

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	<b>Facultatea de Inginerie</b>
1.3 Departamentul	<b>Departamentul de Inginerie Electrică, Electronică și Calculatoare</b>
1.4 Domeniul de studii	<b>Calculatoare și Tehnologia Informației</b>
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	<b>Calculatoare</b>
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	43.10

### 2. Date despre disciplină

2.1 Denumirea disciplinei	Paradigme de programare		
2.2 Titularul de curs	Șef lucrări dr ing. Marieta Gâta, Email: Marieta.GATA@ieec.utcluj.ro		
2.3 Titularul activităților de seminar / laborator / proiect	Șef lucrări dr ing. Marieta Gâta, Email: Marieta.GATA@ieec.utcluj.ro		
2.4 Anul de studiu	III	2.5 Semestrul	II
		2.6 Tipul de evaluare	C
2.7 Regimul disciplinei	Categoría formativă		DD
	Opționalitate		DO

### 3. Timpul total estimat

3.1 Număr de ore pe săptămână	4	din care:	3.2 Curs	2	3.3 Seminar		3.3 Laborator	2	3.3 Proiect	
3.4 Număr de ore pe semestru	56	din care:	3.5 Curs	28	3.6 Seminar		3.6 Laborator	28	3.6 Proiect	
3.7 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										18
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										5
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										18
(d) Tutoriat										1
(e) Examinări										2
(f) Alte activități:										-
3.8 Total ore studiu individual (suma (3.7(a)...3.7(f)))					44					
3.9 Total ore pe semestru (3.4+3.8)					100					
3.10 Numărul de credite					4					

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

4.1 de curriculum	Programarea calculatoarelor și limbaje de programare (I și II), Programare orientată pe obiecte (I și II), Structuri de date, Proiectarea algoritmilor, Tehnici de programare fundamentale
4.2 de competențe	Cunoașterea unui limbaj de programare

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

5.1. de desfășurare a cursului	Sală de curs, videoproiector, tablă, laptop Pentru cursurile online: laptop/tabletă, internet
5.2. de desfășurare a seminarului / laboratorului / proiectului	Sală de laborator cu 25 de calculatoare, Pachetele software conform programei (Visual Studio .NET), internet, tablă. Pentru laboratoarele online: laptop/tabletă, internet

## 6. Competențele specifice acumulate

Competențe profesionale	<p>C1.1 Utilizarea adecvată în comunicarea profesională a conceptelor proprii calculabilității, complexității, paradigmelor de programare și modelării sistemelor de calcul și comunicații.</p> <p>C1.2 Utilizarea de teorii și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) pentru explicarea funcționării și structurii sistemelor hardware, software și de comunicații</p> <p>C2.3 Construirea unor componente hardware, software și de comunicații folosind metode de proiectare, limbaje, algoritmi, structuri de date, protocoale și tehnologii</p> <p>C1.4 Evaluarea formală a caracteristicilor funcționale și nefuncționale ale sistemelor de calcul C1.56</p> <p>Fundamentarea teoretică a caracteristicilor sistemelor proiectate</p> <p>C5.5 Realizarea unui proiect incluzând identificarea și analiza problemei, proiectarea, dezvoltarea și demonstrând o înțelegere a nevoii de calitate</p> <p>Modelarea unei probleme tipice ingineresti folosind aparatul formal caracteristic domeniului</p>
Competențe transversale	<p>CT1. Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei</p> <p>CT2. Identificarea, descrierea și derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă și descrierea clară și concisă, verbal și în scris, în limba română și într-o limbă de circulație internațională, a rezultatelor din domeniul de activitate</p> <p>CT3. Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională</p>

## 7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	<p>Prezentarea principalelor paradigme de programare care domină dezvoltarea sistemelor software la ora aceasta și deprinderea de abilități concrete de utilizare a lor.</p> <p>Dobândirea abilităților necesare pentru dezvoltarea aplicațiilor C# folosind o abordare vizuală, orientată pe obiecte, pornind de la cunoștințele de bază de programare prezentate în cursurile predate anterior.</p>
7.2 Obiectivele specifice	<p>Proiectarea și dezvoltarea în C# a aplicațiilor de complexitate redusă și medie.</p> <p>Punctarea diferențelor esențiale, a avantajelor și a dezavantajelor codului scris în C# față de cel scris în Java, C++.</p> <p>Consolidarea abilităților de testare și depanare a programelor scrise sub mediul de dezvoltare Visual Studio .NET</p>

## 8. Conținuturi

8.1 Curs	Nr. ore	Metode de predare	Observații
Curs 1 Paradigme în programare. Introducere. Programarea orientată pe obiecte. Programarea orientată pe evenimente. Programarea folosind reprezentări abstracte.	2	Pentru cursurile online: Prelegere cu suport pe laptop și tabletă în Meeting din Microsoft Teams	
Curs 2 Programarea vizuală. Programarea funcțională. Programarea logică.	2		
Curs 3 Interfețe Windows în C#.	2		

