SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	22.

2. Data about the subject

2.1 Subject name	***Systems Theory***				
2.2 Course responsible/lecturer	Assoc. prof. dr. eng. Paula Raica – Paula.Raica@aut.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Conf.dr.ing. Paula Raica, Drd. Ing. Zoltan Nagy				
2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DD
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars		Laboratory	2	Project	
3.2 Number of hours per semester	56	of which:	Course	28	Seminars		Laboratory	28	Project	

3.3 Individual study:	
(a) Manual, lecture material and notes, bibliography	20
(b) Supplementary study in the library, online and in the field	1
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	20
(d) Tutoring	
(e) Exams and tests	3
(f) Other activities:	

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	44
3.5 Total hours per semester (3.2+3.4)	100
3.6 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1 Curriculum	Mathematical Analysis_II (Integral calculus and differential equations, Linear algebra
4.2 Competence	Differential equations, complex numbers, Laplace transform, linear algebra

5. Requirements (where appropriate)

5.1. For the course	N/A
5.2. For the applications	Reading and understanding of the lecture notes.

6. Specific competence

6.1 Professional competences	**C1 –** Operating with basic Mathematical, Engineering and Computer Science concepts (4 credits) **C1.1 –** Recognizing and describing concepts that are specific to the fields of calculability, complexity, programming paradigms, and modeling computational and communication systems **C1.2 –** Using specific theories and tools (algorithms, schemes, models, protocols, etc.) for explaining the structure and the functioning of hardware, software and

	communication systems **C1.3 –** Building models for various components of computing systems **C1.4 –** Formal evaluation of the functional and non-functional characteristics of computing systems **C1.5 –** Providing a theoretical background for the characteristics of the designed systems
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	The general objective of the course is to introduce the fundamental principles of linear system modeling, analysis and feedback control and to evaluate feedback control systems with desired behavior.
7.2 Specific objectives	The specific objectives are to acquire the knowledge and techniques related to: - mathematical system modeling (differential equations, input-output representation as transfer functions, block diagrams) for simple applications - linear system analysis (assessment of stability and performance properties of linear systems) in time and frequency domains - design of feedback controllers such as PID, lead and lag compensators for linear systems using s-domain techniques - linear sampled-data system representation and analysis

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction to systems theory and control engineering. Intoduction to system modeling. Linear approximation.	2	Lecture, visual presentations, demonstrations	
Input/output models. System response. State-space models.	2		
Conversion between transfer function and state space. Block diagrams.	2		
Linear system analysis. 1st and 2nd order systems. Steady-state error.	2		
Higher order systems. Dominant poles. Stability of linear continuous systems.	2		
System analysis using root locus.	2		
Frequency response. Bode diagrams.	2		
Controller design. Lead-lag compensation.	2		
System analysis. Applications. Midterm exam.	2		
PID – the basic technique for feedback control.	2		
Controlability. Observability. State feedback.	2		
Sampled-data systems.	2		
Digital control systems	2		
Controller design – aplications. Sampled-data systems – applications.	2		

Bibliography

1. R. C. Dorf, R. Bishop, “Modern Control Systems”, Addison-Wesley, 2004;
2. K. Ogata , “Modern Control Engineering”, Prentice Hall, 1990.
3. K. Dutton, S. Thompson, B. Barraclough, “The Art of Control Engineering”, Addison-Wesley, 1997
4. William S. Levine (editor), “The Control Handbook”, CRC Press and IEEE Press, 1996
5. Lecture notes available on the course webpage: http://cursuri.aut.utcluj.ro

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Introduction to Matlab. Simulation of dynamical systems	4	Class discussion, Supervised exercise solving using Matlab Miniprojects – individul student reports	
Linear approximation of differential equations. Transfer functions. System response.	4		
Block diagram models. 1st and 2nd order system analysis. Steady-state error	4		
System stability. Root locus	4		
Frequency response. Bode diagrams	4		
Lead-lag compensation. PID controllers	4		

State feedback. Sampled-data systems.	4		

Bibliography

1. Paula Raica, “Control Engineering. Exercises”, Editura Mediamira, 2001
2. Lecture notes available on the course webpage: http://cursuri.aut.utcluj.ro

Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

The course content combines theoretical knowledge with applications and focuses on the formulation and solution of specific problems that may occur in various engineering fields. Application of the control theory concepts are specific to most of the engineering disciplines. The course level is introductory and the intent is to motivate and prepare students for further study in related areas and to conduct projects in real-life applications.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Ability to solve exercises related to linear system modeling and analysis Ability to solve exercises related to system design and analysis of sampled-data systems	Midterm exam – writen examination Final exam - writen examination	40% 60%
Seminar			
Laboratory	Answer simple questions from the topic of the lab applications	Lab tests (optional)	30% (optional, but may contribute to a higher grade)
Project			

Minimum standard of performance:
Solution of simple exercises applying the knowledge and techniques presented in the course.
40% Midterm grade + 60%Final grade + 30%Lab grade > 5

Course responsible
Conf.dr.ing. Paula Raica

Head of department
Prof.dr.eng. Rodica Potolea

SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	23.

2. Data about the subject

2.1 Subject name	***Computer Architecture***				
2.2 Course responsible/lecturer	S.l.dr.ing. Mihai Negru – Mihai.Negru@cs.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Conf.dr. ing. Florin Oniga, S.l.dr.ing. Mihai Negru, { Florin.Oniga, Mihai.Negru }@cs.utcluj.ro				
2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DD
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars		Laboratory	2	Project	
3.2 Number of hours per semester	56	of which:	Course	28	Seminars		Laboratory	28	Project	
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										28
(b) Supplementary study in the library, online and in the field										14
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										23
(d) Tutoring										0
(e) Exams and tests										4
(f) Other activities:										0

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	69
3.5 Total hours per semester (3.2+3.4)	125
3.6 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	Logic design >= 5 Digital system design >= 5
4.3 Competence	Ability to design digital circuits and to implement them in VHDL

5. Requirements (where appropriate)

5.1. For the course	blackboard, video projector, laptop
5.2. For the applications	desktop/laptop computer, Xilinx ISE / VIVADO, FPGA development boards

6. Specific competence

6.1 Professional competences	**C2 –** Designing hardware, software and communication components (5 credits) **C2.1 –** Describing the structure and functioning of computational, communication and software components and systems **C2.2 –** Explaining the role, interaction and functioning of hardware, software and communication components **C2.3 –** Building the hardware and software components of some computing systems using algorithms, design methods, protocols, languages, data structures,

	and technologies **C2.4 –** Evaluating the functional and non-functional characteristics of the computing systems using specific metrics **C2.5 –** Implementing hardware, software and communication systems
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	Knowing and understanding the concepts of organization and functioning for central processing units, memories, input/output, and using these concepts for design.
7.2 Specific objectives	• Applying methods for representation and design at system level for digital circuits • Instruction Set Architecture (ISA) specification • Writing simple programs in assembly languages and machine code • Specification, design, implementation, and testing of Central Processing Units (CPU) – micro architecture – data path – command units • Understanding memory organization and I/O operations • Understanding modern trends in computer architectures

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction	2	Oral presentation backed up by multimedia equipment, interactive communication, blackboard problem solving	
High-Level Synthesis	2		
Instruction Set Architecture (ISA)	2		
CPU Design - Single Cycle CPU	2		
Computer Arithmetic and Simple Arithmetic Logic Units	2		
CPU Design - Multi Cycle CPU Data path	2		
CPU Design - Multi Cycle CPU Control	2		
CPU Design – Pipelined CPU	2		
Advanced Pipelining – Static and Dynamic Scheduling of the Execution	2		
Branch Prediction	2		
Superscalar Architectures	2		
Memory	2		
I/O and Interconnection Structures	2		
Problem solving	2		

Bibliography

1. D. A. Patterson, J. L. Hennessy, "Computer Organization and Design: The Hardware/Software Interface",5th edition, ed. Morgan–Kaufmann, 2013.
2. D. A. Patterson and J. L. Hennessy, "Computer Organization and Design: A Quantitative Approach",5th edition, ed. Morgan-Kaufmann, 2011.
3. Vincent P. Heuring, et al., "Computer Systems Design and Architecture", Addison-Wesley, USA, 1997.
4. A. Tanenbaum, "Structured Computer Organization", Prentice Hall, USA, 1999.
5. MIPS32 Architecture for Programmers, Volume I: "Introduction to the MIPS 32™ Architecture".
6. MIPS32 Architecture for Programmers, Volume II: "The MIPS 32™ Instruction Set".

