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	30.

2. Data about the subject

2.1 Subject name	***Design with Microprocessors***				
2.2 Course responsible/lecturer	Prof. dr. eng. Radu Danescu – radu.danescu@cs.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Sl. dr. eng. Mihai Negru – mihai.negru@cs.utcluj.ro As. drd. eng. Razvan Itu - Razvan.Itu@cs.utcluj.ro				
2.4 Year of study	III	2.5 Semester	1	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	1	Project	1
3.2 Number of hours per semester	56	of which:	Course	28	Seminars		Laboratory	14	Project	14
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										28
(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)))	74
3.5 Total hours per semester (3.2+3.4)	130
3.6 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	Computer Architecture, Computer Programming
4.2 Competence	Hardware design, Assembly language programming, C language programming

5. Requirements (where appropriate)

5.1. For the course	Black-board/ White-board, projector, computer
5.2. For the applications	Computer, Atmel Studio, Arduino IDE, Arduino & RPi development boards, Pmods and several other components, modules, sensors etc.

6. Specific competence

6.1 Professional competences	**C2** – Designing hardware, software and communication components (2 credits) **C2.1 -** Describing the structure and operation of hardware, software and communication components **C2.2 -** Explaining the role, interaction and operation of hardware, software and communication components **C2.3 -** Construction of hardware and software components of computing systems using design methods, languages, algorithms, data structures, protocols and technologies **C2.4 -** Metric based evaluation of functional and non-functional characteristics of computing systems

	C2.5 - Implementation of hardware, software and communication components **C5** - Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems (3 credits) **C5.1 -** Specifying the relevant criteria regarding the lifetime cycle, quality, security and computing system's interaction with the environment and human operator **C5.2 -** Using interdisciplinary knowledge for adapting an information system to application domain requirements **C5.3 -** Using fundamental principles and methods for security, reliability and usability assurance of computing systems **C5.4 -** Adequate utilization of quality, safety and security standards in information processing **C5.5 -** Realization of a project including problem identification and analysis, design and development, while proving the understanding of the basic quality needs and requirements
6.2 Cross competences	N/A

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

7.1 General objective	Knowledge, understanding and use of concepts like microprocessor/ microcontroller, bus, memory system, data transfer methods, interface circuits and peripheral devices interfacing, analysis and design of microprocessor systems.
7.2 Specific objectives	To achieve the main objective, specific objectives are pursued: • Knowledge of microprocessors and microcontrollers features and capabilities: hardware capabilities, instruction set architecture, assembly language, and programming solutions. • Knowledge of hardware components used with the microprocessors: electrical and logical characteristics, connection modes. • Development of skills to find solutions based on microprocessors or microcontrollers for real problems with average complexity. • Acquaintance with microcontroller development boards and their software programming tools.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Lecture Overview. Introduction to MP based systems (AVR MCU family)	2	Oral, blackboard and multimedia, interactive teaching style, consultations, involvement of students in research / design.	
AVR registers and instructions	2		
AVR I/O ports and interrupts	2		
Input/output and interrupts for Arduino systems	2		
AVR timers. Timing events with Arduino	2		
Serial data communication. Serial data transfer with Arduino	2		
Analog signals processing	2		
Microcontroller based applications: usage of sensors	2		
Microcontroller based applications: usage of actuators	2		
Introduction to the 8086 microprocessor family	2		
I/O transfer	2		
8086 – the interrupt system	2		
8086 – memory interfacing	2		
DRAM memories. The DMA transfer	2		

Bibliography

1. B. B. Brey, "INTEL Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Prentium ProProcessor, Pentium II, III, 4", ed. 7, Prentice Hall, 2005
2. S. Nedevschi, "Microprocesoare", Editura UTCN, 1994.
3. M.A. Mazidi,S. Naimi, S. Naimi, AVR Microcontroller and Embedded Systems: Using Assembly and C, Prentice Hall, 2010, ISBN 9780138003319.
4. M. Margolis, Arduino Cookbook, 2-nd Edition, O'Reilly, 2012.

Online:

5. http://users.utcluj.ro/~rdanescu/teaching_pmp.html

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

Laboratory			
Introduction to the Arduino boards.	1	Presentation on the blackboard, experiments on microcontroller development boards (Arduino, Raspberry PI, peripherals, sensors), use of specialized IDE design tools (Arduino IDE, Atmel studio), involvement of students in research / design.	
Applications with simple I/O modules	1		
Working with the LCD shield and the interrupt system	1		
Usage of timers	1		
Communication interfaces	1		
Digital sensors. Analogue keypad	1		
Analogue signals processing.	1		
Project			
Project specification	1		
Study of the required technologies	1		
Logic design of the solution.	1		
Implementation of the solution .	1		
Implementation of the solution.	1		
Optimization, testing and validation.	1		
Project assessment.	1		

Bibliography

1. Atmel ATmega2560 - 8 bit AVR Microcontroller datasheet, http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
2. Arduino Mega 2560, http://arduino.cc/en/Main/ArduinoBoardMega2560
3. Abdul Maalik Khan, AVR Project Book, http://www.digisoft.com.pk/products/avr-project-book
4. Mike McRoberts, Beginning Arduino, 2-nd Edition, Technology in Action.
5. M. Margolis, Arduino Cookbook, 2-nd Edition, O'Reilly, 2012.

Online: http://users.utcluj.ro/~rdanescu/teaching_pmp.html

**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 is in the Computer and Information Technology field. Its contents combine fundamentals with specific aspects of the used hardware and software tools, accustoming students with the design principles for microprocessor based systems. The course content was discussed with other universities in the country and abroad, and in conjunction with products /development tools offered by companies in Romania, Europe and the USA (e.g. Digilent, Atmel, Arduino, RaspberryPi) and is rated by the Romanian government agencies (CNEAA and ARACIS).