Realizarea primei aplicații elementare în C#. Noțiuni de C#. Spații de nume. Comentarii.			
Curs 4 Tipuri de date. Operatori. Instrucțiuni de control	2		
Curs 5 Tablouri. Strunguri.	2		
Curs 6 Clase. Obiecte. Metode.	2		
Curs 7 Clase și metode prezentare în detaliu. Supraîncărcarea metodelor. Supraîncărcarea constructorilor. Supraîncărcarea operatorilor	2		
Curs 8 Operații de intrare/ieșire. Stream-uri. Clasele File și Directory. Citirea/scrierea datelor din fișier. Clasele StreamReader și StreamWriter	2		
Curs 9 Colecții generice. Clasele spațiului de nume. Structurile spațiului de nume. Interfețele spațiului de nume. Clasa List. Clasa Directory. Colecții sortate. Cozi și stive	2		
Curs 10 Colecții. Comparații. Conversii. Clasa ArrayList. Clasa DictionaryBase. Sortarea colecțiilor.	2		
Curs 11 Aplicații C# Windows Form. Adăugarea de controale pe o formă C#. Proprietățile unui control C#. Adăugarea de cod C# la un buton	2		
Curs 12 Primele controale. Butonul. RadioButton. CheckBox. GroupBox. Label. LinkLabel. TextBox. Evenimente. Enter. GetFocus. Leave. Validating. Validated. LostFocus. KeyDown. KeyPress. KeyUp. Change. Prima interfața grafică.	2		
Curs 13 Indexări. Proprietăți. Structuri. Enumerări.	2		
Curs 14 Interfețe. Delegări. Evenimente.	2		
<b>Bibliografie</b> 1. <i>Functional Programming. Classic Programming Techniques for Modern Projects</i> , O. Sturmm, Wiley Publishing, 2011, USA 2. <i>Microsoft Visual C# 2013, Step by Step</i> , John Sharp, Microsoft, 2015, USA 3. <i>Learn ASP.NET 4.5, C# and Visual Studio 2012, Essential Skills with the Smart Method</i> , Simion Smart, The Smart Method, 2014, USA 4. <i>Learn ASP.NET 4.5, C# and Visual Studio 2012, Expert Skills with the Smart Method</i> , Simion Smart, The Smart Method, 2014, USA 5. <i>NET Book Zero</i> , Charles Petzold, <a href="http://www.charlespetzold.com/dotnet">http://www.charlespetzold.com/dotnet</a> , 2007 6. <i>Microsoft Visual C# 2010. An Introduction to Object-Oriented Programming</i> , Joyce Farrel, Course Technology, Cengage Learning, Boston, USA, 2011 7. <i>Pro C# 5.0 and the .NET 4.5 Framework</i> , Adrew Troelsen, Apress, 2012 8. <i>Beginning C# 5.0 Databases</i> , Vidya Vrat Agarwal, Apress, 2012			
8.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
Laborator 1 Prima aplicație de tip consolă	2	Modelare, problematizare, studiu de caz.	
Laborator 2 Operații de citire/afișare. Instrucțiuni repetitive.	2		

Laborator 3 Tablouri unidimensionale. Tablouri bidimensionale. Structuri de control	2	Pentru laboratoarele online: folosirea facilităților oferite de Meeting în Microsoft Teams, Office 365, Whiteboard, OneNote, Forms, SharePoint.			
Laborator 4 Lucrul cu set, get, constructori, destructori.	2				
Laborator 5 Transmiterea parametrilor. Funcții cu număr variabil de parametri.	2				
Laborator 6 Clasa DirectoryInfo. Lucrul cu fișiere, directoare. Serializarea și deserializarea unui obiect.	2				
Laborator 7 Liste. Stive. Cozi.	2				
Laborator 8 List. ArrayList. Dictionary.	2				
Laborator 9 Controale. Evenimente. Primul exemplu.	2				
Laborator 10 Aplicații grafice ce folosesc controale și evenimente.	2				
Laborator 11 Lucrul cu baze de date în C#.	2				
Laborator 12 Lucrul cu baze de date relaționale în C#.	2				
Laborator 13 Proiect individual	2				
Laborator 14 Proiect individual	2				
<b>Bibliografie</b>					
1. <i>Functional Programming. Classic Programming Techniques for Modern Projects</i> , O. Sturmm, Wiley Publishing, 2011, USA					
2. <i>Microsoft Visual C# 2013, Step by Step</i> , John Sharp, Microsoft, 2015, USA					
3. <i>Learn ASP.NET 4.5, C# and Visual Studio 2012, Essential Skills with the Smart Method</i> , Simion Smart, The Smart Method, 2014, USA					
4. <i>Learn ASP.NET 4.5, C# and Visual Studio 2012, Expert Skills with the Smart Method</i> , Simion Smart, The Smart Method, 2014, USA					
5. <i>NET Book Zero</i> , Charles Petzold, <a href="http://www.charlespetzold.com/dotnet">http://www.charlespetzold.com/dotnet</a> , 2007					
6. <i>Microsoft Visual C# 2010. An Introduction to Object-Oriented Programming</i> , Joyce Farrel, Course Technology, Cengage Learning, Boston, USA, 2011					
7. <i>Pro C# 5.0 and the .NET 4.5 Framework</i> , Adrew Troelsen, Apress, 2012					
8. <i>Beginning C# 5.0 Databases</i> , Vidya Vrat Agarwal, Apress, 2012					

