

Syllabus

1. Data about the program of study

1.1 Institution	Technical University of Cluj-Napoca
1.2 Faculty	Automation and Computer Science
1.3 Departament	Automation
1.4 Field of study	Automation. Applied Informatics and Intelligent Systems
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Automation and Applied Informatics (English)
1.7 Form of education	Full time

2. Data about the subject

2.1 Subject name			Applied Indformatics			Discipline code		12.00	
2.2 Course responsible/lecturer			Prof.dr.ing. Honoriu VĂLEAN – Honoriu.Valean@aut.utcluj.ro						
2.3 Teachers in charge of applications			Conf.dr.ing. Dan GOȚA – Dan.Gota@aut.utcluj.ro						
			As.dr.ing. Claudiu DOMUTA – Claudiu.Domuta@aut.utcluj.ro						
			As.drd.ing. Alexandru CIOBOTARU – Alexandru.Ciobotaru@aut.utcluj.ro						
2.4 Year of study		1	2.5 Semester	2	2.6 Assessment (E/C/V)				E
2.7 Type of subject		DF – fundamental, DS – specialty, DC – complementary							DF
		DOB – compulsory, DOP – elective, DFA – optional							DOB

3. Estimated total time

3.1 Number of hours per week	4	of which:	3.2 Course	2	3.3 Seminar	0	3.3 Laboratory	2	3.3 Project	0
3.4 Total hours in the teaching plan	56	of which:	3.5 Course	28	3.6 Seminar	0	3.6 Laboratory	28	3.6 Project	0
3.7 Time distribution (hours per semester) for:										
(a) Manual, lecture material and notes, bibliography										4
(b) Supplementary study in the library, online and in the field										30
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										30
(d) Tutoring										0
(e) Exams and tests										5
(f) Other activities:										0
3.8 Total hours individual study hours (sum of (3.3(a))...3.3(f)))	69									
3.9 Total hours per semester (3.2+3.4)	125									
3.10 Number of credits points	5									

4. Pre-requisites (where appropriate)

4.1 Curriculum	Computer programming
4.2 Competence	Operating with fundamental concepts in computer science, information and communication technology

5. Requirements (where appropriate)

5.1. For the course	Amphiteater with projector and internet access
5.2. For the applications	Laboratory equipped with computer network - C ++ programming environment (Visual Studio), internet access, projector, blackboard

6. Specific competences

Professional competences	C2 Operating with basic concepts of computer science, information technology and communication C2.2 Well grounded usage of concepts from informatics and computer technology in solving well defined problems of system engineering and in applications requiring the use of hardware or software in industrial systems or information technology systems. C2.3 Solving common problems of systems engineering using the computer science and information technology concepts for the use of dedicated software tools and computer aided design (CAD) and for the adaptation and extension of these. C4 Design, implementation, testing, operation and maintenance of systems with generic and dedicated equipments, including computer networks for control engineering and applied informatics. C4.1 Defining the requirements of applicable standards and of the methods of implementation, testing, operation and maintenance for the equipments used in the applications of automatic control and applied informatics based on the operation and design principles C4.2 Defining the requirements of applicable standards and of the methods of implementation, testing, operation and maintenance for the equipments used in the applications of automatic control and applied informatics based on the operation and design principles.
Cross competences	

7. Expected learning outcomes

Knowledge	The student/graduate identifies and describes basic concepts, principles and methods in mathematics, physics, chemistry, technical drawing, economics and computer science; explains and interprets theoretical and experimental results in mathematics, physics, chemistry, economics, technical drawing and computer science; describes, identifies and summarizes fundamental concepts in automatic systems, embedded and intelligent systems, computer science and information technology and their application in concrete problems;
Skills	The student/graduate operates with basic concepts, principles and methods from mathematics, physics, chemistry, technical drawing, economics and computer science; describes physical-chemical and economic phenomena and processes; applies evaluation criteria and methods for the identification, modeling, experimentation, analysis and qualitative and quantitative assessment of phenomena and processes specific to the fundamental field using digital technologies; uses programming languages, environments and technologies and specific tools (algorithms, schemes, models, protocols, etc.) in solving well-defined problems in systems engineering; develops software solutions that integrate algorithm design, data management and workflow automation for engineering and administrative tasks; uses intermediate and high-level programming languages, including languages used for the implementation of intelligent systems; designs, implements and queries databases and uses fundamental algorithms; uses advanced technologies in the design and implementation of software applications;
Responsibilities and autonomy	The student/graduate applies the values of ethics and deontology of the engineering profession; practices logical reasoning, evaluation and self-evaluation in decision-making. Tracks personal progress in learning, seeks feedback and adopts strategies for continuous professional and personal development; plans, executes and manages tasks autonomously, from problem definition to solution, demonstrating initiative in adopting new tools, methods and technologies.