Online bibliography

M. Negru, F. Oniga, S. Nedevschi, Lecture slides http://users.utcluj.ro/~negrum

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Introduction in the Xilinx ISE environment and the FPGA development board	2	Blackboard quick overview of key issues, exercises, experimenting with FPGA development boards with specialized	
Design and Implementation of Combinational CPU Components	2		
Design and Implementation of Sequential CPU Components	2		
Design of a Single Cycle CPU 1 (MIPS)	2		
Design of a Single Cycle CPU 2 (MIPS)	2		

Design of a Single Cycle CPU 3 (MIPS)	2	IDEs for circuit design and implementation (Xilinx ISE)	
Design of a Single Cycle CPU 4 (MIPS)	2		
Midterm practical evaluation on the FPGA board	2		
Pipelined CPU Design	2		
Pipelined CPU Design	2		
Pipelined CPU Design	2		
Pipelined CPU interfacing	2		
Practical evaluation of the pipelined CPU on the FPGA board	2		
Final Tests and Evaluation	2		

Bibliography

Online bibliography

M. Negru, F. Oniga, S. Nedevschi, Laboratory guide http://users.utcluj.ro/~negrum

*Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

Computer Architecture is one of the fundamental subjects of the Computer Science and Information Technology field. It combines fundamental and practical aspects used for digital circuits design and implementation. The content of this subject is harmonized with the specific curricula of other national and international universities, and is evaluated by the Romanian government agencies (CNEAA and ARACIS). The practical aspects involve getting familiar with and using development products and tools provided by companies from Romania, Europe, and USA (ex. Xilinx, Digilent).

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Testing the theoretical knowledge, the ability of problem solving, presence and activity	Written exam	50%
Laboratory	Practical ability to solve and implement specific problems related to processor design, presence and activity	Lab exam, periodical assessment of results	50%
Project			

Minimum standard of performance:
Knowing the fundamental theory of the subject, the ability to design and implement a processor with a reduced set of instructions.
Grade calculus: 50% lab + 50% final exam
Conditions for participating in the final exam: Lab ≥ 5
Conditions for promotion: Final exam ≥ 5

Course responsible
S.l.dr.ing. Mihai Negru

Head of department
Prof.dr.eng. Rodica Potolea

SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	24.

2. Data about the subject

2.1 Subject name	***Numerical methods***
2.2 Course responsible/lecturer	Prof. dr. Dumitru Mircea Ivan – mircea.ivan@math.utcluj.ro
2.3 Teachers in charge of seminars/ laboratory/ project	Prof. dr. Daniela ROSCA – daniela.rosca@math.utcluj.ro S.I. Mircia GURZAU – mircia@gurzau@math.utcluj.ro

2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DF
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars		Laboratory	2	Project	
3.2 Number of hours per semester	56	of which:	Course	28	Seminars		Laboratory	28	Project	

3.3 Individual study:	
(a) Manual, lecture material and notes, bibliography	15
(b) Supplementary study in the library, online and in the field	10
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	15
(d) Tutoring	0
(e) Exams and tests	4
(f) Other activities:	0

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	44
3.5 Total hours per semester (3.2+3.4)	100
3.6 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1 Curriculum	Basic knowledge of Differential and Integral Calculus
4.4 Competence	Competences in elementary Differential and Integral Calculus: derivatives, integrals, series.

5. Requirements (where appropriate)

5.1. For the course	Videoprojector
5.2. For the applications	Videoprojector

6. Specific competence

6.1 Professional competences	**C1 –** Operating with basic Mathematical, Engineering and Computer Science concepts (5 credits) **C1.1 -** Recognizing and describing specific concepts to calculability, complexity, programming paradigms and modeling of computing and communication systems **C1.2 -** Using specific theories and tools (algorithms, schemes, models, protocols, etc.) for explaining the structure and the functioning of hardware, software and

	communication systems **C1.3 -** Building models for various components of computing systems **C1.4 -** Formal evaluation of the functional and non-functional characteristics of computing systems **C1.5 -** Providing theoretical background for the characteristics of the designed systems
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	A presentation of the concepts, notions, methods and fundamental techniques used in differential calculus.
7.2 Specific objectives	Use of the differential calculus in order to solve problems in engineering.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Elements of Error Theory. Floating Point Arithmetic. Absolute and Relative Errors.	2	Explanation Demonstration Collaboration Interactive activities	
Numerical Methods in Linear Algebra. Special Types of Matrices. Norms of Vectors and Matrices. Eigenvalues and Eigenvectors. Error Estimation. Matrix Equations. Pivoting Elimination. Improved Solutions of Matrix Equations. Partitioning Methods for Matrix Inversion. LU Factorization. Doolittle's Factorization. Choleski's Factorization Method. Iterative Techniques for Solving Linear Systems. Jacobi Iterative Method. Gauss-Seidel Iterative Method. Relaxation Methods. Characteristic Polynomial: Leverrier Method. Characteristic Polynomial: Fadeev-Frame Method.	4		
Solutions of Nonlinear Equations. Method of Successive Approximation. The Bisection Method. The Newton-Raphson Method. The Secant Method. False Position Method. The Chebyshev Method. Numerical Solutions of Nonlinear Systems of Equations. Newton's Method for Systems of Nonlinear Equations. Steepest Descent Method.	4		
Elements of Interpolation Theory. Lagrange Interpolation. Divided Difference. Mean Value Properties in Lagrange Interpolation. Approximation by Interpolation. Hermite Interpolating Polynomial. Finite Differences. Interpolation of Multivariable Functions. Scattered Data Interpolation. Shepard's Method. Splines. B-splines.	6		
Elements of Numerical Integration. Richardson's Extrapolation. Numerical Quadrature. Error Bounds in the Quadrature Methods. Trapezoidal Rule. Richardson's Deferred Approach to the Limit. Romberg Integration. Newton-Cotes Formulas. Simpson's Rule. Gaussian Quadrature.	4		
Elements of Approximation Theory. Discrete Least Squares Approximation. Orthogonal Polynomials and Least Squares Approximation. Rational Function Approximation. Padé Approximation. Trigonometric Polynomial Approximation. Fast Fourier Transform. Bernstein Polynomial. Bézier Curves. *METAFONT.*	4		
Integration of Ordinary/Partial Differential Equations. The Euler Method. The Taylor Series Method. The Runge-Kutta Method. The Runge-Kutta Method for Systems of Equations. **Integration of Partial Differential Equations.** Parabolic Partial-Differential Equations. Hyperbolic Partial Differential Equations. Elliptic Partial Differential Equations.	4		
Bibliography 1. Dumitru Mircea Ivan. Calculus. Editura Mediamira, Cluj-Napoca, 2002. ISBN 973-9358-88-8. 2. Mircea Ivan and Kálmán Pusztai. Numerical Methods with Mathematica. Mediamira, Cluj-Napoca, 2003. ISBN			

973-9357-41-5.