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Testing theoretical knowledge and problem solving skills	Written exam	50 %
Seminar			
Laboratory	Practical skills for problem solving and implementation of specific problems for applications design. Attendance and activity	Colloquium, lab. work and project evaluation	50 %
Project			

Minimum standard of performance:
Modeling and implementation of typical engineering problems using the theoretical models and applicative tools specific to the domain.
Grade calculus: 25% laboratory + 25% project + 50% final exam
Conditions for participating in the final exam: Laboratory ≥ 5, Project ≥ 5
Conditions for promotion: final exam ≥ 5

Course responsible	Head of department
Prof.dr.eng. Radu Danescu	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	31.

2. Data about the subject

2.1 Subject name	***Logic programming***				
2.2 Course responsible/lecturer	Prof. dr. eng. Rodica Potolea – Rodica.Potolea@cs.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Assoc.prof. dr. eng. Camelia Lemnaru – Camelia.Lemnaru@cs.utcluj.ro				
2.4 Year of study	III	2.5 Semester	1	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	5	of which:	Course	2	Seminars	1	Laboratory	2	Project
3.2 Number of hours per semester	70	of which:	Course	28	Seminars	14	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	10
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	14
(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)))	60
3.5 Total hours per semester (3.2+3.4)	130
3.6 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	Fundamental Algorithms, Programming
4.3 Competence	Logic

5. Requirements (where appropriate)

5.1. For the course	Whiteboard, projector, computer
5.2. For the applications	Computers, specific software (SICStus Prolog). Mandatory attendance of seminars and laboratory works.

6. Specific competence

6.1 Professional competences	**C2** Designing hardware, software and communication components (5 credit points) **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	The main goal of the topic is getting the ability of symbolic processing in general, and logic processing in particular; moreover, acquiring abilities for providing specifications in logic, executable form. Estimating the performance of the solutions designed and implemented in logic formalism.
7.2 Specific objectives	Declarative and procedural semantics Extra-logic operators Meta-programming Data Structures in logic programming. techniques associated with efficiency estimation Incomplete structures, difference lists Types of recursions with advantages and limitations Development of complex applications

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction, first order logic declarative and procedural semantics	2	Interactive Course. Teaching relying on examples, questions and discussions. Continuous evaluation of knowledge aquisition.	
First order logic declarative and procedural semantics (continued)	2		
Negation as failure; Backtracking and cut	2		
Prolog programming techniques	2		
Prolog programming techniques (continued)	2		
Prolog programming techniques (continued)	2		
Prolog programming techniques (continued)	2		
Metalogic predicates	2		
Extra-logic predicates	2		
Nondeterministic Programming	2		
Incomplete data structures; difference lists	2		
Search techniques	2		
Search techniques (continued)	2		
Search techniques (continued)	2		

Bibliography

5. L. Sterling, E. Shapiro, *The Art of Prolog*, MIT Press, 1994.
6. W.F. Clocksin, C.S. Mellish , *Programming în Prolog*, Springer-Verlag Telos, 1994.
7. R. Potolea, *Programare Logică,* vol 1,U.T.Pres, 2007.

8.2 Applications – Seminars/Laboratory	Hours	Teaching methods	Notes
Prolog language	3	Semiras and hands on laboratory works with specific topics. Problem solving with tracing and performance evaluation.	Seminars – design solutions to problem, implementation on board. Laboratory - computer work. (individual)
Sets, sorting	3		
Lists	3		
Basic operations on lists	3		
Incomplete lists; difference lists	3		
Trees	3		
Searching in trees	3		
Incomplete trees	3		
Modeling control structures in Prolog	3		
Graphs	3		
Searching in graphs	3		
Basic graphs algorithms	3		
Metaprogramming	3		
Hands on evaluation	3	Hands on evaluation	mandatory

Bibliography

1. Rodica Potolea, Programare Logica, UT Pres, 2007
2. T.Muresan, R. Potolea, C. Lemnaru, Resources for the laboratory sessions http://users.utcluj.ro/~cameliav/lp.php
3. T. Mureşan, R. Potolea, E. Todoran, A.D. Suciu, *Programare Logică - Indrumător de Laborator*, Romsver, 1998.

*_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

Classical topic of the Computer Science and Information Technology domain, which develops the ability to express executable specifications in a logic language (standard Prolog, Sictus Prolog). The topic enables the assimilation of knowledge and builds necessary skills to other disciplines (AI family), and useful in fundamental / applied research. Ability to analyze specifications and solutions in a unified manner, following partial and total correctness and efficiency at the same time.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Problem solving using specific techniques	Partial Exam (PE) (written) + Final Exam (FE) (written and / or oral)	20% +50%
Seminar	Problem solving	Practical test (Lab) (PC)	30%
Laboratory			
Project			

Minimum standard of performance:
Grade calculus: 20% midterm + 30% laboratory + 50% final exam
Conditions for participating in the final exam: Laboratory ≥ 5
Conditions for promotion: final exam ≥ 5
The laboratory examination can be taken at most twice during one academic year (during the semester and in the winter re-examination session).

Course responsible	Head of department
Prof.dr.eng. Rodica Potolea	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	32.

2. Data about the subject

2.1 Subject name	***Functional programming***				
2.2 Course responsible/lecturer	Lect. dr. eng. Octavian Pop – Octavian.Pop@cs.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Lect. dr. eng. Octavian Pop – Octavian.Pop@cs.utcluj.ro				
2.4 Year of study	III	2.5 Semester	1	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										28
(d) Tutoring										4
(e) Exams and tests										
(f) Other activities:										

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	74
3.5 Total hours per semester (3.2+3.4)	130
3.6 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	Data Structures and Algorithms Course
4.4 Competence	This course assumes no prior knowledge of functional programming, but advises at least one year of programming experience in a regular programming language such as Java, C, C++.

