SYLLABUS

1. Data about the program of study

1.1	Institution	Technical University of Cluj-Napoca
1.2	Faculty	Automation and Computer Science
1.3	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	44.

2. Data about the subject

2.1	Subject name	Computer Networks
2.2	Subject area	Computer Science and Information Technology
2.3	Course responsible/lecturer	Prof. dr. eng. Vasile Dădârlat – vasile.dadarlat@cs.utcluj.ro
2.4	Teachers in charge of applications	Assoc.prof. dr. eng. Peculea Adrian – Adrian.Peculea@cs.utcluj.ro Lect. dr. eng. Iancu Bogdan – Bogdan.Iancu@cs.utcluj.ro

2.5	Year of study	IV	2.6	Semester	7	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			
7	**Computer Networks**	**2**	**-**	**2**	**-**	**28**	**-**	**28**	**-**	**74**	**130**	**5**

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								44
Supplementary study in the library, online and in the field								10
Preparation for seminars/laboratory works, homework, reports, portfolios, essays								20
Tutoring								
Exams and tests								
Other activities								

3.7	Total hours of individual study	74
3.8	Total hours per semester	130
3.9	Number of credit points	5

4. Pre-requisites (where appropriate)

4.1	Curriculum	
4.2	Competence	Basic knowledge in programming languages (C, Java) Computer architecture, Operating systems

5. Requirements (where appropriate)

5.1	For the course	N/A
5.2	For the applications	Classroom, PC with internet access

6. Specific competences

Professional competences	**C2:** Designing hardware, software and communication components **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	Teamwork, working with partial and contradicting specifications
7.2	Specific objectives	Each student able to design LAN's software & hardware architecture

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction. Concepts, network types, characteristics, evolution, standards	Oral Presentations using multimedia means Q & A Interactive teaching	
2	ISO-OSI Reference model and Internet's TCP/IP protocol stack. OSI abstract model presentation, description of protocol functions for every layer. General presentation for TCP/IP protocol stack		
3	Data transmission techniques. Data transmission concepts, analog and digital transmission techniques, coding, communication channels		
4	Types of computer networks. Architectures, evolution, topologies, physical parameters		
5	Physical level. Transmission media, characteristics, performances, connectors, structured cabling system		
6	Medium access control. Medium access techniques for local (wired and wireless) and wide area networks		
7	Data Link level. Functions, problems, protocols, case study: HDLC		
8	Local Area Computer Networks. Fundamentals, architectures, evolution		
9	Local Area Computer Networks. Systems, performances		
10	Computer Networks Interconnection. Devices for network interconnection; presentation of bridges, switches and routers		
11	Internet access. IP (+ ICMP), IPv6 (+IGMP) protocols. Address resolution protocol. Routing protocols		
12	Transport level protocols. TCP protocol; congestion control. TCP and UDP sockets		
13	General introduction to Internet applications. File transfer. Electronic mail, multimedia transmissions, network management		
14	General introduction to Internet applications. Security issues		

Bibliography

1. V.Dadarlat, E.Cebuc - Retele Locale de Calculatoare - de la cablare la interconectare, Editura Albastra (Microinformatica), Cluj, 2006, ISBN 973-650-161-2

2. W. Stallings, *Data and Computer Communications*; Prentice Hall , 2005

3. A. S. Tanenbaum, *Reţele de Calcultoare*; Agora Press,2004

8.2. Applications (Laboratory)		Teaching methods	Notes
1	Lab presentation; Elements of the structured cabling system	Practical exercises Brief presentation of	
2	Network connection techniques		
3	Spanning tree protocol		
4	Copper based media and cabling with UTP		

5	Medium access methods	possible solutions Self testing programmes	
6	Flow control protocols		
7	Protocol Inspector		
8	Optical Fiber and components		
9	Wireless access		
10	IP Addressing		
11	Network Inspector		
12	Network programming using sockets I		
13	Network programming using sockets II		
14	Lab exam		
Bibliography 1. Notes & lab notes available at: ftp.utcluj.ro			

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

Course content is kept state of the art by using latest protocols and devices available on the market

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		Interactivity and initial preparation, intermediary and final written examinations		Written exam (2,5 h).		70%
Applications		Quality of practical work, participation		Continuous assessment, final written colloquium		30%
10.4 Minimum standard of performance						
Grades > 5 for both theoretical and practical assessments						

Course responsible
Prof. dr. eng. Vasile Dădârlat

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	45.

2. Data about the subject

2.1	Subject name	Distributed Systems
2.2	Subject area	Computer Science and Information Technology
2.3	Course responsible/lecturer	Prof. dr. eng. Ioan Salomie – Ioan.Salomie@cs.utcluj.ro
2.4	Teachers in charge of applications	Sl.Dr.Eng. Tudor Cioara, Sl.Dr. Eng. Ionut Anghel, As. Drd. Marcel Antal, As. Drd. Claudia Pop, As. Drd. Dorin Moldovan

2.5	Year of study	IV	2.6	Semester	7	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			
7	**Distributed Systems**	**2**	**-**	**2**	**1**	**28**	**-**	**28**	**14**	**60**	**130**	**5**

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	18
Supplementary study in the library, online and in the field	6
Preparation for seminars/laboratory works, homework, reports, portfolios, essays	24
Tutoring	
Exams and tests	12
Other activities	18

3.7	Total hours of individual study	60
3.8	Total hours per semester	130
3.9	Number of credit points	5

4. Pre-requisites (where appropriate)

4.1	Curriculum	Computer networks
4.2	Competence	Ability to analyze and design a local network using simulators available

5. Requirements (where appropriate)

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

6. Specific competences

Professional competences	**C4 -** Improving the performances of the hardware, software and communication systems (2 credits) **C4.1** - Identifying and describing the defining elements of the performances of the hardware, software and communication systems **C4.2** - Explaining the interaction of the factors that determine the performances of the hardware, software and communication systems **C4.3 -** Applying the fundamental methods and principles for increasing the performances of the hardware, software and communication systems **C4.4** - Choosing the criteria and evaluation methods of the performances of the hardware, software and communication systems **C4.5** - Developing professional solutions for hardware, software and communication systems based on performance optimization **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 the computing system to the specific requirements of the application field **C5.3** - Using fundamental principles and methods for ensuring the security, the safety and ease of exploitation of the computing systems **C5.4** - Proper utilization of the quality, safety and security standards in the field of 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 **C6 -** Designing intelligent systems (1 credit) **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 the criteria and evaluation methods for the quality, performances and limitations of intelligent 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	Capacity to analyse, develop and implement distributed software systems
7.2	Specific objectives	-Capacity of designing distributed systems at both architectural and components' level by using the main concepts and paradigms of the domain as well as the capacity of understanding the relationships of the domain with other computer science areas. -Capacity of identifying the main models, techniques and technologies that could be used in the design of distributed systems by considering a set of functional and non-functional specifications and constraints -Capacity of developing and using service based technologies for designing distributed systems -Capacity of using Java and .NET, Spring, Angular JS and other Web technologies for designing distributed systems

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction – Characterization of Distributed Systems	-Using modern multimedia	
2	Distributed System Models		

3	Non-Functional Requirements	teaching methods and direct access to internet; -Challenging questions during lecturers -Students are invited to collaborate in research projects -Personal assistance hours during the semester and before the exam	
4	Inter-process Communication		
5	Indirect Communication		
6	Remote Procedure Call (RPC)		
7	Distributed Computation Model		
8	Time and Causality in Distributed Systems		
9	Global States and Snapshots		
10	Distributed Data Processing – Concepts , Reference Architectures		
11	Distributed Data Processing – Data Distribution Techniques		
12	Distributed Transactions		
13	Distributed Concurrency Control		
14	Overview of Google Distributed Technologies		