3. Ioan-Adrian Viorel, Dumitru Mircea Ivan, and Loránd Szabó. Metode numerice cu aplicaţii în ingineria electrică. Editura Universităţii din Oradea,
4. Mircea Ivan and Kálmán Pusztai. Mathematics by Computer. Comprex Publishing House, Cluj-Napoca, 1992.

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
The applications follow the topics of the courses.	28	Explanation Demonstration Collaboration Interactive activities	

Bibliography

1. Mircea Ivan and Kálmán Pusztai. Numerical Methods with Mathematica. Mediamira, Cluj-Napoca, 2003. ISBN 973-9357-41-5.
2. Ioan-Adrian Viorel, Dumitru Mircea Ivan, and Loránd Szabó. Metode numerice cu aplicaţii în ingineria electrică. Editura Universităţii din Oradea,

Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

Collaboration with engineers in order to identify and solve problems raised by the market.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Abilities of understanding and using creatively the concepts and proofs	Written examination	30%
Seminar			
Laboratory	Abilities of solving problems and applying algorithms	Written examination	70%
Project			
Minimum standard of performance: Ability to present coherently a theoretical subject and to solve problems with practical content.			

Course responsible
Prof.dr. Dumitru Mircea IVAN

Head of department
Prof.dr.eng. Rodica Potolea

SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	25.

2. Data about the subject

2.1 Subject name	***Fundamental Programming Techniques***				
2.2 Course responsible/lecturer	Prof. dr. eng. Ioan Salomie - Ioan.Salomie@cs.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Assoc.prof. dr. eng. Tudor Cioară,, Sl. dr. eng. Cristina.Pop, S.l. Dr. Marcel Antal, As.drd. Claudia Pop, As. Drd. Dorin Moldovan				
2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DF
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars		Laboratory	2	Project
3.2 Number of hours per semester	56	of which:	Course	28	Seminars		Laboratory	28	Project

3.3 Individual study:	
(a) Manual, lecture material and notes, bibliography	10
(b) Supplementary study in the library, online and in the field	16
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	14
(d) Tutoring	
(e) Exams and tests	4
(f) Other activities:	

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	44
3.5 Total hours per semester (3.2+3.4)	100
3.6 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1 Curriculum	Fundamentals of Object Oriented Programming
4.5 Competence	Knowledge of Object Oriented Programming

5. Requirements (where appropriate)

5.1. For the course	Blackboard, projector, computer, internet
5.2. For the applications	Computers, specific software, internet

6. Specific competence

6.1 Professional competences	**C4 -** Improving the performances of the hardware, software and communication systems **C4.1** - Identifying and describing the defining elements of the performances of the hardware, software and communication systems **C4.2** - Explaining the interaction of the factors that determine the performances of the hardware, software and communication systems **C4.3 -** Applying the fundamental methods and principles for increasing the performances of the hardware, software and communication systems

	C4.4 - Choosing the criteria and evaluation methods of the performances of the hardware, software and communication systems **C4.5** - Developing professional solutions for hardware, software and communication systems based on performance optimization
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	Knowledge and using of object-oriented programming techniques for the development of professional software applications
7.2 Specific objectives	-to use programming techniques for designing of classes and interfaces, including contracts and invariants; -to use programming techniques for code reuse by inheritance and polymorphism -to use generic programming techniques for collection processing -to use programming techniques for reflection and event based -to use programming techniques for concurrent and multi-threading programming -to use object-oriented and functional programming in an integrated approach for the development of flexible and efficient programs -to use design patterns and frameworks -to use programming techniques for performance and software maintenance

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Programming techniques with classes and interfaces	2	-Using modern multimedia teaching methods and direct access to internet; -Challenging questions during lecturers -Students are invited to collaborate in research projects -Personal assistance hours the semester and before the exam	
Programming techniques using inheritance and polymorphism	2		
Programming techniques using contracts and invariants	2		
Generic programming techniques	2		
Reflection techniques	2		
Event-driven techniques	2		
Collection programming techniques	2		
Concurrent and multithreading techniques	2		
Flexibility and reuse through design patterns	2		
Main design patterns of type creational, structural and behavioral	2		
Flexibility and reuse through frameworks	2		
Lambda Expressions and Stream processing	2		
Multiparadigm (functional and OO) programming techniques	2		
Programming techniques for efficiency and performance	2		

Bibliography

1. Ioan Salomie - Tehnici Orientate Obiect, Editura Albastra, Microinformatica, 1995
2. Eric Gamma, Helm, Johnson, Vlissides - Design Patterns, Addison Wesley, 1995 (translated into Romanian by Teora Publ. as "Sabloane de Proiectare")
3. Joshua Bloch - Effective Java, 2/e Addison Wesley, 2008
4. Ioan Salomie, Note de Curs, http://www.coned.utcluj.ro/~salomie/TP

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Intro to lab resources and requirements	2	-Lab sessions with pre-defined exercises and assignments -Using modern multimedia teaching methods and direct access to internet;	
Assignment 1 - Programming with inheritance and polymorphism	4		
Assignment 2 - Programming with contracts (pre and post conditions) and invariants	4		
Assignment 3 Programming with multiple threads	4		
Assignment 4 – Programming with design patterns	4		

Assignment 5 – Programming with generics and Java Collection Framework	4	-Students are invited to collaborate in research projects -Personal assistance hours during the semester and before the exam	
Assignment 6 – Multi-paradigm programming	4		
Lab Evaluation	2		
Bibliography - Steve McConnell - Code Complete, 2/e, Microsoft Press, 2004 - http://docs.oracle.com/javase/tutorial/index.html - http://stackoverflow.com/			

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

Fundamental Programming Techniques is a subject of the domain "Computers and Information Technology". It teaches students to apply object-oriented programming techniques in designing and implementing of software applications. The content was developed based on the analysis of similar disciplines from other universities as well as based on the requirements of the IT employees. The content was also evaluated by Romanian governmental agencies CNEAA and ARACIS.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	How the students are using programming techniques for: (i) designing of classes and interfaces, including contracts and invariants; (ii) promote code reuse by inheritance and polymorphism; (iii) using generic programming techniques for collection processing; (iv) using programming techniques for concurrent and multi-threading programming; (v) using object-oriented and functional programming in an integrated approach for the development of flexible and efficient programs; (vi) using design patterns and frameworks	written exam	50%
Laboratory	-Abilities to effectively specify, design, implement and test quality and performance object – oriented programs -Quality of assessment deliverables -Activity during lab sessions -Presence to lab sessions	-Assessment of programming assignments -Written exam	50%
Minimum standard of performance: -To be able to use object-oriented programming techniques in designing and implementing software applications Grade calculus: 40% laboratory + 60% final exam Conditions for participating in the final exam: Laboratory ≥ 5 Handing over all laboratory assignments and obtain a minimum grade of 5 on each assignment; At least 11 laboratory presences. Conditions for promotion: final exam ≥ 5 Obtain a minimum grade of 5 for each category of exam questions (theory and problem).			

Course responsible	Head of department
Prof.dr.eng. Ioan Salomie	Prof.dr.eng. Rodica Potolea

SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	26.