### 9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

Cursul oferă ajutor pentru materiile din profilul software al specializării Calculatoare.

Cursul creează legături directe/imediate cu următoarele discipline deja predate: Programarea calculatoarelor și limbaje de programare (I și II), Programare orientată pe obiecte (I și II), Structuri de date, Proiectarea algoritmilor, Tehnici de programare fundamentale.

Utilitatea pentru piața muncii: aplicațiile de laborator sunt bazate pe limbajul C# și tehnologii emergente acestuia, oferă cunoștințe în acesta direcție.

Cunoștințele de programare .NET (C#) sunt importante pentru cei care doresc să se angajeze ca programatori, deoarece mulți dintre angajatorii reprezentativi din domeniul aferent programului de studiu solicită cunoștințe de programare .NET.

### 10. Evaluare (prezența fizică / online)

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
----------------	---------------------------	-------------------------	------------------------------

10.4 Curs	Cunoștințe teoretice însușite (cantitatea, corectitudinea, acuratețea)	Evaluarea finală Rezolvarea unui test grilă și a unor probleme (evaluare scrisă)	50% (minim 5)
10.5 Seminar/Laborator/Proiect	Frecvența/relevanța intervențiilor. Cunoașterea modului de utilizare a instrumentelor specifice. Evaluarea unor instrumente sau realizări, prelucrarea și interpretarea unor rezultate.	Portofoliu de lucrări (rezolvarea a 60 de probleme) care sunt verificate la finalul fiecărei ore de laborator (cate 5 probleme în fiecare laborator)	40% (minim 5)
	Calitatea proiectului realizat, corectitudinea documentației proiectului, justificarea soluțiilor alese.	Proiect -Prezentarea proiectului (care consta dintr-o aplicație practică grafică ce folosește controale, evenimente și elemente de programare) -Autoevaluarea critică a proiectului	10% (minim 5)
10.6 Standard minim de performanță			
<p>Cunoașterea noțiunilor de bază prezentate la curs și obținerea a notei minim 5 la evaluarea finală. Admiterea la examen presupune prezența la laboratoare și predarea laboratoarelor și a proiectului.</p> <p>Programele propuse la laborator trebuie să ruleze fără erori de compilare și să rezolve toate cerințele impuse în probleme. În primele 12 laboratoare sunt propuse un număr de 5 până la 10 probleme din care studentul își va alege spre rezolvare un număr de minim 5 probleme (în total, la finalul orelor de laborator, studentul va trebui să aibă rezolvate un număr de 60 de probleme pentru nota 10, pentru nota 5 este necesar să fie rezolvate 30 de probleme). În următoarele două laboratoare studentul va lucra la proiectul pe care și l-a ales (vor exista un număr de teme propuse de către profesor, fiecare student va avea o altă temă aleasă din cele propuse de către profesor sau propusă de către student).</p> <p>Proiectarea, testarea și executarea unei aplicații de complexitate medie (proiect). Proiectul va conține cel puțin cinci controale și cel puțin două elemente de programare. Pentru nota 10 programul va trebui să ruleze fără erori, să rezolve toate cerințele impuse și să fie de dificultate înaltă. Pentru nota 5 proiectul va trebui să ruleze fără erori, să rezolve cel puțin 50% din cerințele impuse și să fie de dificultate joasă.</p> <p>Dobândirea unei gândiri analitice și a unor abilități de generalizare independente de limbajul de programare folosit, care sunt o componentă fundamentală a proiectării oricărei aplicații software complexe.</p> <p>Dobândirea unor abilități practice legate de particularizarea paradigmatelor de programare folosind un limbaj de programare.</p>			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
	Curs	Sef lucrări dr. ing. Marieta GÂTA	
	Aplicații	Şef lucrări dr. ing. Marieta GÂTA	

Data avizării în Consiliul Departamentului _____	Director Departament, Şef lucrări dr. ing. Claudiu Ionel Lung
Data aprobării în Consiliul Facultății, _____	Decan, Conf. dr. ing. Dinu DARABA