5. Requirements (where appropriate)

5.1. For the course	Basic notions of programming
5.2. For the applications	Linux

6. Specific competence

6.1 Professional competences	**C2** Designing hardware, software and communication components (4 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** Implementation of hardware, software and communication components
6.2 Cross competences	N/A

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

7.1 General objective	Increasing the ability to develop more correct and concise code
7.2 Specific objectives	Writing better code with the concepts introduced by functional programming: high order functions, lazy evaluation, lambda calculus, infinite structure.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction. Programming Paradigms	2	Slides, Various student engagement techniques New examples Quick individual work (1 minute) Homework after each class discussed at the beginning of the next class.	
Basic concepts of programming in Hugs, ML, Lisp: functions, constants, primitive data types, recursion, tuples, infix operators, evaluation.	2		
Basic concepts: local declarations, polymorphism.	2		
Lists: list construction, basic operations on lists.	2		
Lists: polymorphic equality.	2		
Lists: list operators (generators, filters, list expressions).	2		
Trees: alternative data, pattern matching, exceptions, binary trees (list-tree conversions).	2		
Trees: binary trees (binary search trees, AVL balanced trees, examples (operations on sets)).	2		
Trees: binary trees (examples (Huffman codes)), propositional reasoner (example).	2		
Higher-order functions: anonymous functions, partial application, functions as data, data as functions, combinator functions, functionals for lists (list operator style, style without lists).	2		
Infinite data: lazy evaluation, unbounded objects, circular structures.	2		
Transformation and reasoning: structural induction, equivalence of functions, structural induction on trees, induction on number of nodes, general principle of induction.	2		
Lambda calculus: Lambda notation, conversions, combinators.	2		
Para-functional programming: basic language, mapped expressions, eager expressions.	2		

Bibliography

1. Haskell - A Purely Functional Language, http://www.haskell.org/
2. I.A. Leţia, Programare funcţională, Ed. UTPres, UTCN, 1996.
3. H. Conrad Cunningham, Notes on Functional Programming with Haskell, 2007
4. Raul Rojas, A Tutorial Introduction to the Lambda Calculus, FU Berlin, WS-97/98

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Lisp objects, form evaluation, primitive Lisp functions.	2	New examples Tracing algorithms Midterm assessment Miniprojects	
Internal representation, control of evaluation, function definition. Recursion and iteration.	2		
Scope of variables, iterative forms. LAMBDA-expressions, higher-order functions, mapping.	2		
Association lists, properties, arrays and structures. Macrodefinitions, functions as data, surgery.	2		
Trees in Lisp. Graphs and backtracking.	2		
Pattern matching. Symbolic processing.	2		
Lisp microinterpreter. Review of programming in Lisp, in preparation for the lab test.	2		
Lab test (Programming in Lisp).	2		
ML Lists, Recursion,.	2		
ML type checking	2		
ML Trees	2		
Haskell – High order functions	2		
Haskell -Lazy evaluation, circular lists, infinite lists.	2		

Lab test (Programming in ML and Haskell).	2		

Bibliography

1. I.A. Leţia, E.Şt. Chifu, C. Cenan, Programare funcţională. Îndrumător de laborator, Ed. Casa cărţii de ştiinţă, 1999.
2. David S. Touretzky, Common Lisp: A Gentle Introduction to Symbolic Computation, The Benjamin/Cummings Publishing Company,Inc, 1989
3. Andrew Cumming, A gentle introduction to ML, Napier University, Edinburgh, 2013

**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 content of the class is similar to the contents taught at other international universities. The students are encouraged to identify elements of functional programming in the current practice of IT companies running at the local level.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Understanding functional programming elements, Class participation, Homework	Midterm assessment, Writing exam	60%
Seminar			
Laboratory	Quantity and quality of code in Lisp, Haskell and ML	Midterm assessment, Practical exam	40%
Project			

Minimum standard of performance:
Understanding and code writing for the following concepts; Recursion, High Order Functions, Pattern Matching.
Grade calculus: 40% laboratory + 60% final exam
Conditions for participating in the final exam: Laboratory ≥ 5
Conditions for promotion: Grade ≥ 5

Course responsible
Lect.dr.eng. Octavian Pop

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	33.

2. Data about the subject

2.1 Subject name	***Software engineering***
2.2 Course responsible/lecturer	Prof. dr eng. Eneia Todoran – Eneia.Todoran@cs.utcluj.ro
2.3 Teachers in charge of seminars/ laboratory/ project	Assoc.prof. dr. Mitrea Paulina – Paulina.Mitrea@cs.utcluj.ro, Assoc.prof. dr. eng. Mitrea Delia Delia.Mitrea@cs.utcluj.ro

2.4 Year of study	III	2.5 Semester	1	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	1	Project	1
3.2 Number of hours per semester	56	of which:	Course	28	Seminars		Laboratory	14	Project	14
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										25
(b) Supplementary study in the library, online and in the field										17
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										17
(d) Tutoring										5
(e) Exams and tests										10
(f) Other activities:										0

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	74
3.5 Total hours per semester (3.2+3.4)	130
3.6 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	Object Oriented Programming, Programming Techniques
4.5 Competence	Competences acquired in the above disciplines

5. Requirements (where appropriate)

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

6. Specific competence

6.1 Professional competences	**C3 -** Problems solving using specific Computer Science and Computer Engineering tools (2 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 -** Comparatively and experimentaly evaluation of the 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 (1 credit) **C4.1 -** Identifying and describing the defining performance elements of hardware, software and communication systems **C4.2 -** Explaining the interaction of the factors that determine the performances of hardware, software and communication systems **C4.3 -** Applying fundamental methods and principles for increasing performance of hardware, software and communication systems **C4.4 -** Choosing criteria and methods for performance evaluation of hardware, software and communication systems **C4.5 -** Developing performance based professional solutions for hardware, software and communication systems **C5 -** Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems (2 credits) **C5.1 -** Specifying the relevant criteria regarding the lifetime cycle, quality, security and the computing system's interaction with the environment and the human operator **C5.2 -** Using interdisciplinary knowledge for adapting an information system to application domain requirements **C5.3 -** Using fundamental principles and methods for security, reliability and usability assurance of computing systems **C5.4 -** Adequate utilization of quality, safety and security standards in information processing **C5.5 -** Creating a project including the problem's identification and analysis, its design and development, also proving an understanding of the basic quality requirements
6.2 Cross competences	N/A

