SYLLABUS

1. Data about the program of study

1.1	Institution	The Technical University of Cluj-Napoca
1.2	Faculty	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	37.

2. Data about the subject

2.1	Subject name					Structure of Computer Systems					
2.2	Subject area					Computer Science and Information Technology					
2.3	Course responsible/lecturer					Prof. dr. eng. Gheorghe Sebestyen – Gheorghe.Sebestyen@cs.utcluj.ro					
2.4	Teachers in charge of applications					S.l.dr.eng. Anca Hangan, As.dr.eng. Madalin Neagu, drd. Vlad Miclea					
2.5	Year of study	III	2.6	Semester	6	2.7	Assessment	exam	2.8	Subject category	DID/OB

3. Estimated total time

Sem.	Subject name	Lecture	Applications			Lecture	Applications			Individual study	TOTAL	Credit
		[hours / week.]				[hours / semester]						
			S	L	P		S	L	P			
6	**Structure of Computer Systems**	**2**	**-**	**2**	**1**	**28**	**-**	**28**	**14**	**34**	**104**	**4**

3.1	Number of hours per week	5	3.2	of which, course	2	3.3	applications	3
3.4	Total hours in the teaching plan	70	3.5	of which, course	28	3.6	applications	42
Individual study								Hours
Manual, lecture material and notes, bibliography								14
Supplementary study in the library, online and in the field								4
Preparation for seminars/laboratory works, homework, reports, portfolios, essays								13
Tutoring								0
Exams and tests								3
Other activities								0

3.7	Total hours of individual study	34
3.8	Total hours per semester	104
3.9	Number of credit points	4

4. Pre-requisites (where appropriate)

4.1	Curriculum	Digital system design, Computer architecture
4.2	Competence	Understand and operate with basic concepts regarding computer system's hardware

5. Requirements (where appropriate)

5.1	For the course	
5.2	For the applications	

6. Specific competences

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
Cross competences	N/A

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

7.1	General objective	The main goal of the course is to present in an accessible way advanced design methods and techniques used in today's microprocessors and computer systems
7.2	Specific objectives	To study: Methods and metrics for computer performance assessment Advanced CPU designs (pipelining, multicore, parallele and distributed computing) Memory hierarchies: cache memory, virtual memory, new DRAM technologies RISC architecture Parallel computers architectures – hardware issues and solutions

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction. Computer Performance Parameters and Methods of Improvement	Lecture based on slides	
2	Computer performance and optimality, Benchmarking		
3	The Arithmetical and Logical Unit (ALU)		
4	The Central Processing Unit (CPU) – MIPS architecture, pipeline, hazard cases		
5	The Central Processing Unit – advance techniques: Scoreboard method, Tomasulo's algorithm, Branch prediction techniques		
6	The Central Processing Unit – multi-core systems		
7	Microprocessors – basic components and advanced implementations		
8	Memory System – memory technologies (SRAM, DRAM) and design principles		
9	Memory Hierarchies – cache and virtual memory		
10	Interconnection Systems – serial and parallel synchronous and asynchronous buses, multipoint interconnections		
11	Parallel Computer Architectures - different levels of parallel execution		
12	RISC Architectures – principles and implementation examples		
13	Distributed Computing – GRID and Cloud Systems		
14	Technological Perspectives in Computer Architectures		

Bibliography

1. Gorgan Dorian, Sebestyen Gheorghe, Structura Calculatoarelor, Editura albastra, Cluj-Napoca 2005
2. Hennessy John, Patterson David, Computer architecture, a Quantitative Approach, Ed. Elsevier, 2007
3. Baruch, Z. F., Structure of Computer Systems, U.T.PRES, Cluj-Napoca, 2002, ISBN 973-8335-44-2.

8.2. Applications (Laboratory)		Teaching methods	Notes
1	Measuring the performance of computer systems with benchmarks	Practical designs,	
2	CPU performance monitoring using the Time-Stamp Counter register		

3	Programming elements in VHDL	experiments and results assesment	
4	Design of ALU components		
5	FPGA Synthesis		
6	Introduction to using PicoBlaze microcontroller with the Nexys3 board		
7	Implementation of a MIPS processor in VHDL - 1		
8	Implementation of a MIPS processor in VHDL - 2		
9	Implementation of a pipelined MIPS processor in VHDL		
10	Memory design - 1		
11	Memory design - 2		
12	Advanced Hardware Design Techniques		
13	Design implementations on NEXYS 3 board		
14	Laboratory Colloquy		
Topics for Project Assignments: Implementation of arithmetic circuits; Design and implementation of processors and controllers; Signal Processing; Hardware implementation of DSP and image processing algorithms; Design of I/O interfaces.			
Bibliography 1. Laboratory works at http://users.utcluj.ro/~ancapop/scs.html			

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

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		Theoretical knowledge level		Written exam		60%
Applications		Hardware Design skills		Practical evaluation		40%
10.4 Minimum standard of performance						
Minimum 5 for the Course and for the Application assesment						

Course responsible
Prof. dr. eng. Gheorghe Sebestyen

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	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	38.

