

Syllabus

1. Programme information

1.1. Institution	THE BUCHAREST UNIVERSITY OF ECONOMIC STUDIES
1.2. Faculty	Economic Cybernetics, Statistics and Informatics
1.3. Departments	Department of Economic Informatics and Cybernetics
1.4. Field of study	Economic Cybernetics, Statistics and Informatics
1.5. Cycle of studies	Licence
1.6. Education type	Full-time
1.7. Study programme	Economic informatics (in english)
1.8. Language of study	English
1.9. Academic year	2022-2023

2. Information on the discipline

2.1. Name	Software Quality and Testing								
2.2. Code	22.0233IF3.2-0006								
2.3. Year of study	3	2.4. Semester	2	2.5. Type of assessment	Exam	2.6. Status of the discipline	O	2.7. Number of ECTS credits	5
2.8. Leaders	C(C)	prof.univ.dr. BOJA Cătălin Emilian					catalin.boja@ie.ase.ro		
	S(S)	prof.univ.dr. BOJA Cătălin Emilian					catalin.boja@ie.ase.ro		
	S(S)	lect.univ.dr. DOINEA-ZURINI V MĂDĂLINA					madalina.zurini@csie.ase.ro		
	S(S)	Asociat drd. CAZACU A ANDREI-ROBERT					andrei.r.cazacu@gmail.com		
	S(S)	conf.univ.dr. ZAMFIROIU I IONUȚ-ALIN					alin.zamfiroiu@csie.ase.ro		

3. Estimated Total Time

3.1. Number of weeks	14.00
3.2. Number of hours per week	4.00 of which
	C(C) 2.00
	S(S) 2.00
3.3. Total hours from curriculum	56.00 of which
	C(C) 28.00
	S(S) 28.00
3.4. Total hours of study per semester (ECTS*25)	125.00
3.5. Total hours of individual study	69.00
<i>Distribution of time for individual study</i>	
Study by the textbook, lecture notes, bibliography and student's own notes	20.00
Additional documentation in the library, on specialized online platforms and in the field	18.00
Preparation of seminars, labs, assignments, portfolios and essays	25.00
Tutorials	6.00
Examinations	
Other activities	

4. Prerequisites

4.1. of curriculum	Windows Applications Programming Java Programming
4.2. of competences	Basic knowledge of Java programming

5. Conditions

for the C(C)	Lectures are held in rooms with internet access and multimedia teaching equipment.
for the S(S)	Seminars are held in rooms equipped with computers with Internet access. Computers will have appropriate software installed for mobile application development

6. Acquired specific competences

PREFESSIONAL	C2	Effective using of the computing and operating systems and Internet resources.
PREFESSIONAL	C4	Developing software components by using data structures, algorithms, techniques and modern programming languages.
PREFESSIONAL	C5	Development of software components using data structures, algorithms, techniques and modern programming languages.

7. Objectives of the discipline

7.1. General objective	Acquiring the necessary skills needed for mobile applications development
7.2. Specific objectives	<ul style="list-style-type: none">• Provide students with basic knowledge related to: mobile devices, mobile communications and mobile applications:• Knowledge and effective use application programming interface for Windows Phone

8. Contents

8.1. C(C)		Teaching/Work methods	Recommendations for students
1	Introduction. Concept of: quality, verification and validation, software testing, testability, software metrics	Idem	
2	Design patterns (Creational, Structural Behavioral)	Idem	
3	Code versioning	Idem	
4	Software testing (test planning, methods and techniques: black box, white box, unit testing, integration testing)	Idem	
5	Testing automation	Idem	
6	Debugging, asserts	Idem	
7	Refactoring, code writing guidelines	Oral presentation + PowerPoint presentations + students interaction	