Bibliography

1. 1. G. Coulouris, J.Dollimore, T.Kindberg – Distributed Systems. Concepts and Design, Addison Wesley, 2005
2. A. Tanenbaum, M. van Steen – Distributed Systems, Prentice Hall, 2002
3. A.D. Kshemkalyan M.Singhal - Distributed Computing, Cambridge Press 2008
4. Ioan Salomie, Tudor Cioara - Lecture Notes, Lab Notes Project Notes and Assignments http://www.coned.utcluj.ro/~salomie/DS_Lic

8.2. Applications (Laboratory, Projects)		Teaching methods	Notes
1	Intro to Lab Resources	-Pre-defined exercises and assignments -Using modern multimedia teaching methods and direct access to internet; -Students are invited to collaborate in research projects -Personal assistance hours during the semester and before the exam	
2	A1. Request-Reply Communication Paradigm		
3	P1. Spring MVC Framework		
4	A1.1 The Basics (examples and hands on)		
5	A1.2 Web application using Request - Reply		
6	A2. Remote Procedure Call (RPC)		
7	A2.1 The Basics (examples and hands on)		
8	A2.2 RPC application using distributed objects (Java RMI or .NET)		
9	P2. Angular JS		
10	A3. Asynchronous communication using messaging		
11	A3.1 The Basics (examples and hands on)		
12	A3.2. Asynchronous DS application using Java or .Net messaging frameworks		
13	A4. Service Oriented Distributed Systems – Application design using 3 services (SOA-Java, SOA-.Net and REST (Java or .Net))		
14	Evaluation		

Bibliography

1. Ioan Salomie, Tudor Cioara, Ionut Anghel, Tudor Salomie – Distributed Computing and Systems – A practical Approach, Albastra Publ. House, 2008
2. Ioan Salomie, Tudor Cioara - Lecture Notes, Lab Notes Project Notes and Assignments http://www.coned.utcluj.ro/~salomie/DS_Lic

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

Distributed Systems is a subject of the domain "Computers and Information Technology".

It teaches students about the development and implementing of distributed software systems. The content was developed based on the analysis of similar disciplines from other universities as well as based on the requirements of the IT employees. The content was also evaluated by Romanian governmental agencies CNEAA and ARACIS.

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		The level of assimilation of the knowledge about distributed systems, teacher during		Written Exam		50%

		the course				
Applications		-Capacity of designing distributed systems at both architectural and components’ level by using the main concepts and paradigms of the domain as well as the capacity of understanding the relationships of the domain with other computer science areas. -Capacity of identifying the main models, techniques and technologies that could be used in the design of distributed systems by considering a set of functional and non-functional specifications and constraints -Individual activity during course, lab and project -Presence		Assignments evaluation, Project evaluation		30% 20%
10.4 Minimum standard of performance						
-To be able to design and implement distributed software systems -At least mark 5 at the exam, lab and project evaluation						

Course responsible
Prof. dr. eng. Ioan Salomie

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	46.10

2. Data about the subject

<table>
<tr><td>2.1</td><td colspan="5">Subject name</td><td colspan="6">Input/Output Systems and Peripheral Devices</td></tr>
<tr><td>2.2</td><td colspan="5">Subject area</td><td colspan="6">Computer Science and Information Technology</td></tr>
<tr><td>2.3</td><td colspan="5">Course responsible/lecturer</td><td colspan="6">Prof. Dr. Eng. Zoltan Francisc Baruch – Zoltan.Baruch@cs.utcluj.ro</td></tr>
<tr><td>2.4</td><td colspan="5">Teachers in charge of applications</td><td colspan="6">Prof. Dr. Eng. Zoltan Francisc Baruch – Zoltan.Baruch@cs.utcluj.ro
Eng. Mihai Grigorescu – mihaigrigorescu13@gmail.com</td></tr>
<tr><td>2.5</td><td>Year of study</td><td>IV</td><td>2.6</td><td>Semester</td><td>7</td><td>2.7</td><td>Assessment</td><td>exam</td><td>2.8</td><td>Subject category</td><td>DS/OP</td></tr>
</table>

3. Estimated total time

<table>
<tr><td rowspan="3">Sem.</td><td rowspan="3">Subject name</td><td>Lecture</td><td colspan="3">Applications</td><td>Lecture</td><td colspan="3">Applications</td><td>Individual study</td><td>TOTAL</td><td>Credit</td></tr>
<tr><td colspan="4">[hours / week.]</td><td colspan="5">[hours / semester]</td><td></td><td></td></tr>
<tr><td></td><td>S</td><td>L</td><td>P</td><td></td><td>S</td><td>L</td><td>P</td><td></td><td></td><td></td></tr>
<tr><td><b>7</b></td><td><b>Input/Output Systems and Peripheral Devices</b></td><td><b>2</b></td><td><b>-</b></td><td><b>2</b></td><td><b>-</b></td><td><b>28</b></td><td><b>-</b></td><td><b>28</b></td><td><b>-</b></td><td><b>74</b></td><td><b>130</b></td><td><b>5</b></td></tr>
</table>

<table>
<tr><td>3.1</td><td>Number of hours per week</td><td>4</td><td>3.2</td><td>of which, course</td><td>2</td><td>3.3</td><td>applications</td><td>2</td></tr>
<tr><td>3.4</td><td>Total hours in the teaching plan</td><td>130</td><td>3.5</td><td>of which, course</td><td>28</td><td>3.6</td><td>applications</td><td>28</td></tr>
<tr><td colspan="8">Individual study</td><td>Hours</td></tr>
<tr><td colspan="8">Manual, lecture material and notes, bibliography</td><td>34</td></tr>
<tr><td colspan="8">Supplementary study in the library, online and in the field</td><td>12</td></tr>
<tr><td colspan="8">Preparation for seminars/laboratory works, homework, reports, portfolios, essays</td><td>18</td></tr>
<tr><td colspan="8">Tutoring</td><td>5</td></tr>
<tr><td colspan="8">Exams and tests</td><td>5</td></tr>
<tr><td colspan="8">Other activities</td><td>0</td></tr>
<tr><td>3.7</td><td colspan="3">Total hours of individual study</td><td>130</td></tr>
<tr><td>3.8</td><td colspan="3">Total hours per semester</td><td>74</td></tr>
<tr><td>3.9</td><td colspan="3">Number of credit points</td><td>5</td></tr>
</table>

4. Pre-requisites (where appropriate)

4.1	Curriculum	Computer Programming, Computer Architecture
4.2	Competence	Competences of disciplines Computer Programming and Computer Architecture

5. Requirements (where appropriate)

5.1	For the course	Projector, computer
5.2	For the applications	Computers, the Microsoft Visual Studio programming environment

6. Specific competences

Professional competences	**C4** – Improving the performances of the hardware, software and communication systems (2 credits) **C4.1** – Identifying and describing the defining elements of the performances of the hardware, software and communication systems **C4.2** – Explaining the interaction of the factors that determine the performances of the hardware, software and communication systems **C4.3** – Applying the fundamental methods and principles for increasing the performances of the hardware, software and communication systems **C4.4** – Choosing the criteria and evaluation methods of the performances of the hardware, software and communication systems **C4.5 -** Developing 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 (3 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 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
Cross competences	N/A

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

7.1	General objective	Knowledge of operation and performance parameters for input/output interfaces and peripheral devices; ability to communicate with controllers of peripheral devices
7.2	Specific objectives	- Using basic methods and principles for enhancing performance of computer systems - Designing input/output interfaces for connecting various devices to the computer - Designing and implementing in software input/output protocols - Writing system programs for controlling input/output interfaces

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction. Programmed I/O	- PowerPoint presentations - Questions, discussions	N/A
2	Interrupt-Driven I/O. Direct Memory Access. I/O Processors		
3	Buses. Electrical Considerations. Synchronous and Asynchronous Buses. Bus Arbitration. VME Bus		
4	Local Buses. PCI Bus. PCI-X Bus. PCI Express Bus		
5	PCI Bus Variants for Personal Computers. PCI Bus Variants for Industrial Systems		
6	Serial Buses: I^{2}C; SPI; USB		
7	Mid-Term Exam		
8	Liquid Crystal Displays. Liquid Crystals. TN Technology. Addressing Methods. Backlighting		
9	Liquid Crystal Displays (cont.). Characteristics. VA Technology. IPS Technology		
10	Plasma Displays. Field Emission Displays. Organic LED Displays		
11	Graphics Adapters. Structure of a Graphics Adapter. Color		