2. Data about the subject

2.1	Subject name					Formal Languages and Translators					
2.2	Subject area					Computer Science and Information Technology					
2.3	Course responsible/lecturer					Assoc.prof. dr.eng. Emil Şt. Chifu – emil.chifu@cs.utcluj.ro					
2.4	Teachers in charge of applications					Ing. Mihai Anton Cerghizan					
2.5	Year of study	III	2.6	Semester	6	2.7	Assessment	exam	2.8	Subject category	DID/OB

3. Estimated total time

Sem.	Subject name	Lecture	Applications			Lecture	Applications			Individual study	TOTAL	Credit
		[hours / week.]				[hours / semester]						
			S	L	P		S	L	P			
6	**Formal Languages and Translators**	**2**	**-**	**2**	**-**	**28**	**-**	**28**	**-**	**48**	**104**	**4**

3.1	Number of hours per week	4	3.2	of which, course	2	3.3	applications	2
3.4	Total hours in the teaching plan	56	3.5	of which, course	28	3.6	applications	28

Individual study	Hours
Manual, lecture material and notes, bibliography	17
Supplementary study in the library, online and in the field	7
Preparation for seminars/laboratory works, homework, reports, portfolios, essays	16
Tutoring	5
Exams and tests	3
Other activities	0

3.7	Total hours of individual study	48
3.8	Total hours per semester	104
3.9	Number of credit points	4

4. Pre-requisites (where appropriate)

4.1	Curriculum	Computer Programming, Data Structures and Algorithms
4.2	Competence	Basic knowledge of programming and data structures (preferably in the C language)

5. Requirements (where appropriate)

5.1	For the course	N/A
5.2	For the applications	Computers, specific software

6. Specific competences

Professional competences	**C1 –** Operating with basic Mathematical, Engineering and Computer Science concepts (2 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 **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
Cross competences	N/A

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

7.1	General objective	- To know the phases, components, and algorithms used by typical language translators. - To provide a formal basis for the development of concepts relating to lexical and syntactic processors in translators.
7.2	Specific objectives	- To know the underlying formal models such as finite state automata and push-down automata, and to understand their connection to language definition through regular expressions and grammars. - To understand the relationships between formal descriptions of the automata in the formal language theory and their practical implementations as lexical and syntactic analyzers in translators. - To know the classes of languages for which a deterministic parser can be implemented. - To describe the syntax of languages to be implemented by using grammars and regular expressions. - To design, develop and test a software project, by utilizing specialized software tools (parser generators), in order to arrive at a translator for an artificial language. - To master and control the phenomena of ambiguity and nondeterminism (conflicts) which occur when using parser generators and lexical analyzer generators.

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Descriptive tools: strings and rewriting systems, grammars.	- The main ideas with multimedia tehniques - Details and examples at the blackboard, in interaction with the students - There are consultation hours - Students are	N/A
2	Descriptive tools: derivations and parse trees.		
3	Regular grammars and finite automata: finite automata.		
4	Regular grammars and finite automata: state diagrams and regular expressions.		
5	Context-free grammars and pushdown automata: pushdown automata.		
6	Top-down analysis and LL(*k*) grammars: LL(*k*) grammars		
7	Top-down analysis and LL(*k*) grammars: the LL(*k*) algorithm		
8	Top-down analysis and LL(*k*) grammars: elimination of left recursion, left factoring.		
9	LL parsers: strong LL(*k*) grammars, the LL(1) parsing algorithm.		
10	LL parsers: the LL(1) parsing algorithm, computation of FIRST and FOLLOW		

	sets.	invited to collaborate in research projects	
11	Bottom-up analysis and LR(*k*) grammars: situations and closure of a nonterminal, the LR(*k*) algorithm.		
12	Bottom-up analysis and LR(*k*) grammars:the LR(*k*) algorithm.		
13	LR parsers: the LR(0) parsing algorithm.		
14	LR parsers: LR(0) states.		

Bibliography

1. W.M. Waite and G. Goos, Compiler Construction, Springer-Verlag, 1984.
2. I.A. Leţia and E.Şt. Chifu, Limbaje formale şi translatoare, Ed. Casa cărţii de ştiinţă, 1998.
3. A.V. Aho, R. Sethi, and J.D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1986.

