



## FISA DISCIPLINEI

### 1. Date despre program

1.1	Institutiile de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2	Facultatea	Electronica, Telecomunicații și Tehnologia Informației
1.3	Departamentul	Comunicații
1.4	Domeniul de studii	Inginerie electronică și telecomunicații
1.5	Ciclul de studii	Master
1.6	Programul de studii/Calificarea	Telecomunicații/Tehnologii Multimedia
1.7	Forma de învățământ	IF - Învățământ cu frecvență
1.8	Codul disciplinei	TM03.00, TC13.40

### 2. Date despre disciplina

2.1 Denumirea disciplinei												Elemente software avansate in telecomunicatii											
2.2 Aria tematica (subject area)						Inginerie electronica si telecomunicatii																	
2.3 Responsabil de curs						Profesor: Mircea-Florin Vaida, PhD Mircea.Vaida@com.utcluj.ro																	
2.4 Titularul disciplinei						Profesor: Mircea-Florin Vaida, PhD																	
2.5 Anul de studii		I		2.6 Semestrul		1		2.7 Evaluarea		Examen		2.8 Regimul disciplinei		O									

### 3. Timpul total estimat

An/ Sem	Denumirea disciplinei	Nr. sapt.	Curs	Aplicații			Curs	Aplicații			Stud. Ind.	TOTAL	Credit
			[ore/săpt.]			[ore/sem.]							
				S	L	P		S	L	P			
I/1	Elemente software avansate in telecomunicatii	14	2		1	.5	28		14	7	89	130	5

3.1	Număr de ore pe săptămână	3	3.2	din care curs	2	3.3	aplicații	1
3.4	Total ore din planul de inv.	49	3.5	din care curs	28	3.6	aplicații	21
Studiul individual								Ore
Studiul după manual, suport de curs, bibliografie și note								24
Documentarea suplimentară în bibliotecă, pe platformele electronice și pe teren								20
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								20
Tutoriat								7
Examinări								4
Alte activități								14
3.7	Total ore studiul individual	89						
3.8	Total ore pe semestru	130						
3.9	Număr de credite	5						

### 4. Precondiții (acolo unde este cazul)

4.1	De curriculum	Programarea calculatoarelor – Limbaje, Algoritmi. Ingineria Programării.
4.2	De competente	Concepte de bază privind dezvoltarea aplicațiilor software, concepte de programare obiectuală, algoritmi și tehnici de programare, elemente de bază privind ingineria programării. Abilități de a utiliza un mediu integrat (Visual Studio:C++/C#, Eclipse-Java, etc.)

### 5. Condiții (acolo unde este cazul)

5.1	De desfășurare a cursului	Video-proiector, ecran, white board, tablă neagră
-----	---------------------------	---

5.2	De desfasurare a aplicatiilor	PC-uri cu acces Internet
-----	-------------------------------	--------------------------

## 6 Competente specifice acumulate

Competente profesionale	Cunoștințe teoretice, (Ce trebuie să cunoască)	După parcurgerea disciplinei studenții vor cunoaște: -modele de dezvoltare a aplicațiilor software -metodologii software moderne -factorii umani și implicațiile în programare -UML -programare generica -programarea paralela/multicore -biblioteci avansate C++ – STL. -facilitati noi C++0x/1y/2z -testarea aplicațiilor software -elemente de baza cloud computing
	Deprinderi dobândite: (Ce știe să facă)	După parcurgerea disciplinei studenții vor fi capabili: – Sa faca distincție între dezvoltarea aplicațiilor software didactice și profesionale; – Sa utilizeze metodologii moderne de dezvoltare a aplicațiilor software în echipe definite pe compatibilitati de tip Agile/Kanban; – Sa utilizeze metodologii de proiectare software obiectuale; – Sa utilizeze programarea generica, multithreading, paralela/multicore – Sa utilizeze metodologii de proiectare software bazate pe diagrame UML; – Sa inteleaga si sa cunoasca rolul testarii aplicațiilor software; – Sa dezvolte aplicații folosind noile facilități C++0x/1y/2z; – Sa cunoască elemente privind dezvoltare de software în cloud.
	Abilități dobândite: (Ce instrumente știe să mănuiască)	După parcurgerea disciplinei studenții vor fi capabili: – Sa cunoasca programarea generica, multithreading și cea paralela/multicore (C++, C#, Java) – Aplicatii obiectuale și cu algoritmi complecși în programare, – Sa cunoască importanța testării manuale și automate a aplicațiilor software folosind JUnit și tool-uri adecvate. – Sa utilizeze un mediu C++0x/1y/2z pentru noile facilități de limbaj.
Competențe transversale		<ul style="list-style-type: none"> <li>Identificarea unor obiective specifice de realizat, a resurselor disponibile, a condițiilor de implementare și a etapelor, termenelor, timpilor aferente realizării unei aplicații software;</li> <li>Utilizarea eficientă a surselor informaționale și a resurselor de comunicare și formare profesională asistată de calculator (internet, aplicații software de specialitate, baze de date, cursuri și seminarii online etc.) atât în limba română, cât și într-o limbă de circulație internațională.</li> </ul>