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

7.1 General objective	The overall objective of discipline consists in the study and application of systematic, disciplined and quantifiable approaches in software systems development
7.2 Specific objectives	• Study and application of software development processes • Understanding the specific activities of software engineering • Knowledge of software engineering models • Knowledge of specific tools that can assist software engineers in the specification, design and validation process • Knowledge of methods for software modeling and performance analysis • Application of processes, methods and tools in small to medium-sized software projects

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction and overview of the course	2		
Software development paradigms: basic paradigms ('waterfall', prototyping, reusable components, formal methods), evolutionary paradigms (incremental development, spiral model, concurrent engineering)	2		
Modern processes: the unified process, agile methods and extreme programming	2		
Basic activities (specification, development, validation, evolution): concepts, principles, processes	2		
Developing requirements: domain analysis, techniques for gathering requirements, capturing requirements as use cases	2		
Formal specification: formal modeling and analysis, model checking, tools in support of formal methods (PRISM)	2		
Modeling with classes: UML class and object diagrams, using design patterns	2		
Modeling with classes: the process of developing class diagrams, semantics of UML class diagrams, implementing class diagrams in Java	2		

Modeling interactions and behavior: UML interaction and state diagrams	2		
Modeling software behavior: UML state diagrams, software performance modeling and analysis.	2		
Architecting and designing software: design principles (increase cohesion, reduce coupling), architectural patterns (Layers, Pipe-and-Filter, etc.)	2		
Testing and inspecting to ensure high quality: testing techniques (equivalence partitioning, path testing) and integration strategies (top-down, bottom-up, scenario-based), inspections	2		
Use case driven development: use case specifications, analysis, design and implementation to realize the use cases, testing the use cases	2		
Program specifications: pre and post assertions, well-founded induction, declarative prototyping	2		

Bibliography

1. I. Sommerville. *Software Engineering* (6th, 7th, 8th, 9th, 10th editions). Addison Wesley (2001, 2004, 2006, 2010, 2015).
2. T. Lethbridge, R. Laganiere. *Object-Oriented Software Engineering: Practical Software Development using UML and Java* (2nd edition). McGraw-Hill, 2005. http://www.lloseng.com.
3. C. Baier, J.P. Katoen, *Principles of Model Checking*. MIT Press, 2008.
4. I. Nikolov. *Scala Design Patterns*. Packt Publishing 2016.
5. E.N. Todoran. *Inginerie software: studii in prototipizare si specificare formala*. Mediamira, Cluj-Napoca, 2006.

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Laboratory			
OCSF – an object client-server framework for reuse oriented development	1		
Simple Chat - an instant messaging system based on OCSF (1)	1		
Simple Chat - an instant messaging system based on OCSF (2)	1		
Using software modeling CASE tools: UML use case, class, interaction, state, component and deployment diagrams	1		
Using CASE tools for performance software modeling and analysis: PRISM, PEPA (1)	1		
Using CASE tools for performance software modeling and analysis: PRISM, PEPA (2)	1		
Test cases design with JUnit	1		
The **project class** attempts to simulate various aspects of the real world of software engineering. The students define the problem to be solved and the scope of the project under the supervision of the teaching assistant. Working alone is permitted, but they are encouraged to work in teams. The students must employ the paradigms and the software development methods that are presented in the taught course. They are expected to deliver three iterations of the project with predefined deadlines. For a traditional 'waterfall' project the deadlines correspond to requirements specification, design, and the final deliverable.			

Bibliography

1. T. Lethbridge, R. Laganiere. *Object-Oriented Software Engineering: Practical Software Development using UML and Java* (2nd edition). McGraw-Hill, 2005. http://www.lloseng.com.
2. PRISM manual, 2016. http://www.prismmodelchecker.org/manual/

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

Software Engineering is a well-established discipline in Computer Science and Information Technology. In this course, students acquire basic knowledge related to software development (paradigms, methods and tools) and learn to apply systematic and quantifiable approaches in the development of software systems. Course content has been developed based on interaction with specialists in Software Engineering from Romania, Europe (UK, Greece) and Canada and has been rated by Romanian government agencies (CNEAA and ARACIS).

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Problem solving skills	Midterm Final exam	15% 60%
Seminar			
Laboratory	Software design and validation skills	Laboratory colloquium,	5%
Project		Project assessment	20%

Minimum standard of performance:
Development of a medium size software project using the skills taught in the Software Engineering course.
Grade calculus: 15% midterm + 5% laboratory + 20% project + 60% final exam
Conditions for participating in the final exam: Laboratory ≥ 5, Project ≥ 5
Conditions for promotion: grade ≥ 5

Course responsible
Prof.dr.eng. Eneia Todoran

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	34.

2. Data about the subject

2.1 Subject name	***Introduction to artificial intelligence***
2.2 Course responsible/lecturer	Prof. dr. eng. Leţia Ioan Alfred – Ioan.Alfred.Letia@cs.utcluj.ro
2.3 Teachers in charge of seminars/ laboratory/ project	Assoc. prof. dr. eng. Groza Adrian – Adrian.Groza@cs.utcluj.ro Assoc. prof. dr. eng. Marginean Anca – Anca.Marginean@cs.utcluj.ro

2.4 Year of study	III	2.5 Semester	1	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	4	Seminars		Laboratory	4	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	18
(b) Supplementary study in the library, online and in the field	5
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	10
(d) Tutoring	6
(e) Exams and tests	9
(f) Other activities:	0

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	48
3.5 Total hours per semester (3.2+3.4)	104
3.6 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1 Curriculum	Logic Programming, Functional Programming
4.6 Competence	Elementary fundamentals of programming