2. Data about the subject

2.1 Subject name	***Operating Systems***				
2.2 Course responsible/lecturer	Conf. dr. eng. Adrian Coleşa – adrian.colesa@cs.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Conf. dr. eng. Adrian Coleşa – adrian.colesa@cs.utcluj.ro Eng. Alexandra Hreniuc – ahreniuc@bitdefender.com Eng. Istvan Szekely – iszekely@bitdefender.com Eng. David Acs – dacs@bitdefender.com Eng. Balint Szabo – bszabo@bitdefender.com Eng. Laslo Ciople – lciople@bitdefender.com Eng. Lilla Nagy – lnagy@bitdefender.com				
2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DD
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars		Laboratory	2	Project	
3.2 Number of hours per semester	56	of which:	Course	28	Seminars		Laboratory	28	Project	

3.3 Individual study:	
(a) Manual, lecture material and notes, bibliography	25
(b) Supplementary study in the library, online and in the field	10
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	28
(d) Tutoring	2
(e) Exams and tests	4
(f) Other activities:	0

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	69
3.5 Total hours per semester (3.2+3.4)	125
3.6 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	Computer Programming, Data Structures and Algorithms
4.6 Competence	C programming

5. Requirements (where appropriate)

5.1. For the course	Blackboard / Whiteboard, Beamer
5.2. For the applications	Computers, Linux, Windows, Blackboard / Whiteboard

6. Specific competence

6.1 Professional competences	**C3:** Problems solving using specific Computer Science and Computer Engineering tools (3 credits) • **C3.1**: Identifying classes of problems and solving methods that are specific to

	computing systems • **C3.2**: Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results • **C3.3**: Applying solution patterns using specific engineering tools and methods • **C3.4**: Evaluating, comparatively and experimentally, the available alternative solutions for performance optimization • **C3.5**: Developing and implementing informatic solutions for concrete problems **C4**: Improving the performances of the hardware, software and communication systems (2 credits) • **C4.1**: Identifying and describing the defining elements of the performances of the hardware, software and communication systems • **C4.2**: Explaining the interaction of the factors that determine the performances of the hardware, software and communication systems • **C4.3**: Applying the fundamental methods and principles for increasing the performances of the hardware, software and communication systems • **C4.4**: Choosing the criteria and evaluation methods of the performances of the hardware, software and communication systems • **C4.5**: Developing professional solutions for hardware, software and communication systems based on performance optimization
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	Provide the students a clear understanding of what an OS is, its role and general functionality and the ability to use fundamental system calls of an OS.
7.2 Specific objectives	Let the students: 1. Know and understand the OS specific terminology. 2. Understand the general structure and functionality of an OS. 3. Understand the specific functionality of the most important OS components, like shell, process manager, file system, memory manager, security manager. 4. Understand the functionality of main synchronization mechanisms and be able to use them to solve real synchronization problems. 5. Be able to write C programs to use an OS's (Linux and Windows) system calls.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction and basic concepts. OS's definition, role, evolution, components, main concepts (file, process, system calls). Basic hardware aspects: CPU, user and kernel mode, memory layers, I/O devices. Basic OS structure.	2	(1) use beamer slides, combined with blackboard illustration; (2) interactions with students: ask their opinion relative to the presented subject; (3) give each class a short evaluation test; let students discuss and argue each other their solution; give them the good	
The Shell (Command Interpreter). Definition, role, functionality, simple and complex commands. Standard input and output redirection.	2		
File systems (1). User Perspective. File and directory concept from the user point of view (definition, role, characteristics, operations).	2		
File systems (2). Windows and Linux File Systems. Permission rights and system calls.	2		
File systems (3). Implementation aspects. Implementation strategies overview, space management and related problems, hard and symbolic links.	2		
Process management. Process model: definition, role, characteristics. Linux and Windows process management system	2		

calls.		solution and let them evaluate their own one; (4) propose 2-3 interesting study cases of OSes to be prepared and presented by students; (5) students are invited to collaborate in research projects.	
Thread management. Thread model: user vs. kernel threads, implementation problems, usage, performance aspects. Basic scheduling algorithms (FIFO, SJF, Priority-based). Linux and Windows process thread system calls.	2		
Process synchronization (1). Theoretical aspects. Context, definition, synchronization mechanisms, techniques and problems (locks, semaphores, monitors, mutual exclusion, starvation, deadlock).	2		
Process synchronization (2). Classical synchronization patterns: producer/consumer, readers/writers, rendez-vous, barrier, dining philosopher, sleeping barber. Similarities between different synchronization mechanisms.	2		
Inter-process communication. Pipe files, shared memory, message queues, signals.	2		
Memory management (1). Context, definition, binding, basic techniques, space management, addresses translation, swapping.	2		
Memory management (2). Paging and segmentation.	2		
I/O Devices Management. Principles, disks, clocks, character-oriented terminals.	2		
Security aspects. Security policies and mechanisms. Basic program's vulnerabilities (buffer overflow).	2		

Bibliography

1. Andrew Tanenbaum. *Modern Operating System*, 2nd Edition, Prentice-Hall, 2005, ISBN 0-13-092641-8.
2. A. Silberschatz, P. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, Wiley, 2010
3. Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau, Operating Systems: Three Easy Pieces, online available at http://pages.cs.wisc.edu/~remzi/OSTEP/

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Laboratory presentation: Purpose, contents, strategies, requirements. **Get familiar with Linux OS**: main characteristics, basic commands, access rights.	2	(1) students are presented a very brief overview of the most important and difficult aspects of the working subject; (2) students are given at the beginning of each class a short evaluation quiz; (3) students are given a hands-on tutorial to practice with working subject's aspects and to solve problems (4) students are given challenging problems for extra credit;	
Linux batch scripts: basic Linux commands, command line structure, scripts, command line parameters, variables, control flow commands, functions.	2		
Linux system calls to access data in files: basic system calls to store and retrieve data to and from regular user files: open, read, write, lseek, close.	2		
Linux system calls for file and directory manipulation: system calls to rename or remove a file, link a file to more directories, get information about a file or directory, change permission rights and listing a directory contents.	2		
Windows case: NTFS and FS system calls.	2		
Linux system calls for process management: system calls for creating a new process, terminating an existing process, waiting for a child process to terminate, loading another executable into an existing process etc.	2		
Linux threads: Linux implementation of POSIX functions used to create and manage threads: pthread_create, pthread_join, pthread_exit etc.	2		
Synchronization mechanisms (1): Linux semaphores. Linux system calls to create and use semaphores: semget, semctl, semop.	2		
Synchronization mechanisms (2): POSIX locks and condition variables. Linux functions used to create and use POSIX locks and condition variables: pthread_mutex_lock, pthread_mutex_unlock, pthread_cond_wait, pthread_cond_signal.	2		
Inter-process Communication Mechanisms (IPC): Linux named (FIFO)	2		

and nameless pipes. System calls for managing and using pipes: pipe and mkfifo.			
Memory management: ELF executable file format. Virtual vs. physical address space. Dynamically allocated memory.	2		
Memory management: memory-mapped files, shared memory.	2		
Security aspects: buffer overflow detection and correction.	2		
Subject review and exam simulation.	2		

Bibliography

1. Lecture slides and laboratory text and support at http://moodle.cs.utcluj.ro/
2. M. Mitchell, J. Oldham, A. Samuel, Advanced Linux Programming, New Riders Publishing, 2001

[*] Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

OS knowledge is a fundamental requirement in the CS field. We follow the ACM curricula guide. We also consult relevant IT companies about their practical expectations regarding OS knowledge and adapt accordingly our course contents. In this sense, Linux and Windows are the most used OSes. Usually the teachers in charge of lab classes are former graduate students of our CS program with consistent experience in industry. They are permanently consulted regarding the OS course curriculum and its applicability in real projects in industry.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Students must understand fundamental OS concepts and be able to correctly define them. They must also be able to apply their knowledge to solve user-space problems related to or dependent by an OS.	Small problem-like subjects requiring students to apply the theoretical learned OS related aspects to give a solution to proposed problem.	0.67
Seminar			
Laboratory	Students must be able to develop C programs that use different OS system calls to solve practical, problems related to or dependent by an OS.	Quiz tests. Programming problems, whose solution has to be implemented in C and run on computers.	0.33
Project			

Minimum standard of performance:

Students must attend minimum **9 lecture classes** to be allowed to take the exam in the regular exam session. Students must attend minimum **7 lecture classes** to be allowed to take the exam in any re-examination sessions. Less than 7 attended lecture classes leads to the interdiction to take any course re-examination in the university year the course is taught.