## 7 Obiectivele disciplinei (reiesind din grila competentelor specific acumulate)

7.1	Obiectivul general al disciplinei	Dezvoltarea de competențe profesionale în domeniul dezvoltării aplicațiilor software
7.2	Obiectivele specifice	Modele, metodologii software. Dezvoltare de software generic și avansat. Software testing. Elemente de baza privind dezvoltarea de software în cloud.

## 8. Continuturi

8.1. Curs (programa analitică)		Metode de predare	Observații
1	Modele software de creare a aplicațiilor în telecomunicații. Ciclul de viața al programelor și al sistemelor.	Expu nere, discu ții	Vide o- proie ctor
2	Metodologii moderne de gestiune a aplicațiilor software: Agile/Scrum		

	si Just in time/Kanban		
3	Factorii umani si implicatiile lor in programare. Sisteme informationale, IS si tehnologia informatiei, IT. Psihologia si memoria umana in procesul IP. Nevoile utilizatorilor. Metodologii de proiectare IS.		
4	Metodologii educationale alternative in vederea dezvoltarii aplicatiilor software (creare echipe de lucru compatibile).		
5	UML. Diagrame de colaborare- conexiuni- legaturi -interactiuni: repetitive, conditionale, multithreading, preconditionii, sincrone, asincrone. Diagrame de stare. Diagrame de punere in functiune. Ciclul de viata al componentelor software in Metodologia OO (OOM). Studiu de caz.		
6	Metodologii de clasificare, structurale: SSADM-MERISE, obiectuale:OOD,OOT si formale. Interfatarea si implementarea. Principiul lui Parnas. Programarea UML: concepte de baza, evolutie. Diagrame de clasa si obiecte. Relatii intre clase, interfete si obiecte.		
7	Concepte privind evolutia programarii generice: functii/metode si clase template in C++. Evolutie si utilizare.		
8	Noi elemente introduse in C++0x/1y/2z. Biblioteca STL. Definire si utilizare.		
9	Noi facilitati oferite de biblioteca C++0x/1y/2z si STL		
10	Programarea generica in Java . Evolutie si utilizare.		
11	Programarea multithreading, paralela/multicore. Concepte, utilizare Java.		
12	Interfete grafice in programare. Design Pattern-uri. Exemplificare MVC.		
13	Testarea manuala si automata a aplicatiilor software. JUnit si alte facilitati de testare a aplicatiilor software.		
14	Cloud computing. Introducere, comparatie tehnologii existente		
8.2. Aplicatii (lucrări/proiect)			
	Metode de predare		
1	Principiile si obiectivele Inginerii Programarii reflectate in dezvoltarea aplicatiilor software	Expunere și teste	Retea de calculatoare dedicata
2	Elemente fundamentale de programare a aplicatiilor folosind un limbaj obiectual (C/C++, C#, Java). Elemente specifice avansate in dezvoltarea aplicatiilor obiectuale.		
3	Metodologii educationale alternative. Grupare in echipe de lucru folosind tipologii ale eneagramei si MBTI.		
4	Metodologii de gestiune/ proiectare a aplicatiilor software tinand cont de specificatii standard de firma.- Redactarea si evaluarea unui raport stiintific implicand dezvoltarea aplicatiilor software avansate. Definire teme pentru echipe.		
	Redactare articole in reviste si conferinte. Utilizarea diagramelor UML in procesul software. - Definire mecanisme creare echipe de lucru.		
5	Dezvoltarea de aplicatii software folosind C++0x/1y/2z si clase si functii/metode template in C++ (generice C++).		
	Dezvoltarea de aplicatii software folosind STL.		
	Etapa intermediara pre-evaluari echipe de lucru- stabilirea caiet sarcini final.		
6	Dezvoltarea de aplicatii software folosind folosind Java Generics. Multithreading in Java. Fork-join in Java pentru programarea paralela/multicore.		
	Testare aplicatii folosind JUnit. – Evaluari echipe de lucru- state of art, tehnologii utilizate		
7	Evaluare teme laborator.		
	Prezentarea unui raport stiintific pe o tematica software conform unor specificatii standard de firma.		
	Evaluari activitate echipe.		
Bibliografie			
1. Mircea-Florin Vaida, Cosmin Porumb, Radu Fotea, Florin Hurducas, Liviu Lazar, Java 2 Enterprise Edition (J2EE). Aplicatii multimedia, Editura Alabastra, 2003			
2. Cosmin Strilețchi, Mircea-F. Vaida, Ligia-D. Chiorean, Adriana Stan, Noțiuni esențiale și tehnologii specifice limbajului Java, Casa Cartii de Stiinta, 2018			
3. B. Stroustrup, The C++ programming language. Addison-Wesley, 2015			