5. Requirements (where appropriate)

5.1. For the course	Projector, Computer
5.2. For the applications	Computers with Linux, Specific Software

6. Specific competence

6.1 Professional competences	**C3 –** Problems solving using specific Computer Science and Computer Engineering tools (1 credit) **C3.1 –** Identification of classes of problems and the methods to solve them characteristic of information systems **C3.2 –** Usage of interdisciplinary knowledge, patterns of solutions and tools, experimentation and interpretation of their results **C3.3 –** Aplication of solution patterns using engineering tools and methods **C3.4 –** Comparative evaluation, including experiments, of alternative solutions, to optimize performance **C3.5 –** Development and implementation of computational solutions for concrete problems

	C5 – Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems (1 credit) **C5.1 –** Stating the criteria relevant to quality, security and system interaction with the environment and human operator **C5.2 –** Usage of interdisciplinary knowledge for the adaptation of the informatic system to the requirements of the application domain **C5.3** – Using fundamental principles and methods for ensuring the security, the safety and ease of exploitation of the computing systems **C5.4 –** Adequate utilization of quality, safety and security standards in information processing **C5.5 –** Realization of a project including problem identification and analysis, design and development, while proving the understanding of the basic quality needs and requirements **C6 –** Designing intelligent systems (2 credits) **C6.1 –** Describing the intelligent systems' components **C6.2 –** Using domain-specific tools for explaining the operation of intelligent systems **C6.3 –** Applying the main methods and principles for specifying solutions for typical problems using intelligent systems **C6.4 –** Choosing criteria and methods for the evaluation of quality, performances and limitations of information systems **C6.5 –** Developing and implementing professional projects for intelligent systems
6.2 Cross competences	N/A

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

7.1 General objective	Knowledge of representation and reasoning of fundamental problems of artificial intelligence
7.2 Specific objectives	Fundamental search methods, Usage of first-order logic and description logics, Basic planning representation and solving methods

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction.	2	Slides, Algorithms, Quality of solutions, Exceptions, Limits in the representation of the real world,	
Intelligent Agents: behavior, environments, structure .	2		
Solving Problems by Searching: uninformed, searching with partial information.	2		
Informed Search Methods and Exploration: heuristics, local search algorithms and optimization problems, local search in continuous spaces.	2		
Constraint Satisfaction Problems: backtracking, local search.	2		
Adversarial Search: alpha-beta pruning, imperfect, real-time decisions, games that include an element of chance.	2		
Logical Agents: knowledge-based agents, propositional logic, effective propositional inference.	2		
First-Order Logic.	2		
Inference in First-Order Logic: forward, backward chaining, resolution.	2		
Knowledge Representation.	2		
Description logics: description languages, terminologies, world description, inferences, reasoning algorithms, language extensions	2		
Planning: partial-order planning, planning graphs.	2		
Planning and Acting in the Real World: schedules and resources, hierarchical network planning, conditional planning, execution monitoring and re-planning, continuous planning.	2		
Course Overview.	2		
Bibliography 1. Artificial Intelligence: A Modern Approach: Russell, Norvig, Prentice Hall, 2002 2. Basic Description Logics: Baader, Nutt, CUP, 2003			
8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes

Introduction to the documentation for the assignment	2	Platform, Documentation, Testing, Examples, New examples	
Studying the documentation for the assignment	2		
Studying the design of the tool	2		
Practicing the exercises provided in the archive	2		
Understanding the main parts of the software	2		
Running the system by tracing at high level	2		
Mastering the running of the system and the examples provided	2		
Conceptual design of new examples	2		
Code for the new examples	2		
Testing and debugging the new cases	2		
Measuring the performance of the system	2		
Documenting the new scenarios	2		
Comparison of the differences between the cases developed and those provided	2		
Final evaluation of the exercises developed	2		
Bibliography Various Artificial Intelligence Tools from the WWW			

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 textbook is one of the most known and used one in the world of the best universities, continuously assessed by the university and research community in the world regarding its influence and use in the software oriented companies.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Problems and theoretical concepts	Written exam	80%
Seminar			
Laboratory	Usage of specific tools on the examples developed and tested by the students	Evaluation in the laboratory	20%
Project			
Minimum standard of performance: Representation of knowledge and its use in solving specific problems using specific tools Grade calculus: 20% laboratory + 80% final exam Conditions for participating in the final exam: Laboratory ≥ 5 Conditions for promotion: grade ≥ 5			

Course responsible
Prof.dr.eng. Ioan Alfred Letia

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	35.

2. Data about the subject

2.1 Subject name	***Economic law***				
2.2 Course responsible/lecturer	Assoc.prof.dr.jur. Roxana Cordos – Roxana.Cordos@mis.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	-				
2.4 Year of study	III	2.5 Semester	1	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										18
(b) Supplementary study in the library, online and in the field										2
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										
(d) Tutoring										2
(e) Exams and tests										2
(f) Other activities:										

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	24
3.5 Total hours per semester (3.2+3.4)	52
3.6 Number of credit points	2

4. Pre-requisites (where appropriate)

4.1 Curriculum	Not the case
4.7 Competence	Not the case

5. Requirements (where appropriate)

5.1. For the course	Not the case
5.2. For the applications	Not the case

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	Applying the general and specific knowledge of technical culture in solving the business issues in this field
7.2 Specific objectives	Knowing the basic legisltion in the field and finding solution for different types of problems.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
General notions of economic legislation.	2	Modern teaching methods	
The merchants. Commerce acts	2		
Bank operations.	2		
Commercial contracts –general notions	2		
Classification of contracts	2		
The contract of sale	2		
The contract leasing.	2		
General rules applied to commercial societies	2		
The constitutive act of a firm	2		
Changes in the constitutive act of a firm.	2		
Types of commercial societies.	2		
The insolvency procedure.	2		
The working contract	2		
General notions of economic legislation.	2		

Bibliography

1. Bacali, L (coord), Antreprenoriat-manualul calificarii, UTPress, 2010 (biblioteca UTCN)
2. Bodu S., Drept commercial completat cu notiuni fundamentale de drept civil- curs universitar, 2005 (biblioteca UTCN)
3. S.Angheni, M.Volonciu, C.Stoica, M.Lostun, Drept comercial, Ed. Oscar Print, Bucuresti, 2000
4. S.Carpenaru, Drept comercial, Ed.All, Bucuresti, 2007
5. I.L.Georgescu, I.Bacanu, Drept comercial român, vol.II, Ed.Lumina Lex, Bucuresti, 2000

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

The students will have the possibility to learn how to put into practice a business idea in the studied domain.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Colloquium	Written test	100%
Seminar			
Laboratory			
Project			
Minimum standard of performance: Grade 5			

Course responsible
Assoc.prof.dr.jur. Roxana Cordos

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	36.