8.2. Applications (Seminars, Laboratory, Projects)		Teaching methods	Notes
1	Lexical analyzer for C.	Brief presentation at the blackboard, implementing and testing homeworks on the computer, individual assignment on the computer	N/A
2	The generator of lexical analyzers Lex: Lex source, Lex regular expressions, Lex actions, ambiguous rules, Lex source definitions.		
3	Lex generator: left context sensitivity, examples.		
4	The bottom-up parser generator Yacc: basic specifications, Yacc syntax, actions, lexical analysis, how the parser works.		
5	Yacc generator: ambiguity and conflicts, precedence and associativity, error handling, the Yacc environment, hints for preparing specifications.		
6	Yacc generator: support for arbitrary value types, examples (expression evaluator).		
7	Yacc/ Lex applications: interpreter for a language operating on lists.		
8	Yacc/ Lex applications: interpreter for a language operating on binary trees.		
9	Yacc/ Lex applications::interpreter for a language operating on matrices.		
10	Yacc/ Lex applications: code generator for an imperative language.		
11	Yacc/ Lex test		
12	Building recursive-descent (RD) parsers: expression parser.		
13	RD parsers: parser for a language operating on binary trees.		
14	RD parsers: parser for a language operating on lists.		

Bibliography

1. The Lex & Yacc Page, http://www.combo.org/lex_yacc_page/
2. I.A. Leţia, D. Marcu, B. Ungureanu, Procesoare de limbaje. Îndrumător de laborator, Universitatea Tehnică din Cluj-Napoca, 1995.

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

It is a specialty course in Computer Science, its syllabus being both classical and modern. It teaches the students with the basic principles in the design of interpreters and translators for artificial languages. The syllabus of the course has been discussed with other important universities and companies from Romania, Europe, and USA. This syllabus has been evaluated by Romanian governmental agencies (CNEAA and ARACIS).

10. Evaluation

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Weight in the final grade
Course	- Problem-solving skills - Attendance, Activity	- Written exam	55%
Applications	- Problem-solving skills - Attendance, Activity	- Assessment of the Yacc/ Lex activity and test - Assessment of the RD activity and written exam	30% 15%
10.4 Minimum standard of performance			
Modeling a typical engineering problems using the domain specific formal apparatus. Obtaining final grade 5			

Course responsible
Assoc.prof. dr. eng. Emil Chifu

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	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	39.

2. Data about the subject

2.1	Subject name					Management and communication					
2.2	Subject area					Computer Science and Information Technology					
2.3	Course responsible/lecturer					Lect. dr. Veronica Maier					
2.4	Teachers in charge of applications										
2.5	Year of study	III	2.6	Semester	6	2.7	Assessment	Colloquium	2.8	Subject category	DC/OB

3. Estimated total time

Sem.	Subject name	Lecture	Applications			Lecture	Applications			Individual study	TOTAL	Credit
		[hours / week.]				[hours / semester]						
			S	L	P		S	L	P			
6	**Management and communication**	**2**	**-**	**-**	**-**	**28**	**-**	**-**	**-**	**24**	**52**	**2**

3.1	Number of hours per week	2	3.2	of which, course	2	3.3	applications	-
3.4	Total hours in the teaching plan	28	3.5	of which, course	28	3.6	applications	-
Individual study								Hours
Manual, lecture material and notes, bibliography								6
Supplementary study in the library, online and in the field								7
Preparation for seminars/laboratory works, homework, reports, portfolios, essays								6
Tutoring								3
Exams and tests								2
Other activities								

3.7	Total hours of individual study	24
3.8	Total hours per semester	52
3.9	Number of credit points	2

4. Pre-requisites (where appropriate)

4.1	Curriculum	Not the case
4.2	Competence	Not the case

5. Requirements (where appropriate)

5.1	For the course	The existence of multimedia equipment
5.2	For the applications	Not the case

6. Specific competences

Professional competences	**C5 –** Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems (1 credit) **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
Cross competences	**CT2 –** Identifying, describing and conducting processes in the projects management field, undertaking different team roles and clearly and concisely describing own profesional results, verbally or in writing, in Romanian and in an international language. (1 credit)

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

7.1	General objective	Understand, assimilate and use of basic management and communication concepts, principles and techniques
7.2	Specific objectives	Understand the basic managerial functions, the organization's internal environment, the motivation of people, the organizational communication, communication barriers, increasing communication, overcoming of internal conflicts and the link between leadership and communication.

