SYLLABUS

1. Program Information

1.1 Higher education institution	Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Department of Automation
1.4 Field of study	Automation, Applied Informatics and Intelligent Systems
1.5 Cycle of studies	Bachelor
1.6 Study Programme/Qualification	Intelligent Automation Systems (dual, in English language)
1.7 Form of education	IF – full-time education
1.8 Course code	17.00

2. Course information

2.1 Course title	**Software Engineering and Databases**					
2.2 Course lecturer	*Assoc. Prof. Eng. PhD Teodora Sanislav – teodora.sanislav@aut.utcluj.ro*					
2.3 Seminar / Laboratory / Project Lecturer	*Ing. Raluca Andrei (Emerson)* *DR.Ing. Alin Burz (Emerson)*					
2.4 Year of study	2	2.5 Semester	1	2.6 Type of assessment		
2.7 Course status	Formative category *(DF, DS, DC)*					DF
	Optionality *(DOB, DOP, DFac)*					DOB

3.Total estimated time

3.1 Number of hours per week	4	of which:	HEI	Lecture	2	Seminar	0	Laboratory	0	Project	0
			CO		0		0		2		0
3.2 Number of hours per semester	56	of which:	HEI	Lecture	28	Seminar	0	Laboratory	0	Project	0
			CO		0		0		28		0

3.3 Distribution of time allocation (hours per semester) for:	HEI	CO
(a) Study based on textbook, course support, bibliography, and notes	12	0
(b) Additional documentation in library, specialized electronic platforms, and fieldwork	5	7
(c) Preparation of seminars/laboratories, assignments, papers, portfolios and essays	0	12
(d) Tutoring	2	3
(e) Examinations	3	0
(f) Other activities:	0	0
3.4 Total individual study hours (sum (3.3(a)... 3.3(f)))	22	22
3.5 Total hours per semester (3.2+3.4)	50	50
3.6 Number of credits per semester	2	2

(HEI = Higher Education Institution, CO = Company)

4. Prerequisites (where applicable)

4.1 Curriculum prerequisites	• Computer Programming and Algorithms Design, • Computer Architectures and Operating Systems
4.2 Competency prerequisites	• Basics of computer use and operating systems, • Basics of computer programming

5. Conditions (where applicable)

5.1. Course organization conditions	• Classroom equipped with computer, video projector, Internet connection, whiteboard/blackboard/smart board • Attendance is compulsory
5.2. Seminar / Laboratory / Project organization conditions	• Laboratory room equipped with computers, network, Internet connection, specialized software, video projector, whiteboard/blackboard/smart board • Attendance is compulsory

6. Specific Competencies Acquired

Professional Competencies	• PC06 Define technical requirements • PC07 Demonstrate disciplinary expertise • PC15 Manage research data • PC17 Operate open-source software • PC20 Record test data • PC23 Synthesize information • PC24 Think abstractly • PC25 Use technical drawing software • PC26 Use information technology tools
Transversal Competencies	• TC02 Think analitically • TC03 Demonstrate responsability

7. Learning outcomes

Knowledge:	• Graduates will be able to explain fundamental concepts from automation, embedded and intelligent systems, computer science, and information technology, and apply these concepts to analyze, design, and develop effective software solutions for real-world technical challenges.
Skills:	• Graduates will be able to effectively use programming languages, development environments, and domain-specific tools and technologies, such as algorithms, diagrams, models, and protocols, to solve well-defined problems in systems engineering.
Responsibility and autonomy:	• Graduates will demonstrate initiative and autonomy in continuously updating their professional, economic, and organizational knowledge, taking responsibility for their own learning and development.

8. Course Objectives

8.1 General objective of the course	• Analysis, design, development, testing, deployment and maintenance of data-driven software applications for industry, following structured software engineering principles
8.2 Specific objectives	• Understanding of Object-Oriented Programming principles • Design, implementation, and management of databases • Design, development, testing of software applications for industry • Integration of databases within software applications • Use of Software Development Life Cycle models and tools like Unified Modeling Language, Entity-Relationship modeling

9. Contents

9.1 Lectures	No. of hours	Teaching methods	Obs.
1.Introduction to software engineering: Software Development Life Cycle (SDLC) definition and models	2	Lectures, discussions with students, demos and exercises	
2.Introduction to C# programming language: Basic elements (data types, variables, operators, statements, methods)	2		
3.Introduction to object-oriented programming (OOP) paradigm: Classes (members, access modifiers, constructors, properties), Objects, Inheritance, Polymorphism, Abstraction, Interface	2		
4.Introduction to databases: Data models, Entity-Relationship (ER) modeling, Schema design and normalization, Database Management Systems (DBMSs)	2		