Bibliography

- Madalina Zurini, Alin Zamfiroiu, Calitate si testare software. Studii de caz, ASE, Bucuresti, 2017, România
- Andy Hunt, Dave Thomas, Pragmatic Unit Testing in Java with JUnit, The Pragmatic Programmers, 2004, Statele Unite ale Americii
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995, https://books.google.ro/books?id=6oHuKQe3TjQC&printsec=frontcover&source=gbs_ge_summary_r&hl=ro#v=onepage&q&f=false, Statele Unite ale Americii
- Robert C. Martin, Clean Code, A Handbook of Agile Software Craftsmanship, Prentice Hall, 2009, Statele Unite ale Americii
- Scott Chacon, Bean Strau, Pro Git, 2nd edition, Apress, 2014, <http://git-scm.com/book/en/v2>, Statele Unite ale Americii
- S. McConnell, Code complete, 2nd ed., Microsoft Press, 2004, Statele Unite ale Americii
- P. Pocatilu, Costurile testării software, ASE, 2004, România
- I. Ivan, C. Boja, S. Capisizu, M. Popa, Managementul calitatii aplicatiilor informatice, ASE, 2006, România
- I. Sommerville, Software Engineering. 9th ed., Addison-Wesley, 2011, Statele Unite ale Americii
- S. Pressman, Software Engineering: A Practitioner's Approach. 7th ed., McGraw-Hill, 2009, Statele Unite ale Americii

8.2. S(S)		Teaching/Work methods	Recommendations for students
1	Introduction. Metrics calculation, software tools, code analysis.	Computer applications	
2	Apply design patterns in practice	Computer applications	
3	Code versioning (SVN, CVS, GIT)	Computer applications	
4	Code Static Analysis (code review)	Computer applications	
5	Unit testing, class testing using JUnit, NUnit and Visual Studio	Computer applications	
6	Automated testing (standalone and Web applications)	Computer applications	

Bibliography

- I. Ivan, C. Boja, S. Capisizu, M. Popa, Managementul calitatii aplicatiilor informatice, ASE, 2006, România
- I. Sommerville, Software Engineering. 9th ed., Addison-Wesley, 2011, Statele Unite ale Americii
- S. Pressman, Software Engineering: A Practitioner's Approach. 7th ed., McGraw-Hill, 2009, Statele Unite ale Americii
- P. Pocatilu, Costurile testării software, ASE, 2004, România
- S. McConnell, Code complete, 2nd ed., Microsoft Press, 2004, România
- Robert C. Martin, Clean Code, A Handbook of Agile Software Craftsmanship, Prentice Hall, 2009, Statele Unite ale Americii
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995, https://books.google.ro/books?id=6oHuKQe3TjQC&printsec=frontcover&source=gbs_ge_summary_r&hl=ro#v=onepage&q&f=false, Statele Unite ale Americii
- Andy Hunt, Dave Thomas, Pragmatic Unit Testing in Java with JUnit, The Pragmatic Programmers, 2004, Statele Unite ale Americii
- Madalina Zurini, Alin Zamfiroiu, Calitate si testare software. Studii de caz, ASE, Bucuresti, 2017, România

9. Corroboration of the contents of the discipline with the expectations of the representatives of the epistemic community, of the professional associations and representative employers in the field associated with the programme

- Mobile technologies are an important part in ICT because of the advantages they offer
- Support for the practical implementation of efficient use of mobile applications and mobile devices

10. Assessment

Type of activity	Assessment criteria	Assessment methods	Percentage in the final grade
10.1. C(C)	Evaluation of aquired knowledge and concepts	Quiz test/Computer test in the last 2 weeks of the semester	10.00
10.2. S(S)	Applying acquired knowledge.	Homework/Project evaluation. Assessing progress is made during the semester. Each student presents the personal contribution during seminar activities by submitting assignments or projects Taking quizz tests during semester during seminars and courses Active participation during course and seminar	10.00
10.3. S(S)	Practical Evaluation of aquired knowldge	Computer tests. Implementing a solution based on given requirements	30.00
10.4. Final assessment	Evaluation of aquired knowledge and concepts	Quiz or computer exam (implementation of a software solution based on requirements)	50.00
10.5. Modality of grading	Whole notes 1-10		

<p>10.6. Minimum standard of performance</p>	<p>Taking the exam in normal session is conditioned by obtaining at least 2 points out of the 5 that can be accumulated during the semester.</p> <p>Active participation during the course and seminar is monitored in order to grade the semester activity</p> <p>Getting at least 5, out of a maximum of 10 points, at the exam.</p> <p>Implementing software solutions using the Java language, Clean Code principles, Design Patterns and testing the source code through unit tests in JUnit. The solutions must not contain compilation errors. All solutions must be implemented individually. Solutions that exceed a degree of similarity of 30% will be canceled.</p>
--	---

Date of listing,
02/29/2024

Signature of the discipline leaders,

Date of approval in the
department

Signature of the Department Director,