data	• Interactive exposure • Explanation. Conversation • Didactical demonstration	
8. Evaluation-based Techniques II 8.1. Comparison with saved Results; 8.2. Diagnostics-based Testing;	• Interactive exposure • Explanation. Conversation • Didactical demonstration	
9. Desired result-based Techniques I 9.1. Focus. Objectives 9.2. User Acceptance Testing	• Interactive exposure • Explanation. Conversation • Didactical demonstration	
10. Desired result-based Techniques II 10.1.Confirmation Testing; 10.2.Certification Testing;	• Interactive exposure • Explanation. Conversation • Didactical demonstration	
11. Risk-based Techniques 11.1.Focus. Objectives 11.2.Quicktests. History-based Testing. Usability Testing	• Interactive exposure • Conversation	
12. Test Design Techniques Analysis	• Interactive exposure • Conversation	

Bibliography

[Pres10] R. S. Pressman, Software engineering: a practinioner's approach, seventh edition, Higher Education, 2010.
[Crs09] L. Crispin, J. Grecory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009.
[You08] M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008.
[Nai08] K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008.
[Kat08] J. P. Katoen, Principles of Model Checking, MIT Press, May 2008.
[Pat05] R. Patton, Software Testing, Sams Publishing, 2005.
[Mye04] Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004.
[Brn02] I. Burnstein, Practical Software Testing, Springer, 2002.
[Kaner99] C. Kaner, J. Falk, H.Q. Nguyen, Testing Computer Software, Wiley, 1999.
[Perry97] W.E.Perry, R.W. Rice, Surviving theTop Ten Challenges of Software Testing – A People Oriented Approach, Dorset House Publishing, 1997.
[Kaner02] C. Kaner, J. Bach, B. Pettichord, Lesson Learned in Software Testing, Wiley, 2002.
[Page08] A. Page, K. Johnston, B. Rollison, Microsoft, How We Test Software at Microsoft, 2008.
[Whitt2012] J. Whittaker, J. Arbon J. Carollo, How Google Tests Software, Google, Pearson Education, 2012.

8.2 Seminar / laboratory	Teaching methods	Remarks
1. Laboratory 1 Testing tools and platforms. Testing Project Setup.	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
2. Laboratory 2 Coverage-based Test Design Techniques	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
3. Laboratory 3 Tester-based Test Design Techniques	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	

4. Laboratory 4 Activity-based Test Design Techniques	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
5. Laboratory 5 Evaluation-based Test Design Techniques	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
6. Laboratory 6 Project turn-in	Evaluation	
References: See references from Lectures.		

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- Students will know how to apply test design techniques for a software product, in a similar way they are used in industry.
- Students will be able to understand the differences between the goals and scope of the various test techniques applied to a software system.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Students will turn in during the examination session a report on a previously stated test design technique topic. The corresponding grade is denoted by **R.**	Oral Examination	50%
10.5 Seminar/laboratory activities	Each lab activity will be graded. The arithmetic average of the grades is denoted by **L**.	Laboratory Activity	50%
Remark: • Laboratory assignments will pe achieved in groups of 2-3 students.			
10.6 Minimum performance standards			
• Students will be able to apply test design techniques according to established goals for a software system. • Students will be able to unstandand the differences between software testing goal, scope, and test design technique concepts. • The final grade (M) is computed as follows: M = 50%R+50%L. • At least M >= 5.00 is favourable to pass this course exam.			

Date	Signature of course coordinator	Signature of seminar coordinator
20.04.2018	**Lect. PhD. Camelia Chisăliţă-Creţu,**	**Lect. PhD. Camelia Chisăliţă-Creţu,**

Date of approval — Signature of the head of department

Prof. PhD. Anca Andreica