2. Data about the subject

2.1 Subject name	***Graphical Processing Systems***					
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	III	2.5 Semester	1	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	6
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	10
(d) Tutoring	3
(e) Exams and tests	9
(f) Other activities:	0

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	48
3.5 Total hours per semester (3.2+3.4)	104
3.6 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1 Curriculum	Computer programming (C language) Elements of Computer Assisted Graphics
4.8 Competence	Applications development in C programming language, Graphical systems architecture, The graphical processing pipeline

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	**C4 –** Improving the performances of the hardware, software and communication systems (4 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	Study and experiment with the 3D photorealistic algorithms. Development of 2D and 3D graphics applications.
7.2 Specific objectives	1. Creation of the graphic model of a 3D scene of objects 2. Implementation and usage of the fundamental 3D graphics algorithms that can be found in the core of a graphic system 3. Development of graphic applications in a high-level programming language (C, C++) based on graphics libraries (ex. OpenGL) 4. Implementation of the main phases of the graphics transformation pipeline, in order to transform a 3D scene into an image.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Computational graphics	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.
Hidden line and surface removal algorithms. Part 1	2		
Hidden line and surface removal algorithms. Part 2	2		
3D objects modeling	2		
Particles based models	2		
Polygonal objects rendering. Part 1	2		
Polygonal objects rendering. Part 2	2		
Illumination models. Local reflection model. Phong model	2		
Shadow computation	2		
Texture mapping. Part1	2		
Texture mapping. Part2	2		
Global reflection models. Ray-tracing algorithm	2		
Global reflection models. Radiosity algorithm	2		
Graphical animation	2		

Bibliography

8. Watt A., "*3D Computer Graphics*". Addison-Wesley, 1998.
9. Watt A., Policarpo F.:*"3D Games. Real-time Rendering and Software Technology*". Addison-Wesley, 2001.
10. Akenine-Moller T., Haines E., *"Real-Time Rendering".* A.K. Peters 2nd edition, 2002.
11. Foley J.D., van Dam, A., Feiner, S.K., Hughes, J.F., "*Computer Graphics. Principles and Practice*". Addison-Wesley Pblishing Comp., 1992.
12. Gorgan D., Rusu, D., "*Elemente de Grafică pe Calculator*". Cluj-Napoca, 1996.

In virtual library

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

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Mathematics for computer graphics	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.
Introduction to modern OpenGL	2		
Basic vertex and fragment shaders	2		
Debugging methods	2		
3D Transformations	2		
3D models and textures	2		
First project evaluation	2		
Lighting model - Part 1	2		
Lighting model - Part 2	2		
Shadow mapping	2		
Second project evaluation	2		
Cube maps and environmental mapping	2		
Normal mapping	2		
Final project assessment	2		

Bibliography

1. Curse 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 3D graphic systems and 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.eng. 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	108.

2. Data about the subject

2.1 Subject name	***Instruire asistată de calculator***
2.2 Course responsible/lecturer	Prof. Dr. Ing. Carmen BAL – carmen.bal@dppd.utcluj.ro
2.3 Teachers in charge of seminars/ laboratory/ project	Prof. Dr. Ing. Carmen BAL – carmen.bal@dppd.utcluj.ro

2.4 Year of study	III	2.5 Semester	1	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	2	of which:	Course	1	Seminars	1	Laboratory		Project	
3.2 Number of hours per semester	28	of which:	Course	14	Seminars	14	Laboratory		Project	

3.3 Individual study:	
(a) Manual, lecture material and notes, bibliography	5
(b) Supplementary study in the library, online and in the field	10
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	5
(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)))	24
3.5 Total hours per semester (3.2+3.4)	52
3.6 Number of credit points	2

4. Pre-requisites (where appropriate)

4.1 Curriculum	Instruire asistată de calculator
4.9 Competence	

5. Requirements (where appropriate)

5.1. For the course	• Sala de curs, videoproiector,
5.2. For the applications	• Prezenţa la seminar 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 utilizând în procesul educaţional calculatorul. - C1.1. Însuşirea noţiunilor de specialitate necesare utilizării calculatorului în procesul de informare şi formare în învăţământul preuniversitar, a contextului psihopedagogic şi metodic aferent; - C1.2. Operarea cu noţiunile şi metodele specifice instruirii asistate de calculator, proiectării şi dezvoltării curriculare; - C1.3. Utilizarea şi evidenţierea unor tehnici didactice de predare – învăţare - evaluare prin intermediul calculatorului; C2. Formarea unei orientări moderne, dinamice şi prospective asupra

