MINISTRY OF NATIONAL EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA

SYLLABUS

1. Study Program

1.1	Higher Education Institute	Technical University of Cluj-Napoca
1.2	Faculty	Electronics, Telecommunications and Information Technology
1.3	Department	Communications
1.4	Study domain	Electronics and Telecommunications Engineering
1.5	Study level	Master
1.6	Study program/ Qualification	Multimedia Technologies/ Telecommunications/ Master
1.7	Type of education	IF (Full-time learning)
1.8	Discipline code	TM-E-03.00/ TC-E16.40

2. Discipline

2.1	Discipline name					Advanced Software Elements in Telecommunications						
2.2	Subject area					Electronics and Telecommunications Engineering						
2.3	Course responsible/lecturer					Professor Mircea-Florin Vaida, PhD Mircea.Vaida@com.utcluj.ro						
2.4	Teachers in charge of applications					Professor Mircea-Florin Vaida, PhD. Assistant Professor Cosmin Striletchi, PhD.						
2.5	Year of study	I	2.6	Semester	1	2.7	Evaluation	Exam	2.8	Type of discipline	DA/DI	

3. Total estimated time

Year/ Sem	Discipline name	No. of weeks	Course	Applications			Course	Applications			Indiv. study	TOTAL	ECTS
			[hours/week]				[hours/week]						
			C	S	L	P		S	L	P			
II/3	Comunicatii cognitive	14	2	0	1	0	28	0	14	0	58	100	4

3.1	Number of hours per week	4	3.2	course	2	3.3	applications	1
3.4	Total hours per curriculum	42	3.5	course	28	3.6	applications	14

Individual study	Hours
Study based on manuals, course materials, references and notes	14
Supplementary documentation in libraries, electronic platforms and on field	10
Preparation of seminars/laboratories, homeworks, essays, portfolios	10
Tutorial work	7
Assesments	3
Other activities	14

3.7	Total hours of individual study	58
3.8	Total hours per semester	100
3.9	ECTS	4

4. Prerequisites (if necessary)

4.1	Curriculum	Computer Programming - Languages, Algorithms, Software Engineering
4.2	Competences	Basic concepts on software development, object oriented programming concepts, algorithms and programming techniques, the basics of software engineering. Ability to use an integrated development environment (Visual Studio C + + / C #, Eclipse, Java, etc.).

5. Requisites (if necessary)

5.1		Course	Video-projector, screen, whiteboard
5.2		Applications	PCs with Internet access

6 Specific competences acquired

Professional competences	Theoretical knowledge (What do the student should know)	The students will know: - about software models development -modern software management methodologies -human factors in programming and their implications -UML -generic, multithreading, parallel/multicore programming -advanced C++ library using – STL. -C++0x/1y/2z new facilities -testing software applications -basic cloud computing
	Acquired skills (What the student is able to do)	The students will be able to: - To distinguish between simple and professional software development; - To use software development methodologies in Agile/Kanban teams; - To use object software design methodologies; - To use the software design methodology based on UML diagrams; - Generic, multithreading, parallel/multicore programming. - Understand and know the role of testing software applications; - To develop applications using new facilities of C++0x/1y/2z; - Basics in software cloud.
	Acquired abilities (what equipment/ instruments/ softwares the student is able to handle)	The students will be able to use: Generic programming (C++, C# or Java). Object oriented applications and complex algorithms programming development. To use JUnit and dedicated tools for manual and automated testing software applications. Use a C++1y/2z IDE for new language features.
Transversal competences		CT3 Adapting to new technologies, professional and personal development through continuing education using electronic documentation and printed sources, in Romanian and in at least one international language (English). Competencies for analysis and synthesis and optimization systems thinking. Flexibility in thinking and ability to work with interdisciplinary concepts and tools.

7 Discipline objectives (based on the grid of specific competences acquired)

7.1	General objective	To develop advanced software in telecommunication.
7.2	Specific objectives	Software models, methodologies. Generics and advanced software development. Software testing. Basic cloud software development.