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 Strilețchi, Kuderna-Iulian Bența, 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 Bența, Mircea-Florin Vaida, Petre Gavril Pop, Cosmin Strilețchi, C/C++ - Ghid teoretic si practic, Casa Cartii de Stiinta, Cluj-Napoca, 2016
8. Cosmin Strilețchi, Ligia-D. Chiorean, Mircea-F. Vaida, Adriana Stan, Stefan Dragos, Tehnologii Java orientate spre aplicații cross-platform, Casa Cartii de Stiinta, 2020
9. Materiale suport de curs/laborator de pe platforma Helios a colectivului de software, <https://helios.utcluj.ro/learn2code/index.php>

On-line Platforms:

- **Zoom:**
- **MirceaF Vaida: Personal Meeting ID 2412138787**

<https://zoom.us/j/2412138787>

pwd: 020418

<https://us04web.zoom.us/j/2412138787?pwd=dnNSaGJKY0VkYlJVVV1BuL1ZsYXpaUT09>

**Microsoft Teams:** Class: ESATC-ITM, Code: wmbt10y

## 9. Coroborarea continuturilor disciplinei cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor, profesionale si angajatori din domeniul aferent programului

Competentele achizitionate vor fi necesare angajatilor in urmatoarele ocupatii posibile conform COR (Inginer electronist, transporturi, telecomunicatii; Proiectant inginer electronist; Proiectant inginer de sisteme si calculatoare; Inginer proiectant comunicatii) sau in noi ocupatii propuse pentru a fi incluse in COR (Inginer suport vânzari; Dezvoltator de aplicatii multimedia; Inginer operare retea; Inginer testare sisteme de comunicatii; Manager proiect; Inginer de trafic; Consultant pentru sisteme de comunicatii)

## 10. Evaluare

Tip activitate	10.1	Criterii de evaluare	10.2	Metode de evaluare	10.3	Ponderea din nota finala
Curs		Test scris cu intrebari. Elaborare raport/lucrare stiintifica		Examen scris/oral (T=25%) + Activitate stiintifica in timpul semestrului (S=25%) $E = T + S$		E=50%
Aplicatii		Activitate practica in echipa dezvoltata la laborator si ACP		Activitate practica (P)		P=50%
10.4 Standard minim de performanta						
Răspuns corect la cel puțin un subiect de teorie, scrierea relațiilor de proiectare adecvate necesare pentru rezolvarea problemei și obținerea unei note minime de 5 în cadrul activităților de laborator.						

Data completarii  
26.09.2020

Titularul de disciplina  
Profesor  
Mircea-Florin Vaida, Ph.D.

Responsabil de curs  
Profesor  
Mircea-Florin Vaida, Ph.D.

Data avizarii in departament  
28.09.2020

Director departament  
Prof.dr.ing. Virgil Dobrota