Students must attend minimum **12 lab classes** to be allowed to take the exam in the regular exam session. Students must attend minimum **10 lab classes** to be allowed to take the exam in any re-examination sessions. Less than 10 attended lab classes leads to the interdiction to take any lab re-examination in the university year the course is taught.

Students are allowed to take the final course examination only after passing the lab examination.

Be able to define the fundamental OS principles and concepts, like process, thread, file, directory, lock, semaphore, paging.

Be able to write C program to use fundamental system calls in Linux for working with files, processes, threads, synchronization mechanisms and memory.

Course responsible
Conf.dr.ing. Adrian Colesa

Head of department
Prof.dr.eng. Rodica Potolea

SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	27.

2. Data about the subject

2.1 Subject name	***Computer Assisted Graphics***				
2.2 Course responsible/lecturer	Prof.dr.eng. Gorgan Dorian – dorian.gorgan@cs.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Lect.dr.eng. Bacu Victor, As.eng. Constantin Nandra, {victor.bacu, constantin.nandra}@cs.utcluj.ro				
2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DF
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars		Laboratory	2	Project	
3.2 Number of hours per semester	56	of which:	Course	28	Seminars		Laboratory	28	Project	
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										20
(b) Supplementary study in the library, online and in the field										6
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										10
(d) Tutoring										3
(e) Exams and tests										5
(f) Other activities:										0

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	44
3.5 Total hours per semester (3.2+3.4)	100
3.6 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1 Curriculum	Computer programming (C language)
4.7 Competence	Applications development in C programming language

5. Requirements (where appropriate)

5.1. For the course	Projector, computer
5.2. For the applications	Laboratory attendance is mandatory Study of laboratory materials from the server

6. Specific competence

6.1 Professional competences	**C3 –** Problems solving using specific Computer Science and Computer Engineering tools (4 credits) **C3.1 –** Identifying classes of problems and solving methods that are specific to computing systems **C3.2 –** Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results **C3.3 –** Applying solution patterns using specific engineering tools and mehods

	C3.4 – Evaluating, comparatively and experimentally, the available alternative solutions for performance optimization **C3.5 –** Developing and implementing informatic solutions for concrete problems
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	Learning about the architecture of a graphic system, the study of the graphic pipeline, the study of 2D graphic algorithms
7.2 Specific objectives	1. Creation of the graphical model of a scene of objects 2. Implementation of the basic algorithms that form the core of a graphic system 3. Development of graphic applications in a high-level programming language (C, C++) 4. Implementation of the main phases of the graphic transformation pipeline

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction. History. Examples	2	New multimedia teaching approaches will be used in classes. The course is interactive and includes demonstrations that exemplify graphical methods and algorithms.	During the semester and before each exam there are a few preparation hours planned.
Graphics systems – architecture, standards	2		
Graphics devices – logic and physics devices, input, output and interactive devices	2		
Graphics transformations pipeline – 2D and 3D transformations. Matrix operators	2		
Mathematics in computer graphics	2		
Lines scan conversion algorithms	2		
Circles scan conversion algorithms	2		
Polygons scan conversion algorithms	2		
Clipping algorithms – point, line, polygon and text	2		
Projections and viewing transformations	2		
Photorealistic presentation of 3D objects – concepts, algorithms, examples	2		
Color models – color perception, color space and standards, color in software design	2		
Graphics formats – vector and raster formats, data compression , Web technologies	2		
Graphics pattern grammars	2		

Bibliography

7. Foley J.D., van Dam, A., Feiner, S.K., Hughes, J.F., "*Computer Graphics. Principles and Practice*". Addison-Wesley Publishing Comp., 1992.
8. Watt A., "*3D Computer Graphics*". Addison-Wesley, 1998.

In virtual library

Course resources, http://cgis.utcluj.ro/teaching/

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Introduction to SDL	2	Documentation and examples will be available to the students, prior to the laboratory classes, on a dedicated server. The students will work independently but will also be assisted by the teacher.	Each student will have to develop a specific project based on the knowledge acquired at the laboratory hours.
Mathematics in computer graphics: vectors	2		
Mathematics in computer graphics: matrices	2		
Graphics transformations	2		
Graphics transformations in SDL	2		
Line rasterization using the Bresenham algorithm	2		
Clipping algorithms for graphical primitives	2		
Viewing transformations	2		
Triangle rasterization using barycentric coordinates	2		
Intermediate assessment	2		
Hidden surface removal using the z-buffer algorithm	2		

Bezier curves	2		
Color computation	2		
Final assessment	2		
Bibliography ***In virtual library*** Course and practical works, http://cgis.utcluj.ro/teaching/			

[*] *Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.*

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

This discipline is integrated into the Computers and Information Technology domain. The content is classic, yet modern, and introduces to students the fundamentals of graphic systems and 2D algorithms. The content of this discipline has been aligned with the information presented in similar disciplines from other major universities and companies from Romania, Europe and USA and has been evaluated by the authorized Romanian governmental agencies (CNEAA and ARACIS).

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	The written exam evaluates the understanding of the information presented in classes and the ability to apply this knowledge. The activity in class evaluates the active involvement of the students in the teaching process and their participation to the discussions, debates and other class activities during the entire semester.	Evaluation is performed through written exam. Evaluation is performed through a very short tests.	60% (E) 10% (AC)
Seminar			
Laboratory	Laboratory assessment evaluates the practical abilities obtained by the students. Through homework assignments the students have the opportunity to develop their skill in applying the notions, concepts and methods presented in class.	Evaluation is performed through written and practical exam.	40% (L)
Project			
Minimum standard of performance: Graduation requirement: M≥5; final mark M=0.5*E+0.4*L+0.1*AC			

Course responsible
Prof.dr.ing. Dorian Gorgan

Head of department
Prof.dr.eng. Rodica Potolea

SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	28.