8. Contents

8.1. Course (titles)		Teaching methods	Obser-vations
1	Software models to create applications in telecommunications. The life cycle of programs and systems.	Presentation, discussions	Videoprojector
2	Modern management methodologies for software applications: Agile / Scrum and Just in Time / Kanban.		
3	Human factors in programming and their implications. Information		

	Systems, IS and Information Technology, IT. Psychology and human memory in the Software Engineering. User needs. IS design methodologies.		
4	Classification methodology, structural: SSADM-MERISE, objectual: OOD, OOT and formal. Interface and implementation. Parnas' principle. UML programming: basic concepts, evolution. Class and object diagrams. Relations between classes, interfaces and objects.		
5	Collaboration diagrams, connections, links, interactions: repetitive, conditional, multithreading, preconditions, synchronous, asynchronous. State charts. Deployment diagrams. The life cycle of software components in OO methodology (OOM). Case study.		
6	Alternative educational methodologies to develop software applications.		
7	Evolution of generic programming concepts: functions / methods and template classes in C++. Evolution and use.		
8	New features introduced in C++0x/1y/2z.		
9	STL library. Define and use.		
10	Generic programming in Java. Evolution and use.		
11	Multithreading, parallel/multicore programming. Concepts, usage.		
12	Manual and automated software testing applications.		
13	JUnit and other software testing facilities of applications.		
14	Cloud computing. Introduction, comparison of existing technologies.		
8.2. Applications (laboratory work)		Teaching methods	Obser-vations
1	Software engineering principles and objectives reflected in software development	Simulations, experiments	PC, simulator
2	Fundamentals in application programming using an object oriented language (C / C ++, C #, Java).		
3	Methodologies management /design of software applications considering company standard specifications		
4	Alternative educational methodologies. Group work in teams using the Enneagram and MBTI types.		
5	Writing and evaluating a scientific report involving advanced software development. Define topics for teams.		
6	Writing articles in journals and conferences. The use of UML diagrams in the software. Defining mechanisms creating teams.		
7	Developing software in C++ using C ++0x/1y/2z and classes and templates functions/methods.		
8	Developing software-using STL.		
9	Intermediate stage pre-assessment team working		
10	Develop software using Java Generics, multithreading, parallel/multicore		
11	Testing applications using JUnit		
12	Laboratory evaluation homework		
13	Presenting a scientific report on a software topic according to company standard specifications		
14	Evaluation of the teams activity		

References:

1. Mircea-Florin Vaida, Cosmin Porumb, Radu Fotea, Florin Hurducas, Liviu Lazar, Java 2 Enterprise Edition (J2EE). Aplicatii multimedia, Editura Albastra, 2003
2. M.F.Vaida, P.G.Pop, C.Striletchi, L.Chiorean, CG.Loghin, Tehnologii avansate privind dezvoltarea aplicatiilor software in limbajul C/C++,C# si Java, Casa Cartii de Stiinta, 2006
3. B. Stroustrup, The C++ programming language, Addison-Wesley, 2013
4. S. Tanasa, C. Olariu, Dezvoltarea aplicatiilor Web folosind Java, Ed. Polirom 2005
5. L. Alboaie, S. Buraga, Servicii Web. Concepte de baza si implementari, Ed. Polirom 2006
6. Mircea-Florin Vaida, Ligia-Domnica Chiorean, Lenuța Alboaie, Petre Gavril Pop, Cosmin Strileţchi, Kuderna-Iulian Benţa, Programarea în limbajul C/C++ cu elemente C++1y. Programare web C++, Casa Cartii de Stiinta, Cluj-Napoca, 2016
7. Ligia-Domnica Chiorean, Kuderna-Iulian Benţa, Mircea-Florin Vaida, Petre Gavril Pop, Cosmin Strileţchi, C/C++ - Ghid teoretic si practic, Casa Cartii de Stiinta, Cluj-Napoca, 2016
   - Web English documents on dedicated subjects

Other information: Support lab and courses materials from: http://helios.utcluj.ro/lab

9. Discipline content corroborated with the expectations of the epistemic community representatives, associations, professional and related program employers

Acquired skills will be needed in the following possible COR occupations: electronics engineer, telecommunications engineer, system and computer design engineer, or new occupations proposed to be included in COR (sales support engineer, developer of multimedia applications, network operating engineer, test engineer, project manager, traffic engineer, communications system consultant.

10. Assessment

Type of activity	10.1	Evaluation criteria	10.2	Evaluation method	10.3	The weight of the final grade
Course		Written test with question Scientific papers		Written test (T=50%) + activity during the semester (S=50%) E = T + S		E = 50%
Applications		Team work activity developed during the semester in the laboratory		Work defended at the end of semester		P = 50%
10.4 Minimum performance standard						
The final grade (N) is calculated as average of marks obtained in the evaluation of ongoing activities and application type: N = (E + P) / 2. The condition for obtaining the ECTS credits is that both components of the final grade to be higher than or equal to 5 (five).						

Date
24.02.2020

Titular
Professor
Mircea-Florin VAIDA, Ph.D.

Responsible
Professor
Mircea-Florin VAIDA, Ph.D.

Date of approval
1.10.2020

Head of Department
Professor Virgil DOBROTA, Ph.D.