



DISCIPLINE ACADEMIC SHEET
ACADEMIC YEAR 2025 – 2026

1. PROGRAMME DATA

1.1. Higher Education Institution	UNIVERSITY OF CRAIOVA
1.2. Faculty	Automation, Computers and Electronics
1.3. Department	Computers and Information Technology
1.4. Field of Study	Computers and Information Technology
1.5. Study level	L(License / Undergraduate)
1.6. Organizational form	IF
1.7. Study Program (name/code) /Qualification	Computers / L2060101010

2. DISCIPLINE DATA

2.1 Discipline Name				Testing of Computer Systems			
2.2 Course Activities Holder				Popescu Horatiu Samir			
2.3. Practical Activities Holder				Popescu Horatiu Samir			
2.4. Study Year	4	2.5. Semester	2	2.6. Evaluation type	E	2.7. Discipline regime	DOP

3. ESTIMATED TOTAL TIME (hours per semester of teaching activities)

3.1. Number of hours per week	4	in which: 3.2 course	3	3.3. sem./lab./pr.	0/0/1
3.4. Total hours of curriculum	40	in which: 3.5 course	30	3.6. sem./lab./pr.	0/0/10
3.7. Time distribution - hours/week					10
Study after manual, course support, bibliography and notes					15
Additional documentation in library, on specialized electronic platforms and field					10
Training seminars / labs, homework, portfolios and essays					15
Tutorial					-
Examinations					10
Other activities.....					10
Total hours per individual activities					60
3.8. Total hours per semester					100
3.9. Number of credits					4

4. PRECONDITIONS (where appropriate)

4.1. of curriculum	• Computer Programming, Object Oriented Programming, Visual Programming Environments
4.2. of competence	• Computer Programming, Object Oriented Programming

5. CONDITIONS (where appropriate)

5.1. of the course	•
5.2. of seminar/ laboratory/project	•

6. COURSE OBJECTIVES – expected learning outcomes that are supported by completing and passing the course

Knowledge	Student/Graduate: 1. Understands the fundamental principles of software and hardware testing in computer systems. 2. Knows various testing methodologies (e.g., unit testing, integration testing, system testing, acceptance testing). 3. Is familiar with tools and frameworks used for automated testing (e.g., JUnit, Selenium, TestNG). 4. Understands the role of testing in the software development lifecycle and its impact on system reliability and performance.
Skills (Abilities)	Student/Graduate: 1. Can design and implement test cases for different types of computer systems. 2. Is able to use automated testing tools to validate software functionality and performance. 3. Can analyze test results and identify system defects or performance bottlenecks. 4. Applies debugging techniques and uses logs and reports to improve system quality. 5. Can work collaboratively in a team to plan and execute testing strategies.
Responsibility and autonomy	Student/Graduate: 1. Demonstrates responsibility in ensuring the quality and reliability of computer systems through rigorous testing. 2. Can independently plan, execute, and document testing activities. 3. Shows initiative in identifying potential risks and proposing mitigation strategies. 4. Is aware of ethical considerations in testing, such as data privacy and system integrity.

7. CONTENT

7.1. Course	The method of implementation	Teaching methods	Allocated time (hours)
1. Principles of Software Testing	face to face	Lectures and presentations	2
2. Quality Characteristics System	face to face	Continuous assessment through practical assignments	6
3. Quality Characteristics System	face to face	Continuous assessment through practical assignments	2
4. Software Testing Techniques	face to face	Continuous assessment through practical assignments	2
5. Static Testing	face to face	Case studies and problem-based learning	4
6. Black-Box Testing	face to face	Case studies and problem-based learning	6
7. White-Box Testing	face to face	Case studies and problem-based learning	6
8. Test cases Management	face to face	Lectures and presentations	2
Bibliography:			
1. Ion Ivan, Panagiotis Siniros – Metrici Software, Inforec, ISBN 973-97435-0-X, Bucuresti 1997			
2. Kaner, Cem; Falk, Jack - Testing Computer Software, ISBN: 0-471-35846-0, New York 1999			

7.2. Seminar/laboratory/project	The method of	Teaching methods	Allocated
--	----------------------	-------------------------	------------------

	implementation		time (hours)
1. Static Testing	face to face	Case studies and problem-based learning	2
2. Black-Box Testing	face to face	Continuous assessment through practical assignments	3
3. White-Box Testing	face to face	Continuous assessment through practical assignments	3
4. Software Testing Techniques	face to face	Case studies and problem-based learning	2
Bibliography:			
1. Rex Black - Managing the testing Process - 2nd Edition, John Wiley Publishing, Inc., 2002			
2. Jeff Tian - Software Quality Engineering; Testing, Quality Assurance, and Quantifiable Improvement - 2005			

8. Correlating the content of the discipline with the expectations of representatives from the epistemic community, professional associations and representative employers in the field related to the program

The content of the discipline is correlated with the current requirements of the software industry, professional associations and academic community in the field of computer systems testing and software quality assurance. The topics covered, such as test planning, black-box and white-box testing, static testing, test case management, defect identification and reporting, support the development of competences required by employers for positions related to software testing, quality assurance, validation and verification of computer systems. The discipline also supports the acquisition of practical and methodological skills necessary for applying modern testing techniques and tools in accordance with industry practices and quality standards.

9. EVALUATION

Activity Type	9.1. Evaluation criteria	9.2. Evaluation methods	9.3. Final mark weight
9.4. Course	One problem from quality characteristics system	Written exam	70%
	One problem from software testing methods	Written exam	70%
9.5. Seminar/laboratory/project	Presentation	Offline assessment	30%
	Weekly homework	Offline assessment	30%
9.6. Minimum standard of performance			
The grade from the written exam must be at least 5. The final grade (FG) is calculated using the formula: $FG = 0.7 * EG + 0.3 * PrG$ where: - EG – the grade from written exam. - PrG – the grade obtained for the project			

Date of Completion	Signature of the Course Activities Holder,	Signature of the Practical Activities Holder,
23.08.2025		

Date of Approval in the Department	Signature of the Department Director,
25.08.2025	Ș.l.dr.ing. Nicolae ENESCU

Date of Approval in the Faculty	Signature of the Dean,
	Prof.dr.ing. Dorin ȘENDRESCU