FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea [1] / Departamentul[2]	Robotică, Inteligență Artificială și Sisteme Autonome / Calculatoare și Tehnologia Informației
1.3 Domeniul de studii (denumire/cod[3])	Informatică / 10
1.4 Ciclul de studii	Master
1.5 Programul de studii (denumire/cod/calificarea)	Tehnologii Informatice / 10 / Tehnologii Informatice

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă[4]			Tehnici avansate de compilare / DS				
2.1b Denumirea disciplinei în limba engleză			Advances Compiling Techniques				
2.2 Titularul activităţilor de curs			Răzvan-Mihai ACIU				
2.3 Titularul activităţilor aplicative[5]			Paul ZINCA				
2.4 Anul de studiu[6]	1	**2.5** Semestrul	1	**2.6** Tipul de evaluare	E	**2.7** Regimul disciplinei[7]	DI

3. Timp total estimat - ore pe semestru (activități directe (asistate integral), activități asistate parțial și activități neasistate[8])

3.1 Număr de ore asistate integral/săptămână	3.5, din care:	ore curs	1.5	ore seminar/laborator/proiect			2
3.1* Număr total de ore asistate integral/sem.	49, din care:	ore curs	21	ore seminar/laborator/proiect			28
3.2 Număr total de ore desfășurate on-line asistate integral/sem.	, din care:	ore curs		ore seminar/laborator/proiect			
3.3 Număr de ore asistate parțial/saptămână	, din care:	ore proiect, cercetare		ore practică		ore elaborare lucrare de disertație	
3.3* Număr total de ore asistate parțial/ semestru	, din care:	ore proiect cercetare		ore practică		ore elaborare lucrare de disertație	
3.4 Număr de ore activități neasistate/ săptămână	7.2, din care:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					2.4
		ore studiu individual după manual, suport de curs, bibliografie și notițe					2.4
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri					2.4
3.4* Număr total de ore activități neasistate/ semestru	101, din care:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					34
		ore studiu individual după manual, suport de curs, bibliografie și notițe					34
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri					33
3.5 Total ore/săptămână[9]	10.7						
3.5* Total ore/semestru	150						
3.6 Număr de credite	6						

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	• Programarea calculatoarelor, Structuri de date și algoritmi
4.2 de rezultate ale învățării	• Cunoștințe de bază de programare, tehnici de programare și algoritmi

5. Condiţii (acolo unde este cazul)

5.1 de desfăşurare a cursului	• O sală de min. 70 locuri, videoproiector, tablă
5.2 de desfăşurare a activităţilor practice	• Laborator cu min. 15 stații și software pentru programare și acces la Campus Virtual

6. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	• C1. Studentul/absolventul alege, descrie, analizează și explică paradigmele moderne de programare, inclusiv programarea funcțională, orientată pe obiect și paralelă, utilizând limbaje și framework-uri actuale.
Abilități	• A1.1 Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software complexe ce utilizează eficient resursele hardware și software. • A1.2 . Studentul/absolventul caută, descoperă și înțelege noile tehnologii de dezvoltare a sistemelor informatice pe măsura ce acestea apar.
Responsabilitate și autonomie	• RA1.1 Studentul/absolventul produce software și îl adaptează continuu la noile tehnologii și cerințe de piață. • RA1.2 Studentul/absolventul asimilează în procesul de dezvoltare software noi limbaje de programare, tehnologii, biblioteci, cadre de dezvoltare aplicații.

7. Obiectivele disciplinei (asociate rezultatelor învățării specifice acumulate)

- Completarea cunoștințelor de limbaje și tehnici de compilare obținute în ciclul de licență
- Însușirea unor aspecte de proiectare și programare avansată aplicate în traducerea/translatarea programelor Perfecționarea abilităților de comunicare, lucru în echipă, management de sarcini complexe, cooperare interdisciplinară Eficientizarea procesului de traducere/translatare

8. Conţinuturi

8.1 Curs	Număr de ore	Din care on-line	Metode de predare
Fazele unui compilator; Specificarea gramaticilor; Analizorul lexical; Analizorul sintactic; Analiza semantică; Optimizarea de cod; Generarea de cod	3		Prezentare PPT, explicații pe tablă, discuții, întrebări și răspunsuri
Integrarea unui limbaj de programare într-o aplicație gazdă; Implementarea extensiilor; Folosirea limbajelor de programare dinamice pentru limbajul integrat;	3		
Generare de cod folosind LLVM (Low Level Virtual Machine); Compilare la execuție (JIT-Just in Time Compiling)	3		
Metaprogramare; Folosirea mai multor etape de compilare; Macrouri; Tipuri de date și funcții generice; Cod executat la compilare; Folosirea unor unelte externe la compilare; Metadata; Reflexie	6		
Compilare adaptată utilizării; Compilare pe calculatorul gazdă (AOT-Ahead of time); Optimizare adaptivă; Compilare la execuție	3		
Interfețe; Implementarea polimorfismului; Comunicare inter-aplicații folosind interfețe; Programare COM (Component Object Model)	3		