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction in management: management concept, managerial functions, the managers, challenges confronting contemporary management	multimedia presentation, interactivity by exemplifying the presented concepts, using the questions-answer method during the course, discussing case studies, playing thematic strategy game, interactive lectures	2h
2	Organization's internal environment: culture and business ethics		2h
3	External environment (general and task environment components)		4h
4	Planning		2h
5	Organizing		2h
6	Coordinating and motivating people		2h
7	Controlling and performance assessment		2h
8	Organizational communication (content, functions, types, networks); interpersonal and group communications		2h
9	Communication barriers		2h
10	Increasing the effectiveness of communication		2h
11	Conflict and conflict management		2h
12	Negotiation and assertive communication		2h
13	Leadership and communication		2h

Bibliography

1. Catana D., Dobra Constantinescu A., Management in Power Point, UTPRES 2004
2. Adler, R.B., Elmhorst,J.M, Communicating at work, Principles and practices for Business and the professions, 8th ed., McGraw Hill, 2005
3. Becker, E.F., Wortmann, J., Mastering communication at work: how to lead, manage and influence, McGraw Hill Professional, 2009, disponibil pe: http://books.google.ro/books/about/Mastering_Communication_at_Work.html?id=0G6LuTp6XhsC&redir_esc=y
4. Bell, A.H., Smith, D.M., Management communication, 2nd ed., John Wiley&Sons Inc., 2006
5. Nicolescu, O. Fundamentele managementului organizaţiei, Editura Universitară, 2008
6. Nicolescu, O., Verboncu, I. Managementul organizaţiei, Editura Economică, Bucureşti, 2007

8.2. Applications (Seminars, Laboratory, Projects)	Teaching methods	Notes

1	Not the case	-	-
Bibliography			

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

In order to provide the content for the lecture and establish the method of the teaching / learning process the professor organizes meetings with entrepreneurs, who speak to students about the needs and expectations of employers in the area.

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		The students answer to open ended and closed questions; involvement during the course by preparing and presenting teamwork papers.		Written exam		100%
Applications		-		-		-
10.4 Minimum standard of performance						
Requirement for the credits: N>5						

Course responsible
Assist. Prof. Veronica Maier, PhD

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	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	40.

2. Data about the subject

2.1	Subject name	Image Processing
2.2	Subject area	Computer Science and Information Technology
2.3	Course responsible/lecturer	Prof dr. eng. Sergiu Nedevschi (Sergiu.Nedevschi@cs.utcluj.ro)
2.4	Teachers in charge of applications	Assoc. Prof. dr. eng. Florin Oniga, Assist. Prof. dr. eng. Ion Giosan, Assist. Prof. dr. eng. Raluca Brehar, {Florin.Oniga, Ion.Giosan, Raluca.Brehar}@cs.utcluj.ro

2.5	Year of study	III	2.6	Semester	6	2.7	Assessment	exam	2.8	Subject category	DID/OB

3. Estimated total time

Sem.	Subject name	Lecture	Applications			Lecture	Applications			Individual study	TOTAL	Credit
		[hours / week.]				[hours / semester]						
			S	L	P		S	L	P			
6	**Image Processing**	**2**	**-**	**2**	**1**	**28**	**-**	**28**	**14**	**34**	**104**	**4**

3.1	Number of hours per week	5	3.2	of which, course	2	3.3	applications	3
3.4	Total hours in the teaching plan	70	3.5	of which, course	28	3.6	applications	42
Individual study								Hours
Manual, lecture material and notes, bibliography								14
Supplementary study in the library, online and in the field								3
Preparation for seminars/laboratory works, homework, reports, portfolios, essays								14
Tutoring								0
Exams and tests								3
Other activities								0

3.7	Total hours of individual study	34
3.8	Total hours per semester	104
3.9	Number of credit points	4

4. Pre-requisites (where appropriate)

4.1	Curriculum	N/A
4.2	Competence	Computer programming (C++), Data structures and algorithms, Linear Algebra, Numerical methods, Special mathematics.

5. Requirements (where appropriate)

5.1	For the course	Blackboard, video projector, computer
5.2	For the applications	Workstations, specific software (Visual Studio, Diblook)

6. Specific competences

Professional competences	**C6 -** Designing intelligent systems **C6.1 -** Describing the components of intelligent systems **C6.2 -** Using domain-specific tools for explaining and understanding the functioning of intelligent systems **C6.3 -** Applying the fundamental 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
Cross competences	N/A