	Representation. Video Memory. Graphics Accelerators. 3D Accelerators		
12	Graphics Processing Units. Digital Interfaces for Monitors: DVI; HDMI; DisplayPort		
13	Optical Discs. Physical Medium. Data Organization and Encoding. The CD-ROM Drive. Types of Compact Discs		
14	DVD Discs. Blu-Ray Discs		
Bibliography 1. Baruch, Z. F., Computer Input/Output Systems (in Romanian), Cartea Albastră, Cluj-Napoca, 2000, ISBN 973-9443-39-7. 2. Rosch, Winn L., Hardware Bible, Sixth Edition, Que Publishing, 2003, ISBN 0-7897-2859-1.			
8.2. Applications (Laboratory)		Teaching methods	Notes
1	The Serial Port (I)	- Additional explanations - Using a programming environment for the C language	N/A
2	The Serial Port (II)		
3	The PCI Express Bus (I)		
4	The PCI Express Bus (II)		
5	The System Management Bus (I)		
6	The System Management Bus (II)		
7	The Universal Serial Bus (I)		
8	The Universal Serial Bus (II)		
9	Printers		
10	The SCSI Interface		
11	The ATA Interface (I)		
12	The ATA Interface (II)		
13	Compact Discs. The ATAPI Interface		
14	Laboratory Colloquy		
Bibliography 1. Lecture slides and laboratory works at http://users.utcluj.ro/~baruch/en/pages/teaching/inputoutput-systems.php			

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

The contents of the discipline has been corroborated with the contents of similar disciplines in the USA and Europe, as well as with chapters related to input/output systems of acknowledged manuals used in prestigious universities. The discipline has been evaluated by the ARACIS agency.

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		Understanding theoretical concepts of input/output systems and the principle of operation for peripheral devices		Written exam		70%
Applications		Ability to write communication programs with controllers of peripheral devices		Written evaluation		30%
10.4 Minimum standard of performance						
Finishing at least one application in each laboratory session Grade > 5 for the written exam; Grade > 5 for the laboratory written evaluation						

Course responsible
Prof. Dr. Eng. Zoltan Baruch

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	46.2

2. Data about the subject

2.1	Subject name					Parallel and Distributed Computing					
2.2	Subject area					Computer Science and Information Technology					
2.3	Course responsible/lecturer					S.l.dr.ing. Anca Hangan – Anca.Hangan@cs.utcluj.ro					
2.4	Teachers in charge of applications					S.l.dr.ing. Anca Hangan – Anca.Hangan@cs.utcluj.ro					
2.5	Year of study	IV	2.6	Semester	7	2.7	Assessment	exam	2.8	Subject category	DS/OP

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			
7	**Parallel and Distributed Computing**	**2**	**-**	**2**	**-**	**28**	**-**	**28**	**-**	**74**	**130**	**5**

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								28
Supplementary study in the library, online and in the field								14
Preparation for seminars/laboratory works, homework, reports, portfolios, essays								24
Tutoring								4
Exams and tests								4
Other activities								N/A

3.7	Total hours of individual study	74
3.8	Total hours per semester	130
3.9	Number of credit points	5

4. Pre-requisites (where appropriate)

4.1	Curriculum	Fundamental Algorithms, Fundamental programming techniques
4.2	Competence	

5. Requirements (where appropriate)

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

6. Specific competences

Professional competences	**C4** Improving the performances of the hardware, software and communication systems (2 credits) **C4.1** Identifying and describing the defining elements of the performances of the hardware, software and communication systems **C4.2** Explaining the interaction of the factors that determine the performances of 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 professional solutions for hardware, software and communication systems based on performance optimization **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 the computing system to the specifc requirements of the application field **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 competences	N/A

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

7.1	General objective	1. Students become aware of differences and similarities between parallel and distributed computing so the students understand the boundaries of both domains. 2. Students become familiar with the principles of designing parallel programs. 3. Students become familiar with the main classes of distributed algorithms.
7.2	Specific objectives	Parallel algorithms performance and scalability. Parallel algorithms design. Distributed algorithms: time synchronization, distributed mutual exclusion, causal ordering, leader election and snapshots.

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction: goal, administrative issues, definition of parallel system and distributed systems.	Interactive lectures using PPT presentations, exercises (at whiteboard) and questions addressed to the students.	
2	Performance and scalability: metrics, scalability definition, Amdahl's law.		
3	Parallel algorithm design: parallelization process, data dependency.		
4	Parallel algorithm design: case study: ocean simulation.		
5	Parallel algorithm design: decomposition techniques, mapping techniques for load balancing.		
6	Interconnection networks: static interconnection networks (metrics, topologies), dynamic interconnection networks (buses, crossbars, multistage networks).		
7	Dense matrix algorithms: matrix-vector multiplication (1D partitioning and 2D partitioning, comparison 1D to 2D), matrix-matrix multiplication (2D partitioning, Cannon algorithm).		
8	Time: physical clocks synchronization (Cristian algorithm, Berkeley algorithm, Network Time Protocol), logical clocks (Scalar time, Vector time, efficient implementation of vector clocks - Singhal-Kshemkalyani).		
9	Distributed mutual exclusion: problem definition, Token-ring, Suzuki-		

	Kasami, central coordinator, Lamport, Ricart-Agrawala.		
10	Causal ordering: problem definition, Birman-Schiper-Stephenson, Schiper-Eggli-Sandoz.		
11	Leader election: problem definition, general networks (FloodMax, OptFloodMax), synchronous / asynchronous ring (LeLann, Chang-Roberts, Hirschberg-Sinclair).		
12	Leader election: synchronous / asynchronous ring (Franklin, Peterson), anonymous ring (Itai-Rodeh).		
13	Snapshot: problem definition, Chandy-Lamport, Spezialetti-Kearns, Lai-Yang.		
14	Data analysis with Hadoop Discussion on parallel vs. distributed vs. concurrent.		

1. *Parallel and Distributed Computing - Lecture notes* **-** C. Ivan ,http://ftp.utcluj.ro/pub/users/civan/PDC
2. *Concepte, mecanisme și soluții middleware pentru dezvoltarea sistemelor distribuite*, C.Ivan, Ed.Risoprint (CNCSIS), ISBN 987-973-53-1776-8, 2016
3. *Arhitecturi paralele de calcul*, C. Ivan , Editura Roprint (CNCSIS) ,ISBN 973-354-23-4-1, 2001
4. *Parallel Programming for Multicore and cluster systems* ,Rauber T, Runger. G, Springer Verlag
5. ISBN 978-3-642-04817-3,2010
6. *Introduction to Distributed Systems* -Concepts and design. George Coulouris, Jean Dollimore and Tim Kindberg, Prentice Hall, ISBN 0201-619-180, 2005 si editia revizuită 2008
7. *Distributed computing : principles, algorithms and systems,* M. Singhal, A Kshemkalyani,Cambridge Unive ISBN-13 978-0521876346 , 2008

8.2. Applications (Laboratory)		Teaching methods	Notes
1	Modern computing models. Concurent vs. parallel vs distributed.	Problem based approach.	
2	Multithreading and concurrency in Java.		
3	Shared memory parallel programming.(OpenMP).Fundamental parallel algorithms implementation and analysis. (serch, graph, matrix)		
4	Shared memory parallel programming.(OpenMP).Fundamental parallel algorithms implementation and analysis.		
5	Message passing parallel programming (MPI). Fundamental parallel algorithms implementation and analysis.(serach, graph, matrix)		
6	Message passing parallel programming (MPI). Fundamental parallel algorithms implementation and analysis.		
7	Other parallel programming libraries (Pj2, PP –Net)		
8	Distributed programming – Sockets		
9	Distributed programming - Remote methods invocation		
10	Distributed algorithms- Logical clocks, leader election, distributed mutual exclusion.		
11	Group communication		
12	P2P systems		
13	Parallel processing in distributed context. Models, frameworks		
14	Project presentation/Laboratory test		