5.Implementation and exploitation of relational databases: Structured Query Language (SQL), Queries, Views, Stored Procedures, Transactions	2		
6.Requirements analysis: Requirements engineering, Use cases and user stories	2		
7.Design and architecture of software applications for industry: Software design principles, Application architecture, Design patterns	2		
8.Design and architecture of software applications for industry: Unified Modeling Language (UML) diagrams	2		
9.Implementation of software applications for industry: Application-database integration, Graphical User Interfaces (GUIs)	2		
10.Implementation of software applications for industry: Implementation of UML design diagrams, Version control	2		
11.Testing of software applications for industry: Unit testing, Integration testing, System testing, Acceptance testing, Debugging and refactoring	2		
12.Deployment and maintenance of software applications for industry	2		
13.Example of applying software engineering principles to develop a complex software application for industry	2		
14.Review	2		

Bibliography

1. Ian Sommerville, *Software Engineering*, 10th Edition, Pearson Education Limited, ISBN: 978-1-292-09613-1, 816 pages, 2016
2. Erich Gamma, et.al., *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st Edition, Addison-Wesley Professional, ISBN: 978-0-201-633-610, 416 pages, 1994
3. Martin Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd Edition, Addison-Wesley Professional, ISBN: 978-0-321-193-681, 208 pages, 2003
4. Mark J. Price, *C# 12 and .NET 8 – Modern Cross-Platform Development*, ISBN: 978-1837635870, 826 pages, 2023
5. Teodora Sanislav, Liviu Miclea, Honoriu Vălean, *Baze de date relaționale și nerelaționale*, U.T.PRESS, ISBN: 978-606-737-113-0, 102 pages, 2015

9.2 Seminar / laboratory / project	Hours HEI	Hours CO	Teaching methods	Obs.
1.Introduction to industrial automation system engineering lifecycle (ISA-88, ISA-95, V-Model in industry)		2	Presentation of examples, description of software programming environments, additional explanations, exercises, discussions	
2.Introduction to scripting and logic in Emerson tools		2		
3.Modular and object-based control strategy design with templates/modules		2		
4.Database fundamentals: Designing data models for control systems		2		
5.Industrial data management and historian systems using Emerson platforms		2		
6.Functional requirements gathering for process systems (P&ID, I/O Lists, ISA Specs)		2		
7.Design principles in control systems		2		
8.Industrial control system design and architecture using control strategy diagrams and plant models		2		
9.Integrating applications with databases		2		
10.Implementation of industrial automation systems: Control strategy realization and change management		2		

11. Software testing: Ensuring quality in control system applications		2		
12. Deploying and maintaining control system applications		2		
13. Software engineering principles in distributed controlled systems. Design & implement a scalable industrial control system		2		
14. Laboratory test		2		
Bibliography 1. Emerson PCSD Books Online				

10. Correlation of course content with the expectations of the epistemic community representatives, professional associations, and major employers in the field related to the program

The content of the course is strategically aligned with the expectations of the software engineering epistemic community. In addition, the curriculum is informed by the current needs and future projections of major employers in the software and technology sectors (the laboratory works are derived from actual Emerson-based control and automation solutions, offering students exposure to industry-grade implementations). This alignment ensures that students are equipped not only with a solid theoretical foundation but also with the practical skills and methodologies required in modern software development environments. Curriculum updates are guided by ongoing collaboration between academic and industry experts, which guarantees that the course remains current, competitive, and professionally valuable for graduates entering the global IT industry.

11. Evaluation

Activity Type	**Evaluation criteria**	**Evaluation methods**	**Weight in final grade**
11.1 Lecture	- Demonstrated ability to apply acquired knowledge effectively - Proven capacity to implement theoretical concepts in practical scenarios	Written exam	50%
11.2 Seminar/ Laboratory/Project	- Demonstrated ability to apply acquired knowledge effectively - Proven capacity to implement theoretical concepts in practical scenarios	Computer test	50%
11.3 Minimum Performance Standard			
Each component of the final grade must be passed with at least grade five.			

Date of completion:	Lecturers	Title First Name LAST NAME	Signature
15.09.2025	Course	Assoc. Prof. Eng. PhD Teodora SANISLAV	

Date of approval by the Department of Automation Council 24.11.2025	Director of the Department of Automation Prof.dr.ing. Honoriu VĂLEAN
Date of approval by the Faculty of Automation and Computer Science Council 28.11.2025	Dean Prof.dr.ing. Vlad MUREȘAN