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

7.1	General objective	Understanding the concepts related to digital images, computer vision and image processing. Learning and applying image processing methods, and designing specific applications.
7.2	Specific objectives	▪ Learning, evaluation and use of image processing specific concepts, algorithms and methods: digital image formats, camera model, statistical analysis, image filtering, image enhancing and restauration, segmentation, measurement. ▪ Acquiring the capacity of finding optimal solutions for image processing algorithm implementation, taking into consideration time and hardware constraints. ▪ Acquiring the capacity of quantitative and qualitative assessement of results, algorithms and systems for image processing. ▪ Learning the use of programming tools and image processing frameworks (Diblook, MS MFC, OPEN CV)

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Computer vision and its applications. Structure and functionality of computer vision systems. Image acquisition systems.	Interactive teaching, using oral presentations supported by multimedia tools, consultations, involving students in research and development activities.	N/A
2	Camera model, the image formation process, coordinate transforms, calibration.		
3	Fundamentals of stereovision, stereo configurations, depth computation, epipolar geometry.		
4	Binary image processing: Simple Geometric Properties		
5	Binary image processing: Labeling, Contour Tracing, Polygonal Approximation		
6	Binary image processing: Mathematical Morphology		
7	Grayscale image processing. Statistical properties. Image quality enhancement.		
8	Grayscale image processing: Convolution and Fourier Transform		
9	Grayscale image processing: Noise in images		
10	Grayscale image processing: Digital filtering.		
11	Grayscale image segmentation: Edge based segmentation		
12	Grayscale image processing: Advanced methods for edge extraction and linking.		
13	Grayscale image processing: Textures.		
14	Image region segmentation using intensity, color and texture features.		

Bibliography

1. R. C. Gonzales, R. E. Woods, "Digital Image Processing-Second Edition",3rd Edition, *Prentice Hall, 2008*
. 2. R. C. Gonzalez, R. E. Woods, S. L. Eddins, “Digital Image Processing Using MATLAB”, 2nd ed., *Gatesmark Publishing*, 2009.
3. E. Trucco, A. Verri, “Introductory Techniques for 3-D Computer Vision”, *Prentice Hall, 1998*.
4. G. X.Ritter, J.N. Wilson, “Handbook of computer vision algorithms în image algebra”, *CRC Press*, 2001.
5. S. Nedevschi, T. Marita, R. Danescu, F. Oniga, R. Brehar, I. Giosan, S. Bota, A. Ciurte, V. Andrei, Image

Processing – Laboratory Guide, *UTPRES*, Cluj-Napoca, 2016 ***Online*** 1. S. Nedevschi, “Prelucrarea imaginilor - Note de curs”, ftp.utcluj.ro/pub/users/nedevschi/IP_2016/			
8.2. Applications (Laboratory)		Teaching methods	Notes
1	Getting started with the DIBLook framework	Presentation using the blackboard and multimedia tools. Experiments and implementation using specific software tools (MS Visual Studio, Diblook) Evaluation of the design and implementation phases.	N/A
2	The color model. Color-grayscale and grayscale-black&white conversions		
3	The histogram of intensity levels		
4	Geometrical features of binary objects		
5	Binary objects labeling		
6	Border tracing algorithm.		
7	Morphological operations on binary images		
8	Statistical properties of grayscale images		
9	Image filtering in the spatial and frequency domains		
10	Noise modeling and digital image filtering		
11	Edge detection (1)		
12	Edge detection (2)		
13	Region-based image segmentation		
14	Evaluation		
8.2. Applications (Projects)			
1	Choosing and discussing the project subject (weeks 1 and 2).		
2	Discussing the literature study and the work schedule (weeks 3 and 4).		
3	Algorithm design (weeks 5 and 6)		
4	Presentation of algorithm implementation. Intermediary evaluation (weeks 7 and 8).		
5	Algorithm testing and validation. Quantitative and qualitative evaluation (weeks 9 and 10).		
6	Algorithm optimization (weeks 11 and 12).		
7	Final project assessment (weeks 13 and 14).		
Bibliography 1 S. Nedevschi, T. Marita, R. Danescu, F. Oniga, R. Brehar, I. Giosan, S. Bota, A. Ciurte, V. Andrei, “Image Processing – Laboratory Guide”, *UTPRES*, Cluj-Napoca, 2016 Online: http://users.utcluj.ro/~igiosan/teaching_ip.html			

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

The subject is part of the Computer Science and Information Technology curriculum, its contents combining fundamental and practical aspects used in the field of visual information processing (an ever growing domain). The subject content is correlated with the specific curricula of other Universities, in Romania and abroad, and is evaluated by government agencies (CNEAA and ARACIS). The subject’s activities are meant to make the students familiar with the applications and the research directions of the image processing field, helped by the internationally renowned experience of the teachers.