Bibliography

1. *Parallel and Distributed Computing – Practical activities***-** C. Ivan ,http://ftp.utcluj.ro/pub/users/civan/PDC
2. *Calcul paralel şi distribuit* - Lucrari practice, C. Ivan ,Editura UTPress , ISBN ISBN 978-973-662-283-0, 2007
3. *Introduction to Parallel Computing*, V.Kumar, A. Grama, A. Gupta, G. Karypis, Benjamin-Cummings,IS 2,2008
4. *Programming on parallel machines* - GPU, multicore and clusters,N. Mathloff, Universityof California Davis, 2016 , http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf

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

As Cluj software workforce market gets more sophisticated, having solid knowledge of how to develop parallel programs and mastering the distributed computing are qualities that software companies look for.

10. Evaluation

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Weight in the final grade
Course	Formal assessment to test theoretical knowledge and problem solving skills. Attendance and activity.	Written exam.	40%
Applications	Formal assessment to test practical skills for designing parallel and distributed solutions and implementation . Attendance and activity.	Project, Research report and Homeworks during term.	60%
10.4 Minimum standard of performance			
Design and implementation of parallel/distributed solutions using the theoretical models and tools (OpenMP, MPI, PP.Net, Sockets,RMI). Pre-requisite for written exam: 6 mandatory lecture attendances.			

Course responsible
S.l.dr.ing. Anca Hangan

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	47.1

2. Data about the subject

2.1	Subject name	Operating Systems Design
2.2	Subject area	Computer Science and Information Technology
2.3	Course responsible/lecturer	Assoc.prof. dr. eng. Adrian Coleşa – adrian.colesa@cs.utcluj.ro
2.4	Teachers in charge of applications	Assoc.prof. dr. eng. Adrian Coleşa – adrian.colesa@cs.utcluj.ro Eng. Radu Ciocas – rciocas@bitdefender.com Eng. Alexandru Gurzou – agurzou@bitdefender.com Eng. Radu Portase – rportase@bitdefender.com Eng. Alexandru Brînduşe – abrinduse@bitdefender.com

2.5	Year of study	IV	2.6	Semester	7	2.7	Assessment	exam	2.8	Subject category	DS/OP

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			
7	**Operating Systems Design**	**2**	**-**	**2**	**1**	**28**	**-**	**28**	**14**	**85**	**155**	**6**

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								40
Supplementary study in the library, online and in the field								0
Preparation for seminars/laboratory works, homework, reports, portfolios, essays								42
Tutoring								1
Exams and tests								2
Other activities								0
3.7	Total hours of individual study	85						
3.8	Total hours per semester	155						
3.9	Number of credit points	6						

4. Pre-requisites (where appropriate)

4.1	Curriculum	Operating Systems
4.2	Competence	C programming; Define and use basic OS concepts and system calls

5. Requirements (where appropriate)

5.1	For the course	Blackboard / Whiteboard, Beamer
5.2	For the applications	64-bit Computers with hardware virtualization support, 64-bit Linux and Windows, VMware Workstation, Blackboard / Whiteboard

6. Specific competences

Professional competences	**C5**: Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems **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 the computing system to the specific requirements of the application field **C5.3**: Using fundamental principles and methods for ensuring the security, the safety and ease of exploitation of the computing systems **C5.4**: Proper utilization of the quality, safety and security standards in the field of 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 competences	N/A

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

7.1	General objective	Provide the students a clear understanding of an OS' internal structure, its main components' role and functionality, and the fundamental OS design and implementation strategies.
7.2	Specific objectives	Let the students: 1. Know and understand the possible OS internal structures. 2. Know and understand the possible design and implementation alternatives of the main OS components, like the scheduler, process and thread manager, memory manager etc. 3. Be able to analyze a specific OS design problems and find solutions to them. 4. Be able to implement in C or assembly different OS components and system calls. 5. Be able to work in team and manage relatively complex software projects.

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	**General structure of an OS**. Possible OS structures (monolithic, layered, micro-kernel, virtual machine, exokernel), its components, their functionality, role, interconnectivity.	(1) use beamer slides, combined with blackboard illustration; (2) interactions with students: ask their opinion relative to the presented subject; (3) give each class a short evaluation test; let students discuss and argue each other their solution; give them the good solution and let them evaluate their own one; (4) propose 2-3 interesting study	
2	**Process and thread management (1)**. Scheduling algorithms. FCFS, SJF, Priority-based, Lottery. Priority inversion.		
3	**Process and thread management (2)**. Scheduling algorithms: RR, MLFQ. Use cases: Solaris, Windows and Linux scheduling policies.		
4	**Synchronization mechanisms (1)**. General Design Principles. Hardware mechanisms used for implementation of higher-level synchronization mechanisms. Design and implementation of locks, semaphores, condition variables. Deadlock avoidance.		
5	**Synchronization mechanisms (2)**. Linux and Windows Use Cases. The synchronization mechanisms provided by Linux and Windows. The way they are implemented.		
6	**Synchronization mechanisms (3)**. Deadlock. Deadlock avoidance, prevention and detection algorithms.		
7	**Process management (1)**. Definition of the process concept, system call mechanism and possible implementations, handle (file descriptor) management, basic system calls for process management.		
8	**Process management (2)**. Process memory address space structure, argument passing on the stack, process creation strategies, multi-threading support.		
9	**Memory management (1)**. General aspects, design and implementation alternatives of different memory management techniques and		

	mechanisms: contiguous allocation, segmentation, and paging.	cases of OSes to be prepared and presented by students; (5) students are invited to collaborate in research projects.	
10	**Memory management (2)**. Paging specific problems like page table hierarchical structure, memory sharing, page tables for Intel architecture.		
11	**Memory management (3)**. Virtual memory's design and implementation aspects: swapping and lazy loading. Page replacement algorithms.		
12	**File systems (1)**. General Design Aspects. Design and implementation alternatives of file systems concepts (files, directories), storage space management. Advantages and disadvantages.		
13	**File systems (2)**. Linux and Windows File Systems. Design and implementation of Ext2 and NTFS.		
14	**Security aspect. Subject review.** Basic security aspects design. Overview of all presented subjects.		

Bibliography

1. A. Silberschatz, G. Gagne, P. B. Galvin, *Operating Systems Concepts*, 7th edition, Wiley, 2005, ISBN 978-0-471-69466-3
2. A. Tanenbaum, A. Woodhull. *Operating Systems Design and Implementation*. 3rd edition, Prentice Hall, 2006, ISBN: 0131429388
3. Daniel Pierre Bovet, *Understanding Linux Kernel*, O'Reilly & Associates, 2001, ISBN 0-596-00002-2.