	Bibliografie[10] 1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman: Compilers. Principles, Techniques and Tools, Second edition, Addison-Wesley Publishing Company, 2007. 2. Răzvan-Mihai Aciu: Limbaje formale și tehnici de compilare – curs și laborator online pentru programul de studii Informatică, https://cv.upt.ro/course/view.php?id=1896 3. Răzvan-Mihai Aciu: Tehnici avansate de compilare, curs și proiect online, https://cv.upt.ro/course/view.php?id=191 4. Horia Ciocârlie, Rodica Ciocârlie, Limbaje formale și translatoare, Editura Eurostampa, 2017. 5. Terence Parr: The Definitive ANTLR4 Reference, 2012 6. Cooper Keith, Linda Torczon: Engineering a Compiler, 3rd edition, 2022

8.2 Activităţi aplicative[11]	Număr de ore	Din care on-line	Metode de predare
Folosirea unor unelte software (ANTLR) pentru generarea analizoarelor lexicale și sintactice - Parcurgerea Arborelui sintactic generat din gramatica dată folosind tiparul Visitor - Implementarea unor acțiuni semantice care preiau informațiile din codul sursă și le procesează - Managementul proiectelor complexe folosind CMake	6		Prezentarea temelor pe videoproiector, explicații pe tablă, discuții, întrebări și răspunsuri. Lucru individual cu studenții
Completarea funcţiilor din biblioteca unui compilator (runtime), astfel încât compilatorul să poată executa un program dat - Prezentarea unui compilator de JavaScript care generează cod C - Familiarizarea cu unele mecanisme de implementare a limbajelor dinamice și a bibliotecii acestora - Implementarea funcțiilor care lipsesc din biblioteca compilatorului prezentat - Testarea implementărilor prin execuția unui program de test	8		
Folosirea LLVM pentru codul generat de compilatorul de JavaScript și pentru execuția acestuia în regim JIT - Prezentarea unui compilator implementat parțial, care generează cod LLVM pentru un program sursă JavaScript - Familiarizarea cu felul în care este generat cod pentru LLVM - Implementarea generării de cod pentru construcțiile de limbaj care încă nu au fost implementate în compilator - Testarea codului generat prin afișarea sa și prin execuția sa în regim JIT	14		

	Bibliografie[12] . 1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman: Compilers. Principles, Techniques and Tools, Second edition, Addison-Wesley Publishing Company, 2007. 2. Răzvan-Mihai Aciu: Limbaje formale și tehnici de compilare – curs și laborator online pentru programul de studii Informatică, https://cv.upt.ro/course/view.php?id=1896 3. Răzvan-Mihai Aciu: Tehnici avansate de compilare, curs și proiect online, https://cv.upt.ro/course/view.php?id=191 4. Horia Ciocârlie, Rodica Ciocârlie, Limbaje formale și translatoare, Editura Eurostampa, 2017. 5. Terence Parr: The Definitive ANTLR4 Reference, 2012 6. Cooper Keith, Linda Torczon: Engineering a Compiler, 3rd edition, 2022

9. Evaluare

Tip activitate	**9.1** Criterii de evaluare[13]	**9.2** Metode de evaluare	**9.3** Pondere din nota finală
9.4 Curs	Întrebări din materia parcursă	Examen scris	70 %
9.5 Activităţi aplicative	**S:**		
	L:		
	P: Fiecare student completează implementarea unui proiect implementat parțial, pentru fiecare dintre cele 3 activități aplicative	Studentul prezintă practic proiectele completate de el. Cadrul didactic pune întrebări relativ la proiecte	25 %
	Pr:		
	Tc-R[14]:		5 %
9.6 Standard minim de performanţă (volumul de cunoştinţe minim necesar pentru promovarea disciplinei şi modul în care se verifică stăpânirea lui)[15]			
• Funcționarea programelor corespunzătoare proiectelor – se verifică pe calculator Abordarea și rezolvarea a min. 50 % din cerințele examenului scris – nota 5			

Data completării	**Titular de curs (semnătura)**	**Titular activităţi aplicative (semnătura)**
21.09.2026	Răzvan-Mihai ACIU	Paul ZINCA

Director de departament (semnătura)	**Data avizării în Consiliul Facultăţii[16]**	**Decan (semnătura)**
Prof.univ.dr.ing. Mihai MICEA		ș.l.univ.dr.ing. Robert KRISTOF