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		Testing the theoretical knowledge acquired, and the practical abilities of problem solving.		Written exam		50 %
Applications		Testing the practical abilities of designing and implementing solutions to specific problems. Attendance and activity.		Lab assessment, project assessment		50 %
10.4 Minimum standard of performance						
Modeling and implementation of solutions to specific engineering problems, using the domain’s formal apparatus.						

Course responsible
Prof. dr. ing. Sergiu Nedevschi

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	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	41.

2. Data about the subject

2.1	Subject name	Software Design									
2.2	Subject area	Computer Science and Information Technology									
2.3	Course responsible/lecturer	Prof.dr.eng. Mihaela Dinsoreanu – mihaela.dinsoreanu@cs.utcluj.ro									
2.4	Teachers in charge of applications	Prof.dr.eng. Mihaela Dinsoreanu									
2.5	Year of study	III	2.6	Semester	6	2.7	Assessment	exam	2.8	Subject category	DS/OB

3. Estimated total time

Sem.	Subject name	Lecture	Applications			Lecture	Applications			Individual study	TOTAL	Credit
		[hours / week.]				[hours / semester]						
			S	L	P		S	L	P			
6	**Software Design**	**2**	**-**	**2**	**1**	**28**	**-**	**28**	**14**	**34**	**104**	**4**

3.1	Number of hours per week	5	3.2	of which, course	2	3.3	applications	3
3.4	Total hours in the teaching plan	70	3.5	of which, course	28	3.6	applications	42
Individual study								Hours
Manual, lecture material and notes, bibliography								10
Supplementary study in the library, online and in the field								5
Preparation for seminars/laboratory works, homework, reports, portfolios, essays								10
Tutoring								4
Exams and tests								5
Other activities								

3.7	Total hours of individual study	34
3.8	Total hours per semester	104
3.9	Number of credit points	4

4. Pre-requisites (where appropriate)

4.1	Curriculum	Programming Techniques, Software Engineering
4.2	Competence	

5. Requirements (where appropriate)

5.1	For the course	Video projector (compulsory), internet connected computer (optional)
5.2	For the applications	16 internet connected computers

6. Specific competences

Professional competences	**C3** - Problem solving using specific Computer Science and Computer Engineering tools **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 software solutions for specific problems
Cross competences	N/A

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

7.1	General objective	Understand and model requirements, analyse and design appropriate solutions
7.2	Specific objectives	• Identify the most relevant functional and non-functional requirements of a software system and to document them • Design and motivate software architecture for (large scale) software systems • Recognize and apply major software architectural styles, design patterns, and frameworks • Describe a software architecture using various documentation approaches and architectural description languages • Generate architectural alternatives for a problem and select among them

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction and basic concepts review	Face-to-Face lecture, Powerpoint slides	
2	Architectural Styles (Structural)		
3	Architectural Styles (Distributed)		
4	Business logic architectural patterns		
5	Data Access and hybrid architectural patterns		
6	Presentation and Concurrency architectural patterns		
7	Midterm exam		
8	Applying Creational Design Patterns		
9	Applying Structural Design Patterns		
10	Applying Behavioral Design Patterns		
11	Class Design Principles (SOLID, GRASP)		
12	Package design Principles		
13	Service oriented architectures		
14	Software Design Quality metrics and final review		

Bibliography

Ian Gorton, Essential Software Architecture, Springer, second ed. 2011.

Taylor, R., Medvidovic, N., Dashofy, E., Software Architecture: Foundations, Theory, and Practice, 2010, Wiley.

Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, 3rd edition, 2013.

David Patterson, Armando Fox, Engineering Long-Lasting Software: An Agile Approach Using SaaS and Cloud Computing, Alpha Ed.

Buschmann, Frank, Regine Meunier, Hans Rohnert, Peter Sornmerlad, and Michael Stal. 2001. Pattern-oriented system architecture, volume 1: A system of patterns. Hoboken, NJ: John Wiley & Sons. [POSA book]

Fowler Martin, Patterns of Enterprise Application Architecture, Addison-Wesley Professional, 2002.

Course materials published at https://users.utcluj.ro/~dinso/PS2017

8.2. Applications (Seminars, Laboratory, Projects)		Teaching methods	Notes
1	Revision exercises (OOP, UML, testing techniques)	Face-to-Face tutoring,	
2	Database connections and operations		

3	Architectural styles exercises	additional materials	
4	Assignment 1 presentation and discussion		
5	Assignment 1 progress and discussion		
6	XML basics - exercises		
7	Design patterns exercises		
8	Assignment 2 presentation and discussion		
9	Assignment 2 progress and discussion		
10	Class design principles exercises		
11	Package design principles exercises		
12	Assignment 3 presentation and discussion		
13	Assignment 3 progress and discussion		
14	Assignments catch-up session		
Bibliography Course materials published at https://users.utcluj.ro/~dinso/PS2017 Java tutorial - docs.oracle.com/javase/tutorial/ C# tutorial – msdn.microsoft.com			