8.2. Applications (Laboratory, Projects)		Teaching methods	Notes
1	**Introduction**. Presentation of the lab / project OS (Pintos or HAL9000).	(1) students are presented a very brief overview of the most important and difficult aspects of the working subject; (2) students are given at the beginning of each class a short evaluation quiz; (3) students are given a hands-on tutorial to practice with working subject's aspects and to solve problems (4) students are given challenging problems for extra credit;	
2	**OS Debugging**. Techniques and tools to debug an OS.		
3	***Thread management***. *Support for managing multiple executions inside the OS kernel.*		
4	***Synchronization mechanisms***. *Implementation of locks, semaphores and condition variables.*		
5	**Scheduling algorithms**. Round-Robin, priority-based, multi-level feedback queue (MLFQ).		
6	**User application support (1). System call mechanism**. Learn the way the system calls are implemented and used. Basic system call handling in the OS kernel.		
7	**User application support (2)**. Basic memory management. Implementation of basic system calls.		
8	**User application support (3)**. Multi-threaded application support.		
9	**Virtual memory (1)**. Lazy-loading mechanisms.		
10	**Virtual memory (2)**. Memory-mapped files.		
11	**Virtual memory (3)**. Swapping and page-replacement algorithms.		
12	**File system** (1). Basic aspects of file implementation.		
13	**File system** (2). Basic aspects of directory implementation.		
14	**Lab examination.**		

Bibliography

1. Lecture slides and laboratory text and support at http://moodle.cs.utcluj.ro/
2. Pintos and HAL9000 manual.

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

OS knowledge is a fundamental requirement in the CS field. The OSD course presents techniques for hardware and software resources management, which are applicable on any complex management software application. Besides, it provides students detailed knowledge about modern OSes' internals, making them capable of developing more efficient applications. We follow the ACM curricula guide. OSD course's curriculum also maps the IT companies expectations, especially those dealing with direct access to OS services or developing kernel drivers or modules. Such companies are, for instance, system and data security and antivirus detection companies. Usually the teachers in charge of lab classes are former graduate students of our CS program with consistent experience in industry, in companies like those mentioned above. They are permanently consulted regarding the OS course curriculum and its applicability in real projects in industry.

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment	10.3	Weight in the final

				methods		grade
Course		Students must understand fundamental OS structure and design alternatives and be able to explicitly describe it. They must also be able to apply their knowledge to give solutions to specific OS design problems.		Oral examination. Detailed discussion about design alternatives of different OS components.		0.67
Applications		Students must be able to develop different OS components writing code in C and assembly.		*Lab*: implementation of different problems in the lab OS. *Project*: presentation of design and implementation solutions		0.33

10.4 Minimum standard of performance

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

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

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

Be able to describe the internal aspects of the fundamental OS design principles, like locks, priority-based and RR scheduling, system calls, paging, virtual memory.

Be able to write functional C code that pass at least one test from the provided test set.

Course responsible
Assoc.prof.dr.eng. Adrian Colesa

Head of department
Prof.dr.eng. Rodica Potolea

SYLLABUS

1. Data about the program of study

1.1	Institution	The Technical University of Cluj-Napoca
1.2	Faculty	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	47.2

2. Data about the subject

2.1	Subject name	User Interface Design
2.2	Subject area	Computer Science and Information Technology
2.3	Course responsible/lecturer	Prof.dr.eng. Gorgan Dorian – dorian.gorgan@cs.utcluj.ro
2.4	Teachers in charge of applications	S.l.dr.eng. Ştefănuţ Teodor, teodor.stefanut@cs.utcluj.ro, As.dr.eng. Sabou Adrian, adrian.sabou@cs.utcluj.ro

2.5	Year of study	IV	2.6	Semester	7	2.7	Assessment	exam	2.8	Subject category	DS/OP

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			
7	**User Interface Design**	**2**	**-**	**2**	**1**	**28**	**-**	**2**	**1**	**85**	**155**	**6**

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	40
Supplementary study in the library, online and in the field	10
Preparation for seminars/laboratory works, homework, reports, portfolios, essays	20
Tutoring	6
Exams and tests	9
Other activities	0

3.7	Total hours of individual study	85
3.8	Total hours per semester	155
3.9	Number of credit points	6

4. Pre-requisites (where appropriate)

4.1	Curriculum	Computer programming (C or Java) Elements of Computer Assisted Graphics Software Engineering
4.2	Competence	The fundamental methodology for the development of software applications

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 competences

Professional competences	**C5 -** Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems (6 credite) **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 the computing system to the specifc requirements of the application field **C5.3 -** Using fundamental principles and methods for ensuring the security, the safety and ease of exploitation of the computing systems **C5.4 -** Proper utilization of the quality, safety and security standards in the field of 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 competences	N/A

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

7.1	General objective	Study and experiment the methodology of interactive software applications development. Study Human-Computer interaction techniques.
7.2	Specific objectives	1. Apply the user centered software development methodology 2. Study and experiment the techniques that are specific to the flexible methodology of the development of interactive applications and graphical user interfaces 3. Implementation of new and efficient human-computer interaction techniques 4. Usability evaluation in interactive applications

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction. History	New multimedia teaching approaches will be used in classes. The course is interactive and includes demonstrations that exemplify different user interaction techniques and the software development methodology.	During the semester and before each exam there are a few preparation hours planned.
2	User interface development concepts		
3	Input and output communication concepts		
4	User oriented design methodology		
5	User interface design methodology		
6	User interface usability		
7	User requirements definition		
8	Task description and analysis		
9	User interface prototyping		
10	Cognitive walkthrough and heuristic evaluation		
11	Interaction styles and techniques		
12	Web technologies. Audio and video technologies		
13	Wireless technologies		
14	User interface development environments		

Bibliography

8. Shneiderman B.: "*Designing the User Interface. Strategies for Effective Human Computer Interaction*", Addison-Wesley, 1992.
9. Galitz W.O.: "*The Essential Guide to User Interface Design*". John Wiley & Sons, 1997.

In virtual library

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

8.2. Applications (Laboratory)		Teaching methods	Notes
1	Best practice in UI development	Documentation and examples	
2	Introduction into HTML		

3	Basic notions of CSS formatting	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.	
4	User interaction through JavaScript		
5	Intermediate knowledge assessment		
6	Best practice in Mobile Applications development		
7	Introduction in Android		
8	UI layout best practices. List controls.		
9	UI elements for advanced user interactions		
10	Intermediate knowledge assessment		
11	Introduction in Windows Mobile		
12	UI layout best practices. List controls.		
13	UI elements for advanced user interaction		
14	Final knowledge assessment		
Applications (Projects)		Teaching methods	Notes
1	Project proposal: subject, methodology, phases, organization, project contents, project evaluation	Documentation and examples will be available to the students on a dedicated server.	Each student will have to develop a specific project based on the knowledge acquired at the laboratory hours.
2	Project definition. Evaluation report		
3	Task description and analysis		
4	Low fidelity prototyping, and scenarios		
5	Cognitive walkthrough		
6	Heuristic evaluation		
7	Prototyping plan		
8	Prototype codification		
9	User test cases		
10	Prototype evaluation and evaluation reports		
11	Iterative enhancement of the prototype		
12	Final user interface development		
13	Document writing		
14	Project presentation and evaluation		
Bibliography 1. Teodor Ștefănuț, Dănuț Mihon, Victor Bâcu, Dorian Gorgan. *Proiectarea interfețelor utilizator - Îndrumător de laborator*, Editura U.T. PRESS Cluj-Napoca, ISBN 978-606-737-068-3, http://biblioteca.utcluj.ro/, 2015. ***In virtual library*** 2. Course resources, http://cgis.utcluj.ro/teaching/			

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 user centered methodology for the development of interactive software applications and graphical interfaces. 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	10.1	Assessment criteria	10.2	Assessment methods	10.3	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.		Evaluation is performed through written examination.		60% (E)
Course activity		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 a very short tests.		10% (AC)

Applications		Laboratory assessment evaluates the practical abilities obtained by the students. Through project 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 exam and project assessment.		25% (C) 25% (P)
10.4 Minimum standard of performance						
Graduation requirement: M≥5, final mark M=0.4*E+0.25*C+0.25*P+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	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	48.1

2. Data about the subject

2.1	Subject name	Pattern Recognition Systems
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	Conf. dr. eng. Florin Oniga, Sef lucrari Raluca Brehar, Sef lucrari Ion Giosan{ Florin.Oniga, Raluca.Brehar, Ion.Giosan}@cs.utcluj.ro

2.5	Year of study	IV	2.6	Semester	7	2.7	Assessment	exam	2.8	Subject category	DS/OP

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			
7	**Pattern Recognition Systems**	**2**	**-**	**2**	**1**	**28**	**-**	**28**	**14**	**85**	**155**	**6**

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								28
Supplementary study in the library, online and in the field								20
Preparation for seminars/laboratory works, homework, reports, portfolios, essays								28
Tutoring								4
Exams and tests								5
Other activities								0

3.7	Total hours of individual study	85
3.8	Total hours per semester	155
3.9	Number of credit points	6

4. Pre-requisites (where appropriate)

4.1	Curriculum	Image Processing
4.2	Competence	Computer programming, Data structures and algorithms, Probability Theory, Artificial Intelligence.