	problematicii cursului.
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 calculator, 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 utilizînd calculatorul ca instrument didactic.
7.2 Specific objectives	• Formarea competenţelor de organizare, proiectare şi evaluare a activităţilor didactice la disciplinele tehnice utilizând calculatorul. • Utilizarea adecvată a conceptelor reformei curiculare. • Formarea competenţelor de proiectare curriculară în domeniul disciplinelor tehnice utilizarea calculatorului şi a softurilor educationale. • Cunoaşterea metodelor de învăţământ utilizate la predarea disciplinelor tehnice**.** • Cunoaşterea formelor de organizare a activităţii elevilor. • Formarea competenţelor de evaluare la disciplinelor tehnice prin utilizarea softurilor educationale.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
1. Noţiuni generale de IAC. Definiţia interacţiunii elev-computer	1	Expunerea dialogul, problematizarea.	
2. Modalităţi de utilizare a calculatorului în procesul de predare învăţare.	1	Exemplificare, dialog , comunicarea euristică	
3. Programe de instruire asistată pe calculator. Softul educaţional	1		
4. Noţiuni de didactică informatică;	1	Comunicare euristică, problematizarea, dialogul	
5. Formarea elevilor/studenţilor prin IAC;	1	Comunicare euristică, problematizarea, dialogul,	
6. TIC ansamblul resurselor de difuzare, stocare şi gestionare a informaţiei destinată procesului educativ.	1	Comunicare euristică, problematizare, studiu de caz,	
7. Educaţia la distanţă noţiuni de e-learning	1	Studiu de caz, realizarea unui mini proiect de lecţie.	

Bibliography

1. Adăscăliţei, Adrian (2007) : Instruire asistată de calculator. Didactică informatică, Ed. Polirom, Iaşi.
2. Carmen Bal, Instruire Asistata de Calculator, de la teorie la practică, Editura ALMA MATER, 2009, ISBN978-606-504-066-3.
3. Bârză, Silviu (2002) : Bazele informaticii şi noţiuni de birotică. Ed. Fundaţiei României de mâine, Bucureşti.
4. Creţu, Carmen (1999) : Teoria curriculum-ului şi conţinuturile educaţiei, Ed. Univ. „Al. I. Cuza”, Iaşi.
5. Cucoş, Constantin (1999) : Pedagogie, Polirom, Iaşi.
6. Damian, Alexandru-Miron (2000-2001) : Teoria şi metodologia instruirii, Ed. Fundaţiei „România de Mâine”, Bucureşti.
7. Ionescu, C. (1998) : Metodica predării informaticii, Univ. Babeş-Bolyai, Cluj.

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
1. Elaborarea unui program de instruire	1	Lucrul pe grupe de 4, cu materiale didactice,	
2. Aspecte specifice ale proiectării activităţilor didactice.	1	Lucrul pe echipe şi realizarea de proiecte de lecţie pe calculator.	
3. Proiectarea unei lecţii de specialitate cu ajutorul calculatorului sau cu ajutorul unui soft educaţional..	1		
4. Comparaţie între două metode în predarea a aceluiaşi conţinut	1	Întocmirea de documente	

5. Simularea predării unei lecţii de specialitate cu ajutorul unui soft educaţional (AEL)	1	didactice şi realizarea de proiecte de lecţie pe calculator.	
6. Aplicaţie. Elaborarea unui proiect de lecţie cu ajutorul computerului.	1	Realizarea diferitelor proiecte de lecţie	
7. Evaluarea prin intermediul calculatorului	1	Întocmirea unui portofoliu didactic.	

Bibliography

1. Adăscăliţei, Adrian (2007) : Instruire asistată de calculator. Didactică informatică, Ed. Polirom, Iaşi.
2. Carmen Bal, (2009), Instruire Asistata de Calculator, de la teorie la practică, Editura ALMA MATER, , ISBN978-606-504-066-3.
3. Bârză, Silviu (2002) : Bazele informaticii şi noţiuni de birotică. Ed. Fundaţiei României de mâine, Bucureşti.
4. Creţu, Carmen (1999) : Teoria curriculum-ului şi conţinuturile educaţiei, Ed. Univ. „Al. I. Cuza”, Iaşi.
5. Cucoş, Constantin (1999) : Pedagogie, Polirom, Iaşi.
6. Damian, Alexandru-Miron (2000-2001) : Teoria şi metodologia instruirii, Ed. Fundaţiei „România de Mâine”, Bucureşti.
7. Ionescu, C. (1998) : Metodica predării informaticii, Univ. Babeş-Bolyai, Cluj

**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 cu ajutorul calculatorului.

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%; examinare finală – 40%.		50% din punctajul evaluarii finale + 50% din punctajul evaluarii 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.eng. Carmen Bal

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	109.

2. Data about the subject

2.1 Subject name	***Practica pedagogica nivel I licenta***
2.2 Course responsible/lecturer	Prof. Dr. Ing. Carmen BAL – carmen.bal@dppd.utcluj.ro
2.3 Teachers in charge of seminars/ laboratory/ project	Prof. Dr. Ing. Carmen BAL – carmen.bal@dppd.utcluj.ro

2.4 Year of study	III	2.5 Semester	1	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	DFac

3. Estimated total time

3.1 Number of hours per week	3	of which:	Course		Seminars		Laboratory		Project	3
3.2 Number of hours per semester	42	of which:	Course		Seminars		Laboratory		Project	42

3.3 Individual study:	
(a) Manual, lecture material and notes, bibliography	12
(b) Supplementary study in the library, online and in the field	4
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	6
(d) Tutoring	6
(e) Exams and tests	4
(f) Other activities:	4

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	36
3.5 Total hours per semester (3.2+3.4)	78
3.6 Number of credit points	3

4. Pre-requisites (where appropriate)

4.1 Curriculum	Cunostinte de bază în ştiintele educaţiei, dobîndite pe parcursul studiilor de modul psihopedagogic, prin experienţă profesională sau si in contexte4 nonformale msau informale de invăţare.
4.10Competence	Competenţe de operare pe calculator (Word, Excel, Power Point şi Internet Explorer)

5. Requirements (where appropriate)

5.1. For the course	• Participare activă; • Lectura suportului de curs
5.2. For the applications	• Lectura bibliografiei recomandate; • Elaborarea şi sustinerrea lucrarilor planificate şi asamblarea acestora într-un portofoliu de evaluare; • Participare active.