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

ACM Curriculum compliant course

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		Ability to understand requirements, analyse alternative solutions and design an appropriate solution		Written exam		60%
Applications		Analyse requirements and alternative solutions, design an appropriate solution and implement it in either java or C#.		Periodic presentations of the required deliverables		40%
10.4 Minimum standard of performance						
Grade of each lab assignment >= 5 Grade of each project deliverable >=5 Grade of the final exam >=5						

Course responsible
Prof.dr.eng. Mihaela Dinsoreanu

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	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	42.

2. Data about the subject

2.1	Subject name	Intelligent Systems
2.2	Subject area	Computer Science and Information Technology
2.3	Course responsible/lecturer	Prof. dr. eng. Leţia Ioan Alfred – Ioan.Alfred.Letia@cs.utcluj.ro
2.4	Teachers in charge of applications	Lect. dr. eng. Razvan Slăvescu – Razvan.Slavescu@cs.utcluj.ro Lect. dr. eng. Anca Marginean – Anca.Marginean@cs.utcluj.ro

2.5	Year of study	III	2.6	Semester	6	2.7	Assessment	exam	2.8	Subject category	DS/OB

3. Estimated total time

Sem.	Subject name	Lecture	Applications			Lecture	Applications			Individual study	TOTAL	Credit
		[hours / week.]				[hours / semester]						
			S	L	P		S	L	P			
6	**Intelligent Systems**	**2**	**-**	**2**	**-**	**28**	**-**	**28**	**-**	**48**	**104**	**4**

3.1	Number of hours per week	4	3.2	of which, course	2	3.3	applications	2
3.4	Total hours in the teaching plan	56	3.5	of which, course	28	3.6	applications	28

Individual study	Hours
Manual, lecture material and notes, bibliography	18
Supplementary study in the library, online and in the field	5
Preparation for seminars/laboratory works, homework, reports, portfolios, essays	10
Tutoring	6
Exams and tests	9
Other activities	0

3.7	Total hours of individual study	48
3.8	Total hours per semester	104
3.9	Number of credit points	4

4. Pre-requisites (where appropriate)

4.1	Curriculum	Logic Programming, Functional Programming
4.2	Competence	Fundamentals of Computer Programming

5. Requirements (where appropriate)

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

6. Specific competences

Professional competences	**C6 –** Design of intelligent systems (4 credits) **C6.1 –** Describing the components of intelligent systems **C6.2 –** Usage of specific instruments of the domain for explaining and understanding the functioning of intelligent systems **C6.3 –** Application of principles and basic methods for the specification of solutions typical problems using intelligent systems **C6.4 –** Choosing criteria and methods for the evaluation of quality, performance and limits of intelligent systems **C6.5 –** Development and implementation of professional designs for intelligent systems
Cross competences	N/A

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

7.1	General objective	Knowledge of representation and reasoning of fundamental problems of intelligent systems
7.2	Specific objectives	Fundamental methods for basic representations in intelligent systems: uncertainty, learning, communication

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction.	Slides, Algorithms, Quality of solutions, Exceptions, Limits in the representation of the real world	
2	Uncertainty: inference using full joint distributions, Bayes' rule and its use.		
3	Probabilistic Reasoning: semantics of Bayesian networks, efficient representation, exact inference, approximate.		
4	Probabilistic Reasoning over Time: hidden Markov models, dynamic Bayesian networks.		
5	Making Simple Decisions: utility functions, decision networks, value of information.		
6	Making Complex Decisions: value iteration, policy iteration, partially observable MDPs, game theory.		
7	Learning from Observations: learning decision trees, ensemble learning.		
8	Knowledge in Learning: explanation-based, relevance information, inductive logic programming.		
9	Statistical Learning Methods: hidden variables, instance-based, neural networks, kernel machines.		
10	Reinforcement Learning.		
11	Association analysis: frequent itemset generation, rule generation, compact representation of frequent itemsets, alternative methods of generating frequent itemsets, FP-growth algorithm.		
12	Communication: syntactic analysis, semantic interpretation.		
13	Perception, representation and action in multi-agent systems.		
14	Overview on Intelligent Systems: Present and Future.		
Bibliography 1. Artificial Intelligence: A Modern Approach: Russell, Norvig, Prentice Hall, 2002 2. Tan, Steinbach, Kumar: Data Mining: Association Analysis, 2004			
8.2. Applications (Seminars, Laboratory, Projects)		Teaching methods	Notes
1	Introduction to the documentation for the assignment	Platform, Documentation, Testing, Examples, New examples	
2	Studying the documentation for the assignment		
3	Studying the design of the tool		
4	Practicing the exercises provided in the archive		
5	Understanding the main parts of the software		
6	Running the system by tracing at high level		
7	Mastering the running of the system and the examples provided		
8	Conceptual design of new examples		

9	Code for the new examples		
10	Testing and debugging the new cases		
11	Measuring the performance of the system		
12	Documenting the new scenarios		
13	Comparison of the differences between the cases developed and those provided		
14	Final evaluation of the exercises developed		
Bibliography 1. Various Intelligent Systems Tools from the WWW.			