5. Requirements (where appropriate)

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

6. Specific competences

Professional competences	**C4 –** Improving the performances of the hardware, software and communication systems (2 credits) **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 the computing system to the specifc requirements of the application field **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 -** Creating a project including the problem's identification and analysis, its design and development, also proving an understanding of the basic quality requirements **C6 –** Designing intelligent systems (2 credits) **C6.1 -** Describing intelligent systems' components **C6.2 -** Using domain-specific tools for explaining the operation of intelligent systems **C6.3 -** Applying 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	Knowledge, understanding and use of concepts related to pattern recognition.
7.2	Specific objectives	Knowledge, understanding and use of model-based pattern recognition methods using statistical approaches, linear discriminant methods, support vectors, and ensemble of classifiers. Knowledge, understanding and use of the specific operations of a pattern recognition system: data preprocessing, dimensional reduction, relevant feature selection, building the prediction model, selection of the optimum model, performance analysis.

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Introduction	Interactive teaching, using oral presentations supported by multimedia tools, consultations, involving students in research and	N/A
2	Probability Review		
3	Bayesian Decision Theory 1		
4	Bayesian Decision Theory 2		
5	Bayesian Decision Theory Case Studies		
6	Maximum Likelihood Estimation		
7	Kernel Density Estimation		
8	K Nearest Neighbors Estimation		
9	Linear Discriminant Functions		
10	Kernel Methods		

11	Support Vector Machines	development activities.	
12	Ensemble Methods		
13	Clustering methods		
14	Feature Selection and Performance Estimation		

Bibliography

1. Richard O. Duda, Peter E. Hart , David G . Stork, "Pattern Clasification", *John Wiley and Sons,* 2001.
2. C.M. Bishop, “Pattern Recognition and Machine Learning”, *Springer*, 2006
3. S. Theodoridis, K. Koutroumbas, “Pattern Recognition”, 2-nd Edition, *Academic Press*, 2008.
4. K. Murphy, “Machine Learning: A Probabilistic Perspective”, The MIT Press, 2012

8.2. Applications (Laboratory)		Teaching methods	Notes
1	Introduction		
2	Least Mean Squares Line Fitting	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
3	RANSAC – fitting a line to a set of points		
4	Hough Transform for line detection		
5	Distance Transform (DT). Pattern Matching using DT		
6	Probability Density Estimation		
7	K-Means Clustering		
8	Principal Component Analysis		
9	K-Nearest Neighbor Classifier		
10	Naïve Bayes Classifier: Simple Digit Recognition Application		
11	Linear classifiers. Perceptron algorithm		
12	Adabost with Decision Stumps		
13	Support Vector Machine		
14	Lab Assessment		
Applications - Projects			
1	Topic assignment (week 1, 2)		
2	Analyzes, specification and design (week 3,4)		
3	Presentation of the approach (week 5,6)		
4	Implementation (week 6,7,8,9,10); Intermediate presentation (week 9,10)		
5	Evaluation and optimization (week 11,12)		
6	Report elaboration (week 12,13)		
7	Final Presentation (week 13,14)		

Bibliography

S. Nedevschi, "Lecture Notes", ftp.utcluj.ro/pub/users/nedevschi/SRF/

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 pattern recognition. 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	Head of department
Prof. dr. eng. Sergiu Nedevschi	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	48.2

2.Data about the subject

2.1	Subject name	Translators Design									
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	IV	2.6	Semester	7	2.7	Assessment	exam	2.8	Subject category	DS/OP

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			
7	**Translators Design**	**2**	**-**	**2**	**1**	**28**	**-**	**28**	**14**	**85**	**155**	**6**

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								30
Supplementary study in the library, online and in the field								15
Preparation for seminars/laboratory works, homework, reports, portfolios, essays								27
Tutoring								10
Exams and tests								3
Other activities								0

3.7	Total hours of individual study	85
3.8	Total hours per semester	155
3.9	Number of credit points	6

4. Pre-requisites (where appropriate)

4.1	Curriculum	Formal Languages and Translators, Computer Programming, Data Structures and Algorithms
4.2	Competence	- Basic knowledge of programming and data structures (preferably in the C and Java languages) - Concepts of generative grammars and formal languages - To know the basic principles in the design of interpretors and translators for languages artificial - Basic knowledge of relational databases and web applications

5. Requirements (where appropriate)

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

6. Specific competences

Professional competences	**C4 -** Improving the performances of the hardware, software and communication systems (2 credits) **C4.1** - Identifying and describing the defining elements of the performances of the hardware, software and communication systems **C4.2** - Explaining the interaction of the factors that determine the performances of the hardware, software and communication systems **C4.3 -** Applying the fundamental methods and principles for increasing the performances of the hardware, software and communication systems **C4.4** - Choosing the criteria and evaluation methods of the performances of the hardware, software and communication systems **C4.5** - Developing professional solutions for hardware, software and communication systems based on performance optimization **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 the computing system to the specific requirements of the application field **C5.3** - Using fundamental principles and methods for ensuring the security, the safety and ease of exploitation of the computing systems **C5.4** - Proper utilization of the quality, safety and security standards in the field of 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 **C6 -** Designing intelligent systems (1 credit) **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 the criteria and evaluation methods for the quality, performances and limitations of intelligent 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	- To know the phases of programming language translators: lexical analysis, syntactic analysis, and code generation. - To master the tree structure representation of XML documents.
7.2	Specific objectives	- To know the classes of languages for which efficient translators and interpreters can be implemented. - To know the rules for processing typical statements for interpreters. - To understand the difference between structure and presentation of documents. - By using the Java language, to implement parsers of type SAX and DOM for XML documents containing DTD validation information. - By using the Java language, to implement XML document transformators, based on XSLT transformations.

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Descriptive tools: extended Backus-Naur form.	- The main ideas	N/A

2	Regular grammars and finite automata: finite automata, state diagrams and regular expressions.	with multimedia tehniques - Details and examples at the blackboard, in interaction with the students - There are consultation hours - Students are invited to collaborate in research projects	
3	Context-free grammars and pushdown auromata: examples.		
4	Lexical analysis: modules and interfaces (decomposition of the grammar, lexical analyzer interface), construction of the lexical analyzer (state diagrams, reserved words method).		
5	LL parsers: the LL(1) parsing algorithm for extended BNF grammars.		
6	LL parsers: computation of FIRST and FOLLOW sets.		
7	LL parsers: examples of recursive-descent applications.		
8	Theoretical results concerning the LL(*k*) and LR(*k*) grammars.		
9	LR parsers: LR(0) states, SLR(1) grammars.		
10	LR parsers: LALR(1) grammars.		
11	LR parsers: the LALR(1) algorithm.		
12	LR parsers: shift-reduce transitions, chain production elimination.		
13	LR parsers: LR table compression.		
14	Basic concepts of attribute grammars.		