2. Data about the subject

2.1 Subject name	***Foreign Language II (English, French, German - Technical documents elaboration)***				
2.2 Course responsible/lecturer	Assoc.prof. dr. Sanda Paduretu				
2.3 Teachers in charge of seminars/ laboratory/ project	-				
2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	C
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DC
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Estimated total time

3.1 Number of hours per week	2	of which:	Course	2	Seminars		Laboratory		Project	
3.2 Number of hours per semester	28	of which:	Course	28	Seminars		Laboratory		Project	
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										
(b) Supplementary study in the library, online and in the field										
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										22
(d) Tutoring										
(e) Exams and tests										
(f) Other activities:										

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	22
3.5 Total hours per semester (3.2+3.4)	50
3.6 Number of credit points	2

4. Pre-requisites (where appropriate)

4.1 Curriculum	None
4.8 Competence	Minimum B2 level (CEFR)

5. Requirements (where appropriate)

5.1. For the course	N/A
5.2. For the applications	Class attendance, individual study

6. Specific competence

6.1 Professional competences	N/A
6.2 Cross competences	**CT3 –** Demonstrating the spirit of initiative and action for updating professional, economical and organizational culture knowledge (2 credits)

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	Students should acquire knowledge and integrated skills to communicate in a

	foreign language in professional (technical and engineering) contexts and on job related topics.
7.2 Specific objectives	At the end of this course, the students will be able to: - identify and apply the main principles of effective communication in English - read and write using effective academic and technical writing techniques; -participate and express their opinion, evaluation and recommendation in technical exchange of information; -take notes on specialized topics within their field of specialization; -have the necessary skills read and write scientific articles -read and extract specific and general information from a variety of technical texts;

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction to communication. Communication in an academic setting. Communication at work.	2	Lecture by teacher, drill and practice, class discussion , questions and answers, textbook / reading assignments, formative assessment	
The writing process. Features and stages of the writing process.	2		
Readability. Characteristics and formulae for readability.	2		
Improving readability. Web-page / computer programming readability.	2		
Fundamentals of effective technical writing.	2		
Overview of technical and scientific language used in written communication. Best words and phrases. Reading grammar. Formal and informal language.	2		
Paragraphs. What is a paragraph? Elements of a paragraph. Development of a paragraph.	2		
Basic types of documents. User manuals, technical reports, specification sheets.	2		
Citation: plagiarism, paraphrasing, summary, academic conventions	2		
Plagiarism I: Complexities of definition. Plagiarism in Academic contexts. The Academy's response to plagiarism	2		
Plagiarism II: Learning to write from sources. The "shock" of referencing. Avoiding plagiarism.	2		
Plagiarism III: The art of finding plagiarism. Types of academic misconduct (ghost-writing, contract cheating, falsifying data).	2		
Plagiarism IV: Student's research on typologies of plagiarism. Assignment discussion. Identifying main types (copy-paste, verbatim, translations, disguised, shake and paste, clause quilts, structural, cut and slide, self-plagiarism).	2		
Style. Final conclusion.	2		

Bibliography

1.Marinela Granescu, Ema Adam, Effective academic and technical writing, UTPress, Cluj-Napoca, 2010
2. Justine Jobel, Writing for Computer Science: the art of effective communication, Springer Verlag, Melbourne, 2000
3. Simon Haines, Real writing with answers, Cambridge University Press, 2008
4. R.R. Jordan, Academic writing course, Nelson, 1992

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
-			

Bibliography

-

Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

Mastering the elements of effective academic and technical writing will help the students in the field of computer science to integrate better in the labour market and improve personal development. The introduction in the language for specific purposes and academic discourse will facilitate reading and writing more documents in the field of study, making informed decisions on various types of information, and keeping up-to-date with state of the art knowledge in students' professional field. Most engineers or scientists work in organizational settings where team work is essential and good team work is impossible without good communication.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Completion of end-term evaluation, individual study, attendance to course	On-going class-work evaluation, and one end-term test (integrated skills)	Class-work evaluation - 20% End-term test – 80%
Seminar			
Laboratory			
Project			
Minimum standard of performance: at least 50% of all components of tasks solved correctly.			

Course responsible
Assoc.prof.dr. Sanda Paduretu

Head of department
Conf.univ.dr. Ruxanda Literat

SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	29.

2. Data about the subject

2.1 Subject name	***Sport II***				
2.2 Course responsible/lecturer					
2.3 Teachers in charge of seminars/ laboratory/ project	As.dr. Adrian Suciu				
2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	V
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DC
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Estimated total time

3.1 Number of hours per week	2	of which:	Course		Seminars	2	Laboratory		Project	
3.2 Number of hours per semester	2	of which:	Course		Seminars	28	Laboratory		Project	

3.3 Individual study:	
(a) Manual, lecture material and notes, bibliography	
(b) Supplementary study in the library, online and in the field	
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	
(d) Tutoring	
(e) Exams and tests	22
(f) Other activities:	

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	22
3.5 Total hours per semester (3.2+3.4)	50
3.6 Number of credit points	2

4. Pre-requisites (where appropriate)

4.1 Curriculum	
4.9 Competence	physically fit, necessary skills, knowledge, skills and abilities gained in classes I-XII

5. Requirements (where appropriate)

5.1. For the course	Muncii Blvd, no.103-105, Cluj-Napoca, Politehnica Swimming Complex
5.2. For the applications	Sports Hall, Muncii Blvd, no.103-105, Cluj-Napoca Outdoor and Fitness - Complex Polytechnic

6. Specific competence

6.1 Professional competences	N/A
6.2 Cross competences	**CT2 –** Identifying, describing and conducting processes in the projects management field, assuming different roles inside the team and clearly and concisely describing, verbally or in writing, in Romanian and in an international language, the own results from the activity field.

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	• Harmonious physical development • Maintain health at a high standard
7.2 Specific objectives	• Capacity development effort • Learning and motor skills development • Education volitional qualities

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
-			
Bibliography -			

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Improvement and maintenance of health, athletic ability and fitness	4	interactive	
Improving tehnical exercises learned before using tactic tasks	4		
Automatization of technical and tactics in game conditions (competition).	4		
Learning regulations of different sports, to be able to practice and organize leisure-time sport activity.	4		
Necessary skills to practice independent physical activity	4		
Improving the drills, combinations, schemes in different sport games	4		
Close the school situation by passing physical test	4		
Bibliography 1. Curs de Educaţie fizică – Litografiat UTC-N 2. Dezvoltare fizică generală pentru studenţi – UTC-N 3. Cultură fizică pentru tineret - UTPRES			

Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

Sports activity there in the curriculum of universities and faculties in the country and abroad. Content is consistent with the expectations of professional associates and employers epistemic community representative of the afferent program.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course			
Seminar	70% + 30% Frequency Active Participation, sports skills and advances	By passing control samples	100%
Laboratory			
Project			
Minimum standard of performance: Fulfilling the criteria of evaluation with emphasis on active participation in class, advancements, sports skills.			

Course responsible
As.dr. Adrian Suciu

Head of department
Prof.dr.eng. Rodica Potolea

SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	105.

2. Data about the subject

2.1 Subject name	***Elements of mechanics***				
2.2 Course responsible/lecturer	Şef lucr. dr. ing. Sergiu-Dan Stan				
2.3 Teachers in charge of seminars/ laboratory/ project	Şef lucr. dr. ing. Sergiu-Dan Stan				
2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	C
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DF
	DI – Impusă, DOp – opțională, DFac – facultativă				DFac

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars	2	Laboratory		Project	
3.2 Number of hours per semester	56	of which:	Course	28	Seminars	28	Laboratory		Project	
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										15
(b) Supplementary study in the library, online and in the field										5
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										16
(d) Tutoring										5
(e) Exams and tests										3
(f) Other activities:										

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	44
3.5 Total hours per semester (3.2+3.4)	100
3.6 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1 Curriculum	• Physics, Mathematics
4.10Competence	• Mathematics, Physics,

5. Requirements (where appropriate)

5.1. For the course	Whiteboard, projector, computer
5.2. For the applications	Reading and understanding of the lecture notes.