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

The course outline represents the most known and used one in the world methods for intelligent systems, continuously assessed by the research community in the world regarding its influence and use in software technology.

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		Problems and theoretical concepts		Written exam		80%
Applications		Usage of specific tools on the examples developed and tested by the students		Evaluation in the laboratory		20%
10.4 Minimum standard of performance						
Representation of knowledge and its use in solving specific intelligent systems problems using specific tools						

Course responsible
Prof. dr. eng. Leţia Ioan Alfred

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	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	43.

2. Data about the subject

2.1	Subject name					Practical Placement					
2.2	Subject area					Computer Science and Information Technology					
2.3	Course responsible/lecturer					Assoc. prof. dr. eng. Tiberiu Marita					
2.4	Teachers in charge of applications					Internship supervisors appointed by the faculty					
2.5	Year of study	III	2.6	Semester	6	2.7	Assessment	Verification	2.8	Subject category	DID/OB

3. Estimated total time

Sem.	Subject name	Lecture	Applications			Lecture	Applications			Individual study	TOTAL	Credit
		[hours / week.]				[hours / semester]						
			S	L	P		S	L	P			
6	**Practical Placement**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**240**	**240**	**8**

3.1	Number of hours per week	-	3.2	of which, course	-	3.3	applications	-
3.4	Total hours in the teaching plan	-	3.5	of which, course	-	3.6	applications	-

Individual study	Hours
Manual, lecture material and notes, bibliography	0
Supplementary study in the library, online and in the field	40
Preparation for seminars/laboratory works, homework, reports, portfolios, essays	10
Tutoring	18
Exams and tests	2
Other activities	170

3.7	Total hours of individual study	240
3.8	Total hours per semester	240
3.9	Number of credit points	8

4. Pre-requisites (where appropriate)

4.1	Curriculum	N/A
4.2	Competence	N/A

5. Requirements (where appropriate)

5.1	For the course	N/A
5.2	For the applications	N/A

6. Specific competences

Professional competences	**C2** Designing hardware, software and communication components (2 credits) **C2.1** Describing the structure and functioning of computational, communication and software components and systems **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 **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 **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 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** Creating a project including the problem's identification and analysis, its design and development, also proving an understanding of the basic quality requirements
Cross competenc	**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 results from the activity field. (2 credits)

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

7.1	General objective	Application of fundamental and applied knowledge gained in the projects development within a specialized company or research team (theme set by the project manager)
7.2	Specific objectives	Acquaintance and student involvement in every development stage of a hardware / software / communication project and connected aspects of design activities: - Analysis and documentation - Study and acquaintance with specific design tools - Design, implementation, testing and validation of the project - Preparation of documentations, technical reports - Team work and communication skills - Project management activities

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Not applicable	N/A	
Bibliography			
8.2. Applications		Teaching methods	Notes
1	• Study / documentation / training / • analysis of the potential methodologies and/or technologies • preparation of the project specifications • implementation and deployment of the hardware or software system • product testing and validation	N/A	

	- product documenting		

Bibliography

For the project development, the draft bibliography is the one recommended by the project leader from the company or by the research team at which the implementation is performed and the one resulted in the documenting phase.

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

This discipline provides education and training of the students at the workplace site, with benefits for both sides. Students are familiarized with the working and professional requirements needed to work in a company, and companies have the opportunity to shape students to facilitate their employment after graduation (to reduce training expenses / training). Also it aims to increase cohesion between academia and employment in a priority area in terms of national and European level in order to improve the skills of employees and to prepare and maintain them in the labor market in a particularly dynamic and competitive domain (mainly existing competition with Eastern European countries and Asia - India and China).

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		N/A		N/A		N/A
Applications		Attendance (min 240 h), activity, tutor assessment		Colloquy		100%
10.4 Minimum standard of performance						
Development of a hardware / software / communication engineering project						

Course responsible
Assoc.prof.dr.eng. Tiberiu Marita

Head of department
Prof.dr.eng. Rodica Potolea