Bibliography

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

8.2. Applications (Laboratory, Projects)		Teaching methods	Notes
L1	W3C XML Recommendation version 1.0.	Brief presentation at the blackboard, examples and exercises implemented and tested on the computers, followed by homework for each topic	
L2	Parsing XML documents (“well-formed”).		
L3	XML document validation using DTD.		
L4	XML document validation using XSD.		
L5	W3C XPath Recommendation version 1.0.		
L6	W3C XSLT Recommendation version 1.0.		
L7	XSL-FO (XML Stylesheet Language - Formatting Objects) 1.1.		
L8	XML usage for storing Microsoft Office 2007/2010 documents - Apache POI/XSSF 3.13.		
L9	XML Data Binding using JAXB 2.0.		
L10	W3C XQuery Recommendation 1.0, XPath & XSLT 2.0.		
L11	XML document storage in databases.		
L12	eXist-db XML native DBMS 2.2.		
L13	XQuery 3.0 and Update extensions in eXist-db		
L14	XRX (XForms + REST(ful) + XQuery) Web Application Architecture & Development in eXist-db.		
P1	Building recursive-descent parsers from extended BNF grammars.	Brief presentation at the blackboard, implementing and testing the assignment on the computer	
P2	Recursive-descent (RD) applications:.expression evaluator.		
P3	RD applications: interpreter for a language operating on binary trees.		
P4	RD applications: interpreter for a language operating on lists.		
P5	RD applications: interpreter for a language operating on matrices.		
P6	RD applications: code generator for an imperative language.		
P7	RD test.		

Bibliography

1. W3C Recommendations (i.e. Standards) appropriate for each Topic.
2. Teach Yourself XML in 21 days, Steven Holzner, SAMS Publishing, 2004.
3. XML Pocket Reference, 3rd Edition, Simon St. Laurent, Michael Fitzgerald, O’Reilly Media, 2005.
4. 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 principles of efficient design and implementation 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 ARAIS).

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		44%
Applications		- Problem-solving skills - Attendance, Activity		- Assessment of the XML activity, homework, and written exam - Assessment of the RD activity and test		35% 21%
10.4 Minimum standard of performance						
Modelling 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	49.1

2.Data about the subject

2.1	Subject name	Marketing
2.2	Subject area	Computer Science and Information Technology
2.3	Course responsible/lecturer	Lector dr. Veronica Maier – veronica.maier@enm.utcluj.ro
2.4	Teachers in charge of applications	

2.5	Year of study	IV	2.6	Semester	7	2.7	Assessment	Colloquium	2.8	Subject category	DC/OP

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			
7	**Marketing**	**2**	**-**	**-**	**-**	**28**	**-**	**-**	**-**	**48**	**76**	**3**

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	14
Supplementary study in the library, online and in the field	7
Preparation for seminars/laboratory works, homework, reports, portfolios, essays	14
Tutoring	9
Exams and tests	4
Other activities	-

3.7	Total hours of individual study	48
3.8	Total hours per semester	76
3.9	Number of credit points	3

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 (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 the computing system to the specifc requirements of the application field **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
Cross competences	**CT1 –** Honorable, responsible, ethical behavior, in the spirit of the law, in order to ensure the professional reputation (1 credit)

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

7.1	General objective	Understand, assimilate and use of basic marketing concepts, principles and techniques
7.2	Specific objectives	Understand the marketing orientation, the components of the marketing mix, the methods and techniques for collecting and processing marketing data, the strategic analysis determining the competitive advantage, the market segmentation and the buying decision criteria.

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Marketing role in big and small companies and in society: to contribute to company prosperity through creating a high value for the customers; to make the company responsible on the long range in front of community, society and environment. Marketing specific activities	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	Marketing concepts (philosophies) in contemporary organizations: Volume? Quality? Sales? Customer satisfaction?		2h
3	Marketing environment analysis. Micro and macro environment: suppliers, interest groups, customers, economic, demographic, technological, natural, legal and cultural environment		2h
4	Marketing research: research plan, data collection; data analysis quantitative and qualitative techniques; experiments; research report. Marketing information systems		2h
5	Marketing strategic planning: creating and maintaining the balance between objectives, resources and market opportunities. Methods of strategic analysis.		2h
6	Designing the strategic plan at four levels: company, divisions, strategic units and brands		2h
7	Consumer behavior analysis: patterns of behavior		2h
8	Buying decision process		2h
9	Market segmentation. Criteria and methods of market segmentation		2h
10	Product policy. Product life cycle. Researching and developing new products		2h
11	Product strategies for the life cycle stages. Positioning strategies		2h
12	Pricing. Pricing policy objectives. Pricing and legal constraints. Pricing policies: market penetration and market skimming		2h
13	Product distribution. Choosing the distribution channels. Managing and controlling the distribution channels		2h

14	Marketing communication. Communication process. Marketing communication mix: advertising,		2h

Bibliography

1. D. Catana, Gh. A. Catana, Fundamentals of Marketing, UTPRES, 2009
2. Gh. A. Catana, A. Dobra Constantinescu, Marketing in powerpoint, UTPRES, 2004
3. Gheorghe Alexandru Catană*, Marketing: filozofia succesului de piaţă*,vol. I, Editura Dacia, Cluj-Napoca 2003

Virtual materials

1. D. Catana, Gh. A. Catana, Marketing, 2010 www.marketing.utcluj.ro

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 marketers, 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		Not the case		-		-
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	49.2

2. Data about the subject

2.1	Subject name	Personal and professional development									
2.2	Subject area	Computer Science and Information Technology									
2.3	Course responsible/lecturer	Dipl. Psy. Dorin Stanciu PhD, Lecturer (ionut.stanciu@dppd.utcluj.ro)									
2.4	Teachers in charge of aapplications	-									
2.5	Year of study	IV	2.6	Semester	7	2.7	Assessment	Colloquium	2.8	Subject category	DC/OP

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			
7	**Personal and professional development**	**2**	**-**	**-**	**-**	**28**	**-**	**-**	**-**	**48**	**76**	**3**

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	16
Supplementary study in the library, online and in the field	14
Preparation for seminars/laboratory works, homework, reports, portfolios, essays	14
Tutoring	-
Exams and tests	4
Other activities	-

3.7	Total hours of individual study	48
3.8	Total hours per semester	76
3.9	Number of credit points	3

4. Pre-requisites (where appropriate)

4.1	Curriculum	-
4.2	Competence	Baccalaureate level of instruction and general knowledge Beginner literacy with desktop applications, including Internet utilization

5. Requirements (where appropriate)

5.1	For the course	Auditorium or large lecture room. Audio-video installation for on-screen presentations (with room speakers). WiFi or cable Internet connectivity.
5.2	For the applications	Auditorium or large lecture room. Audio-video installation for on-screen presentations (with room speakers). WiFi or cable Internet connectivity. Writing board (classical or interactive) / Flip chart

6. Specific competences

Professional competences	**C5** - Design, lifecycle management, integration and 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** - 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	**TC1 -** Honorable, responsible, ethical behavior, in the spirit of the law, to ensure the professional reputation (1 credit)

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

7.1	General objective	To provide the course graduate with a set of competencies, skills and level of knowledge about him/herself which allows the formation of a competitive advantage and to provide to course graduate with a better understanding of his/her current academic status and curriculum
7.2	Specific objectives	To facilitate domain-specific learning and knowledge acquisition by providing a larger perspective on personal and professional development; To enhance personal determination and academic engagement as a basis for future competitiveness; To allow the course graduate to acquire specific tools and skills needed for personal and professional assessment, engagement, planning, organizing, expression, and networking.

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1	Key-concepts of personal and professional development. Self-concepts, qualities and virtues, self-awareness and self-knowledge (self-assessment versus external evaluations)	Interactive lectures: - Exposition - Discourse - Debating - Case studies - Problem-solving - Heuristic conversations - Role playing	
2	Motivation and self-determination. Goals, objectives, interests, needs, desires, ideals, aspirations, expectations and incentives.		
3	Learning and learning styles. Self-directed learning, adult learning and continuous learning (lifelong learning)		
4	Social modelling and key-persons/models. The basics of social learning and the significant others		
5	Rationality, control, self-regulation and decision making. Processes, strategies and decision making tools		
6	Critical thinking and scientific reasoning. Cognitive biases, logical fallacies and cognitive distortions		
7	Assertive communication, persuasion and negotiation		
8	Significant personal objectives: Qualities and virtues		
9	Significant personal and professional objectives: Health, safety, fulfilment, satisfaction and happiness		
10	Tools, means and methods for productivity enhancement: Strategic planning, Decision-making, Information management		
11	Tools, means and methods for productivity enhancement: Organization, scheduling, planning and budget management		
12	Tools, means and methods for productivity enhancement: CV building		

	(principles, alternatives, instruments)		
13	Tools, means and methods for productivity enhancement: Social networking, Professional networking (virtual dedicated networks and communities)		
14	Personal and professional development plans. Design and presentation		

Bibliography

Study materials: Course synthesis, Lecture presentations, additional multimedia presentations)

Aggarwal, R. S. (2000). *A Modern Approach to Vernbal & Non Verbal Reasoning*: S.Chand.