6. Specific competence

6.1 Professional competences	C1 Utilizarea, interpretarea , prelucrarea şi aplicarea cunoştinţelor de specialitate psihopedagogice şi metodologice în cadrl întregului demers didcatic de proiectare a activităţilor instructiv-educative şi a materialelor didactice; C2 Identificarea şi apliocarea principiilor şi strategiilor didactice în proiectarea

	activităţilor instructiv educative specifice nivelului de vârstă al clasei cuc are lucrează; C3. Elaborarea modelelor de proiectare a activităţilor instructiv educative şi /sau extracurriculare.
6.2 Cross competences	• CT1 – Aplicarea principiilor şi a nhormelor de deontologie profesională fundamentale pe opţiuni valorice explicite, specifice specialistului în ştiinţele educaţiei. • CT2 – Cooperarea eficientă în echipe de lucru profesionale, interdisciplinare, specifice dersfăşurării proiectelor şi programelor educaţionale; • CT3Utilizarea metodelor şi tehn icilor eficiente de învăţare pe tot parcursul vieţiiîn vederea formării şi dezvoltării profesionale; • CT3 – Prtomovarea v alorilor unui învăţământ de calitate, în conformitate cu politicile educaţionaleinterne şi în acord cu cele elaborate şi popularizate la nivel european.

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

7.1 General objective	Cunoaşterea specificului cercetării procesului de învăţământ (caracterisitici, etape, funcţii, tipuri, metodologii etc.) din pertspectiva practiocii pedagogice desfăşurate în cadrul învăţământului preuniversitar).
7.2 Specific objectives	Dezvoltarea capacităţii de observare, consemnare, analiză şi apreciere a activităţilor instructiv-educsative; Formarea unuzi sistem de capacităţi opertaţionale de a proiecta, realiza şi evalua activităţile instructiv-educative: capacitatea de a proiecta activităţi inegral, de diferite tipuri şi variante, precum şi alte forme de organizare a procesului de învăţământ; capacitatea de a conduce integral activităţi de tipuri/variante diferite; capacitatea de a măsura, aprecia, decide cu privire la desfăşurarea unor activităţi, capacitatea de a regla/autoregla activităţile în funcţie de rezultatele evaluării; Dezvoltarea capacităţii de a colabora cu diferiţi factori educativi, antrenându-i în activităţile instructiv-educative.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Bibliography -			
8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Observarea şi înregtistrarea integraslă a diferitellor tipuri/variante de lecţii, cu ajutorul unor instrumente şcolare (grile, fişe, ghiduri, etc.).	3	Practică observativă Practică efectivă Dezbarea în grup	
Analiza, dezbaterea şi aprecierea în grup a lecţiilor observate, cel puţin 3-4 variante de lecţii pentru fiecarte tip categorie de lecţie şi 1-2 forme de activitate.	3		
Elaborarea proiectului unor unităţi de învăţare şi a unor lecţii de tipuri şi variante diferite, precum şi a altor forme de organizare a procesujlui de învăţământ.	3		
Conducerea integrală a unor lecţii de tipuri şi variante diferite, precum şi a altor forme de organizare a procesujlui de învăţământ, conform planificării rea lizate de coordonatorul şi mentorul de proactică pedagogică.	3		
Utilizarea unor instrumente de evaluare (autoevaluarea) lecţiei/sistemelor de lecţii şi a altor forme de organizare a procesului de învăţământ; măsurarea şi aprecierea realizării unor obiective şi a lecţiei integral.	3		
Exerciţii de elaborare a unor alternative de lecţii, integral sau pe secvenţe, în funcţie de rezultatele evaluării.	3		
Exersarea unor atitudini pozitive faţă de elevi şi profesie şi a unor atitudini creative în desfăşurarea activităţilor instructiv-educative.	3		
Aplicarea creatoare, la specificul situaiei, a principalelor tehnici de învăţare eficientă – stilul activităţilor intelectuale. Aplicarea unor metode şi procedee de prevenire şi comb atere a rămânerii în urmă la	3		

învăţătura a unor elevii			
Aplicarea unor strategii de identificare şi dezvoltare a înclinaţiilor şi aptitudinilor elevilor, părin individualizarea activităţilor de învăţare în scopul dezvoltării performanţelor maxime..	3		
Aplicarea unor strategii caracteristice pentru dezvoltarea cooperării/comunicării şi dezvoltării unor relaţii psihosociale pozitiver /simulativeâ, a unor motive superioare de apartenenţă de grup, de afiliere, de dezvoltare a grupului ca entitate etc.	3		
Recunoaşterea (identificarea) caracterisiticilor unei cercetări, a etapelor, funcţiilor etc. Prin analiza unei cercetări empăirice desfăşurate la nivelul unităţii şcolare, prin discuţie de grup.	3		
Aplicarea în cadrul unui proiect de cercetare a metodelor principale de cercetare: dezbaterea, argumentarea observarea, experimentul, ancheta, etc.	3		
Utilizarea tehnicilor de negociere argumentare, contraargumentare, de prognoză, de raţionare şi exprimare, de persuasiune.	3		
Activităţi practice de sfătuire a elevilor pentru a valorifica plenar valenţele timpului liber pentru recreere şi autodezvoltare.	3		
Bibliography 1. Curriculum-ul pentru invăţământul preuniveristare tehnic (plan de invăţământ, programe scolare pentru clasele V-VII, IX- XII), ghiduri, îndrumătoare, manuale de specialitate etc. 2. Carmen Bal, Noţiuni de didactica specialităţii tehnice, Editura UTPRES Cluj Napoca, 2007;			

**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

Conţinuturile disciplinei acoperă un segment foarte important al formării profesionalela nivel de licenţă fiind în acord cu aşteptările comunităţii specialiţştilor în domenikul tehnic şi în cel al angajatorilor din domeniul educaţional tehnic.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course			
Seminar			
Laboratory			
Project	Practică observativă; Practică efectorie.	Portofoliu de practică pedagogică	100
Minimum standard of performance: 70% rezultat după însumarea puntajelor ponderate			

Course responsible
Prof.dr.eng. Carmen Bal

Head of department
Prof.dr.eng. Rodica Potolea