8. Course objectives

General objective	Development of the theoretical and practical skills necessary for the analysis, design, implementation and testing of average complexity applications in C ++ and C # languages, using the OOP paradigm
Specific objectives	Understanding the paradigm of object-oriented programming.

9. Contents

9.1 Lecture	No.hours	Teaching methods	Notes
The paradigm of object-oriented programming. OOP utility.	2	Teaching using laptop and projector,	
Difference between structure and class. Encapsulation. Levels of protection.	2		

Constructors, destructors. Dynamic allocation, copy constructors.	2	interactive course, debate	
Operators overloading. Functions. Friend functions.	2		
Inheritance. Simple, multiple, public, private inheritance.	2		
Polymorphism. Methods overloading. Virtual methods. Abstract methods.	2		
Template	2		
C# programming language.	2		
OOP in C#. Comparison with C++. Inheritance.	2		
Polymorphism in C#. Abstract classes. Interfaces.	2		
Generics.	2		
Graphic programming under Windows. Writing simple programs.	2		
Graphic programming in OOP context.	2		
The paradigm of object-oriented programming. OOP utility.	2		
Bibliography			
1. J. Liberty. C++ unleashed. Sams. Pub., ISBN 0672312395, 1999			
2. H. Valean, L. Miclea, M.Damian. Introducere in Visual C++. Ed. U.T. Pres, 2005, 290 pag, ISBN 973-662-154-5			
3. K.U. Subhash. Object Oriented Programming with C++. Pearson Education, ISBN 8131733297, 2010			
4. P.B. Kotur. Object Oriented Programming with C++. Sapna Book House, ISBN 978-81-280-1853-4, 2012			
5. Course notes available on Teams			
9.2 Applications (seminar/laboratory/project/practical work)	No.hours	Teaching methods	Notes
The paradigm of object-oriented programming. OOP utility.	2	Presentation of examples, demonstrations, discussions, practical applications / or online on Teams platform	Mandatory attendance
Difference between structure and class. Encapsulation. Levels of protection.	2		
Constructors, destructors. Dynamic allocation, copy constructors.	2		
Operators overloading. Functions. Friend functions.	2		
Inheritance. Simple, multiple, public, private inheritance.	2		
Polymorphism. Methods overloading. Virtual methods. Abstract methods.	2		
Template	2		
C# programming language.	2		
OOP in C#. Comparison with C++. Inheritance.	2		
Polymorphism in C#. Abstract classes. Interfaces.	2		
Generics.	2		
Graphic programming under Windows. Writing simple programs.	2		
Graphic programming in OOP context.	2		
Laboratory Assessment	2		
Bibliography			
1. J. Liberty. C++ unleashed. Sams. Pub., ISBN 0672312395, 1999			
2. H. Valean, L. Miclea, M.Damian. Introducere in Visual C++. Ed. U.T. Pres, 2005, 290 pag, ISBN 973-662-154-5			
3. K.U. Subhash. Object Oriented Programming with C++. Pearson Education, ISBN 8131733297, 2010			
4. P.B. Kotur. Object Oriented Programming with C++. Sapna Book House, ISBN 978-81-280-1853-4, 2012			
5. Course notes available on Teams			

10. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

The content of the discipline, together with the skills and abilities acquired, correspond to the expectations of the professional organizations of profile, the companies of profile, as well as of the national and international bodies of quality assurance (ARACIS). It also ensures the adoption of ethical standards appropriate to the engineering practice

11. Evaluation

Activity type	11.1 Assessment criteria	11.2 Assessment methods	11.3 Weight in the final grade
11.4 Course	Assessment of knowledge through a test based on the knowledge gained following participation in the course	Written exam / online exam using Teams	60%

11.5 seminar/ laboratory/ project/ practical work	Examination of the skills and knowledge acquired through the participation in the laboratory.	Practical assessment / online assesment using Teams	40%
11.6 Minimum performance standard Written exam rank > 5 and practical assessment rank > 5			

Date of filling in:		Title Firstname NAME	Signature
10.07.2026	Course	Prof.dr.ing. Honoriu VĂLEAN	
	Aplications	Conf.dr.ing. Dan GOȚA	
		SL.dr.ing. Claudiu DOMUȚA	
		As.drd.ing. Alexandru CIOBOTARU	

Date of approval by the Department Board	Head of Departament Prof.dr.ing. Honoriu VĂLEAN
Date of approval by the Faculty Council	Dean Prof.dr.ing. Vlad MUREȘAN