Andrews, K. R. (1971). *The concept of corporate strategy*: Dow Jones-Irwin.

Ariely, D. (2009). *Predictably Irrational: The Hidden Forces that Shape Our Decisions*: HarperCollins Publishers.

Aspinwall, L. G., & Staudinger, U. M. (2003). *A Psychology of Human Strengths: Fundamental Questions and Future Directions for a Positive Psychology*: American Psychological Association.

Bercovitch, J., Kremenyuk, V., & Zartman, I. W. (2008). *The SAGE Handbook of Conflict Resolution*: SAGE Publications.

Bishop, S. (2013). *Develop Your Assertiveness*: Kogan Page.

Bolt, M. (2004). *Pursuing Human Strengths: A Positive Psychology Guide*: Worth Publishers.

Cameron, N., & Bogin, B. (2012). *Human Growth and Development*: Academic Press.

Caputi, P., Viney, L. L., Walker, B. M., & Crittenden, N. (2011). *Personal Construct Methodology*: Wiley.

Cava, R. (2004). *Dealing With Difficult People: How to Deal With Nasty Customers, Demanding Bosses and Annoying Co-Workers*: Firefly Books, Limited.

Chamorro-Premuzic, T., & Furnham, A. (2005). *Personality and Intellectual Competence*: Taylor & Francis.

Clark, K. S., Murphy, M. M., & Banas, S. L. (2009). *Handling Peer Pressure*: Facts On File, Incorporated.

Deutsch, M., Coleman, P. T., & Marcus, E. C. (2011). *The Handbook of Conflict Resolution: Theory and Practice*: Wiley.

Dreher, J. C., & Tremblay, L. (2009). *Handbook of Reward and Decision Making*: Elsevier Science.

Dunne, G. (2003). *Anger and Conflict Management: Personal Handbook*: Personhood Press.

Fells, R. (2012). *Effective Negotiation: From Research to Results*: Cambridge University Press.

Freeley, A. J. (2012). *Argumentation and Debate, 13rd ed*: Wadsworth/Cengage Learning.

Gigerenzer, G. (2000). *Adaptive Thinking : Rationality in the Real World*: Oxford University Press, USA.

Gilbert, D. (2006). *Stumbling on Happiness*: Knopf Doubleday Publishing Group.

Gilovich, T. (2008). *How We Know What Isn't So*: Free Press.

Glickman, R. (2002). *Optimal Thinking: How to Be Your Best Self*: Wiley.

Hammond, J. S., Keeney, R. L., & Raiffa, H. (1999). *Smart Choices: A Practical Guide to Making Better Decisions*: Harvard Business School Press.

Haslam, N. (2007). *Introduction to Personality and Intelligence*: SAGE Publications.

Hunter, D. (2013). *A Practical Guide to Critical Thinking: Deciding What to Do and Believe*: Wiley.

Johnson, G., & Whittington, R. (2009). *Fundamentals of Strategy*: Pearson Education, Limited.

Kahneman, D. (2011). *Thinking, Fast and Slow*: Farrar, Straus and Giroux.

Kuhn, D. (2009). *Education for Thinking*: Harvard University Press.

Larson, C. U. (2009). *Persuasion and Responsibility: Reception and Responsibility*: Cengage Learning.

Lau, J. Y. F. (2011). *An introduction to critical thinking and creativity: Think more, think better*. Hoboken, N.J.: Wiley.

Lilley, R. (2013). *Dealing with Difficult People*: Kogan Page.

Lopez, S. J. (2008). *Positive Psychology: Discovering human strengths*: Praeger.

Macnamara, J. (1986). *A Border Dispute: The Place of Logic in Psychology*: A Bradford Book.

MTD Training. (2010). *Dealing with Conflict and Complaints*: MTD Training & Ventus Publishing ApS.

Mynatt, J. (2009). *Encyclopedia of Management*: Gale.

Narvaez, D., & Lapsley, D. K. (2009). *Personality, Identity, and Character: Explorations in Moral Psychology*: Cambridge University Press.

Oliver, D. (2010). *How to Negotiate Effectively*: Kogan Page.

Peterson, C., & Seligman, M. E. P. (2004). *Character Strengths and Virtues: A Handbook and Classification*: Oxford University Press, USA.

Raiffa, H., Richardson, J., & Metcalfe, D. (2002). *Negotiation Analysis: The Science and Art of Collaborative Decision Making*: Belknap Press of Harvard University Press.

Rechner, A. (2009). *The in Crowd: Dealing with Peer Pressure*: Compass Point Books.

Schick, T., & Vaughn, L. (2013). *How to Think About Weird Things: Critical Thinking for a New Age*: McGraw-Hill Education.

Seligman, M. E. (2011). *Learned Optimism: How to Change Your Mind and Your Life*: Knopf Doubleday Publishing Group.

Smith, J. C. (2011). *Pseudoscience and Extraordinary Claims of the Paranormal: A Critical Thinker's Toolkit*: Wiley.

Staley, C. C. (2007). *Focus on college success*. Boston, MA: Wadsworth Cengage Learning.

Stanciu, I. D. (2013). *Raţionalitate şi control în autoreglarea învățării la studenţi. Modelare conceptuală și intervenții experimentale de validare*. Cluj-Napoca: Presa Universitară Clujeană.

Stanovich, K. E. (2009). *Decision making and rationality in the modern world*: Oxford University Press.

Sternberg, R. J., & Zhang, L. (2001). *Perspectives on Thinking, Learning, and Cognitive Styles*: Taylor & Francis.

Tavris, C., & Aronson, E. (2008). *Mistakes Were Made (But Not by Me): Why We Justify Foolish Beliefs, Bad Decisions, and Hurtful Acts*: Houghton Mifflin Harcourt.

van den Brink-Budgen, R. (2000). *Critical Thinking for Students: Learn the Skills of Critical Assessment and Effective Argument*: How To Books.

Zhang, L., Sternberg, R. J., & Rayner, S. (2012). *Handbook of Intellectual Styles: Preferences in Cognition, Learning, and Thinking*: Springer Publishing Company.

8.2. Applications (Seminars, Laboratory, Projects)		Teaching methods	Notes
1	N/A		
Bibliography			

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

By completing this course, the course graduate is expected to have acquired a series of specific and general declarative and procedural knowledge, as well as have built a series of competencies, which, in their togetherness contribute to an increased capacity to find employment, communicate professionally and informally, collaborate and work closely with other professionals and non-professional, and an increased ability to promote and capitalize upon personal and professional traits and activities.

10. Evaluation

Activity type	10.1	Assessment criteria	10.2	Assessment methods	10.3	Weight in the final grade
Course		Standardized written test with multiple choice questions.		Written test Duration: 1 hr.		60%
Applications		Collaborative and individual semester projects. Collaborative and individual homework. Assessment criteria include: accuracy/precision, completeness, fluency, and relevance		Individual portfolio		40%
10.4 Minimum standard of performance						
The total weighed score exceeds the equivalent of 5/10 of the final grade. Each assessment exceeds 50% of the allotted grading.						

Course responsible
DPsy. Dorin Stanciu PhD, Lecturer

Head of department
Prof.dr.eng. Rodica Potolea