6. Specific competence

6.1 Professional competences	C1 – Operating with basic Mathematical, Engineering and Computer Science concepts C1.1 – Recognizing and describing concepts that are specific to the fields of calculability, complexity, programming paradigms, and modeling computational and communication systems C1.2 – Using specific theories and tools (algorithms, schemes, models, protocols, etc.) for explaining the structure and the functioning of hardware, software and communication systems

	C1.4 – Formal evaluation of the functional and non-functional characteristics of computing systems C1.5 – Providing a theoretical background for the characteristics of the designed systems
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	• The general objective of the course is to introduce the fundamental principles of mechanical systems, to know the structure, functioning, design fundamentals of mechanical systems that integrates mechanical components, electrical and software technology. • To know the main types of mobile mechanical systems (mechanisms), fundamentals of theory of mechanisms, robot mechanisms and flexible fabrication systems, basic problems in the study of them, used terminology as well as CAD methods of approaching them.
7.2 Specific objectives	The specific objectives are to acquire the knowledge and techniques related to: • usage of methods to determine the functional parameters of different mechanical systems; • to use the mathematical formulae, software package to simulate the different mechanical systems, robot mechanisms, • to analyse and evaluate the experimental data used in mechanical engineering; • to understand, to analyse and compare technical solutions specific to mechanical engineering.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
1. Introduction to mechanical engineering, design and realization of mechatronics systems. V-model as the standard process model for mechatronic system development, mechatronics design methodology.	2	Lecture, visual presentations, demonstrations	
2. Structural analysis of linkages. Degree of Freedom, Classification of Mechanisms.	2		
3. Position Analysis of Mechanisms. Joint variables, Loop Closure Equations, Solution Techniques for Loop Closure Equations.	2		
4. Kinematic analysis of linkages. Transfer function method.	2		
5. Dynamic analysis of linkages.	2		
6. Mechanical Power Transmission. Bar Linkages.	2		
7. Variators. Continuously variable transmission (CVT). Belt drives. Chain drive.	2		
8. Couplings, gearbox, gears, reducers and timing belts. Simple Gear Trains, Planetary Gear Trains.	2		
9. The role of dynamics and kinematics of robotic devices in design of mechatronics systems. Kinematics and dynamics of robotic type devices, articulation, speed, accuracy, bandwidth, inertia, vibration, static and dynamic loading, materials, integration of design requirements.	2		
10. Serial robots. Introduction. Direct and inverse kinematics problems, Examples of kinematics of common serial robots, workspace of a serial robot.	2		
11. Parallel robots. Degrees-of-freedom of parallel mechanisms and manipulators, Active and passive joints, Constraint and loop-closure equations, Direct kinematics problem, Mobility of parallel manipulators, Closed-from and numerical solution, Inverse kinematics of parallel manipulators and mechanisms, Direct kinematics several parallel robots.	2		

12. Robot Jacobians for serial and parallel manipulators, Velocity ellipse and ellipsoids, Singularity analysis for serial and parallel robots.	2		
13. Exoskeleton systems. Hardware Design of the Exoskeleton systems. Mechanical Design. Actuators.	2		
14. Flexible systems of fabrication.	2		

Bibliography

1. Handra-Luca, V., Mecanisme, Ed.UT Pres, Cluj-Napoca, 1981.
2. Handra-Luca, V., ş.a.– Introducere în teoria mecanismelor, Editura Dacia, Cluj-Napoca, vol. I-II, 1982, 1983.
3. Maros, D., ş.a. – Mecanisme, Indrumător de lucrări, Lito. I.P.C-N, Cluj-Napoca, 1984.
4. **** journals (library of TU Cluj-Napoca)
5. **** internet

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
1. Identification of basic elements from the structure of mechanical systems (linkages). Kinematic schema and construction design.	2		
2. Elements of CAD design, modelling and simulation of mechanical structures. Construction variants of mechanical systems. Specific materials from the mechanical systems.	2		
3. Position analysis linkages. Problems.	2		
4. Kinematic analysis of linkages. Problems.	2		
5. Dynamic analysis of linkages.	2		
6 Study of Mechanical Power Transmission.	2		
7. Variators. Continuously variable transmission, gear trains problems.	2		
8. Kinematics and dynamics of robotic systems.	2		
9. Examples of kinematics of common serial robots, workspace of a serial robot.	2		
10. Inverse kinematics of parallel manipulators and mechanisms, workspace of several parallel robots.	2		
11. Inverse kinematics of parallel robots.	2		
12. Singularity analysis for serial and parallel robots.	2		
13. Study of the hardware Design of the Exoskeleton systems. Mechanical Design.	2		
14. Study of flexible systems of fabrication.	2		

Bibliography

1. Handra-Luca, V., Mecanisme, Ed.UT Pres, Cluj-Napoca, 1981.
2. Handra-Luca, V., ş.a.– Introducere în teoria mecanismelor, Editura Dacia, Cluj-Napoca, vol. I-II, 1982, 1983.
3. Maros, D., ş.a. – Mecanisme, Indrumător de lucrări, Lito. I.P.C-N, Cluj-Napoca, 1984.
4. **** journals (library of TU Cluj-Napoca)
5. **** internet

**Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.*

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

Application of the mechanical engineering concepts are specific to most of the engineering disciplines.
The course level is introductory and the intent is to motivate and prepare students for further study in mechanical engineering areas and to conduct projects in real-life applications.
The course content combines theoretical knowledge with applications and focuses on the formulation and solution of specific problems that may occur in various engineering fields.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Knowledge	Midterm exam	40%

	Knowledge	Final exam	60%
Seminar	Answer simple questions from the topic of the lab applications Submitting and defending a miniproject on a given subject	Lab tests (optional) Individual student report (optional)	20% 20%
Laboratory			
Project			
Minimum standard of performance: Solution of simple exercises applying the knowledge and techniques presented in the course.			

Course responsible
Șef lucr. dr. ing. Sergiu-Dan Stan

Head of department
Prof.dr.eng. Rodica Potolea

SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	**Computer Science and Information Technology**
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	106.

2. Data about the subject

2.1 Subject name	***Didactica specialităţii tehnice/ Teaching Methods***				
2.2 Course responsible/lecturer	Prof. Dr. ing. Carmen BAL – carmen.bal@dppd.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Assist drd. ing. Iuhos Carmen Ioana – ioana.iuhos@dppd.utcluj.ro				
2.4 Year of study	II	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DC
	DI – Impusă, DOp – opțională, DFac – facultativă				DFac

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars	2	Laboratory		Project	
3.2 Number of hours per semester	56	of which:	Course	28	Seminars	28	Laboratory		Project	
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										25
(b) Supplementary study in the library, online and in the field										20
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										20
(d) Tutoring										
(e) Exams and tests										2
(f) Other activities:										2

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	69
3.5 Total hours per semester (3.2+3.4)	125
3.6 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	didactica disciplinelor tehnice
4.11Competence	

5. Requirements (where appropriate)

5.1. For the course	• Sala de curs
5.2. For the applications	• Prezenţa la laborator este obligatorie

6. Specific competence

6.1 Professional competences	C1. Operarea cu metodelor şi procedeelor utilizate în predarea disciplinelor tehnice, a instrumentelor de predare-învăţare şi a instrumentelor de evaluare pentru aceste discipline din planul de învăţământ. • C1.1. Cunoaşterea noţiunilor de didactică şi a celor de curriculum . • C1.2. Folosirea corectă a metodelor de învăţământ în cadrul lecţiilor de specialitate tehnică. • C1.3. Utilizarea corectă a obiectivelor şi strategiilor didactice în cadrul lecţiilor de specialitate tehnică.

	• C1.4. Însuşirea de către studenţi a obiectivelor generale ale învăţării disciplinelor de specialitate tehnică în şcoală. • C1.5. Utilizarea corectă a metodelor şi instrumentelor de evaluare în cadrul lecţiilor de specialitate tehnică. C2. Prezentarea unor modele de proiecte didactice.
6.2 Cross competences	CT3 - Autoevaluarea obiectivă şi diagnoza nevoii de formare profesională continuă în scopul inserţiei pe piaţa muncii şi al adaptării la dinamica cerinţelor acesteia şi pentru dezvoltarea personală şi profesională. Autocontrolul învăţării şi utilizarea eficientă a cunoştinţelor de didactica specialităţii tehnice, dezvoltă o buna gestionare a activităţilor personale, precum şi cea de comunicare.

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	• Însuşirea de către studenţi a conceptelor de bază de proiectare didactică a metodelor şi strategiilor de predare învăţare - evaluare, a tehnicilor de formare a echipelor de lucru, planificare a timpului şi întocmirea documentaţiei didactice necesare în procesul de predare – învăţare – evaluare.
7.2 Specific objectives	• Formarea competenţelor de organizare, proiectare şi evaluare a activităţilor didactice la disciplinele tehnice. • *Utilizarea adecvată a conceptelor reformei curiculare.* • Formarea competenţelor de proiectare curriculară în domeniul disciplinelor tehnice. • Înţelegerea necesităţii operaţionalizării obiectivelor educaţionale • Cunoaşterea metodelor de învăţământ utilizate la predarea disciplinelor tehnice. • Cunoaşterea formelor de organizare a activităţii elevilor la disciplinele tehnice. • Formarea competenţelor de evaluare la disciplinelor tehnice.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Conţinutul învăţământului tehnic. Conceptul de Curriculumul. Componentele Curriculum-ului Naţional. Clasificare	2	Expunerea dialogul, problematizarea.	
Organizarea activităţii didactice. Conceptul de lecţie. Tipuri de lecţii.	2	Exemplificare, dialog, comunicarea euristică	
Strategii didactice a profesorului de specialitate. Integrarea mijloacelor de învăţământ în procesul de predare - învăţare - evaluare a disciplinelor de specialitate.	2		
Metode specifice de predare –învăţarea a disciplinelor de specialitate tehnică. Criterii de alegere a metodelor de învăţământ;	2	Comunicare euristică, problematizarea, dialogul	
Proiectarea demersului didactic pentru filiera tehnologică, profil tehnic. Planificarea calendaristică;	2		
Proiectarea unităţii de învăţare; Proiectarea activităţii didactice	2	Comunicare euristică, problematizare, studiu de caz,	
Exigente in stabilirea si formularea obiectivelor educaţionale. Niveluri de definire a obiectivelor educaţionale; Obiective cadru, obiective de referinţa, obiective operaţionale	2	Studiu de caz, realizarea unui mini proiect de lecţie.	
Competente generale, competente; specifice. Transpunerea competenţelor în obiective operaţionale; Metodologia operaţionalizării obiectivelor	2		
Mijloace de învăţământ	2	Conversaţia euristică, problematizarea.	
Alegerea mijloacelor de învăţământ în funcţie de tipul de lecţie	2		
Evaluarea şi funcţiile ei;	2		
Metode de evaluare. Clasificarea acestora	2	Problematizarea, lucrul în grupe, studiu de caz.	
Instrumente de evaluare folosite în cadrul lecţiilor .	2		

Itemi şi clasificarea itemilor de evaluare.	2		

Bibliography

1. Ciot, Gabriela - Elemente de pedagogie şi teoria şi metodologia curriculumului, Ed. Universităţii din Oradea , 2003.
2. Carmen Bal, Noţiuni de didactica specialităţii tehnice, Editura UTPRES Cluj Napoca, 2007;
3. Jurcău Nicolae**,** Carmen Bal (coordonator şi coautor), Metodica disciplinelor tehnice, Editura UTPRES;
4. Jurcău Nicolae**,** Carmen Bal (coordonator şi coautor), Didactica disciplinelor tehnice, Editura UTPRES, Cluj Napoca, 2006;
5. Jurcău, N., - Pedagogie, , U.T.Pres, Cluj, 2001;r
6. Jurcău, N., - Metodica predării disciplinelor tehnice, Atelierul de multiplicare al Institutului Politehnic, Cluj, 1984
7. Ionescu, M. – Lecţia între proiect şi realizare, Ed. Dacia, Cluj 1982

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Finalităţile şi obiectivele studierii disciplinelor tehnice - exemple de programe şcolare din cadrul curriculum-ului Tehnologii.	4	Lucrul pe grupe de 4, cu materiale didactice, plan invatamant, programa şcolară.	
Conţinutul lecţiei - exemple de lucru.	4	Lucrul pe echipe de recunoaştere şi fixare de obiective şi competenţe în funcţie de diferite conţinuturi şi tipuri de lecţii.	
Realizarea unui planificări calendaristice orientative – aplicaţie.Obiectivele lecţiei şi modul de fixare a acestora în cadrul unei lecţii.	4		
Studiu privind metodele de predare-învăţare eficiente pentru atingerea obiectivelor	4	Întocmirea de documente didactice şi realizarea de proiecte de lecţie.	
Eficientizarea metodelor de învăţământ - studiu de caz	4		
Proiectarea didactică. Realizarea unui planificări calendaristice orientative.	4	Realizarea diferitelor proiecte de lecţie	
Obiectivele lecţiei şi modul de fixare a acestora în cadrul unei lecţii.	4	Întocmirea unui portofoliu didactic.	

Bibliography

- Ciot, Gabriela - Elemente de pedagogie şi teoria şi metodologia curriculumului, Ed. Universităţii din Oradea , 2003.
- Carmen Bal, Noţiuni de didactica specialităţii tehnice, Editura UTPRES Cluj Napoca, 2007;
- Jurcău Nicolae**,** Carmen Bal (coordonator şi coautor), Metodica disciplinelor tehnice, Editura UTPRES;
- Jurcău Nicolae**,** Carmen Bal (coordonator şi coautor), Didactica disciplinelor tehnice, Editura UTPRES, Cluj Napoca, 2006;
- Jurcău, N., - Pedagogie, , U.T.Pres, Cluj, 2001;
- Jurcău, N., - Metodica predării disciplinelor tehnice, Atelierul de multiplicare al Institutului Politehnic, Cluj, 1984
- Ionescu, M. – Lecţia între proiect şi realizare, Ed. Dacia, Cluj 1982
- Consiliul Naţional pentru Curriculum - Ghid metodologic pentru aplicarea programelor şcolare, TEHNOLOGII, Liceu tehnologic-profil tehnic, Editat de Aramis Print, 2002.
- Curriculum Naţional. Programe şcolare pentru clasa a IX-a. Volumele 1-3, M.E.N., C.N.C. Editura Cicero, Bucureşti, 1999

*Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

Disciplina este una fundamentală în cadrul modului de psihopedagogie şi transmite studenţilor noţiuni menite să le dezvolte abilităţile de proiectare didactică, utilizarea eficientă a metodelor şi strategiilor de predare - învăţare – evaluare.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course			
Seminar	activitate la seminar – 20%; portofoliu (elaborare proiecte didactice şi teste de evaluare) – 40%;		50% din punctajul evaluarii finale + 50% din punctajul evaluarii

	examinare finală – 40%.		finale.
Laboratory			
Project			
Minimum standard of performance: predarea proiectului de lectie; predarea unui set de probe de evaluare; obţinerea a 50 % din punctajul verificării finale.			

Course responsible
Prof. Dr. ing. Carmen Bal

Head of department
Prof.dr.eng. Rodica Potolea