

FIȘA DISCIPLINEI¹

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea „Politehnica” din Timișoara
1.2 Facultatea ² / Departamentul ³	Automatică și Calculatoare /
1.3 Catedra	-
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență - Învățământ la distanță
1.6 Programul de studii / Calificarea	Informatică / Informatician

2. Date despre disciplină

2.1 Denumirea disciplinei	Tehnici de Programare						
2.2 Titularul activităților de curs	Asist. Drd. Ing. Mirella Amelia Mioc						
2.3 Titularul (titularii) activităților de seminar/laborator (AA)	Asist. Drd. Ing. Mirella Amelia Mioc						
2.4. Categoria formativă (fundamentală, complementară, domeniu, specialitate)	fundamentală						
2.5. Anul de studiu	1	2.5 Semestrul	2	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	impusa

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	4	din care:3.2 SI (studiu individual-curs)	2	3.3 TC (teme de casă)		3.4. AT (activități tutoriate)		3.5 AA (seminar/laborator)	2
3.6 Total ore din planul de învățământ	56	din care:3.7 SI (studiu individual-curs)	28	3.8 TC (teme de casă)		3.9. AT (activități tutoriate)		3.10 AA (seminar/laborator)	28
Distribuția fondului de timp									ore
Studiul după manual, suport de curs, bibliografie și notițe									26
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren									14
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri									12
Tutoriat									12
Examinări									6
Alte activități									
3.11 Total ore studiu individual		74							
3.12 Total ore pe semestru		130							
3.13 Numărul de credite		5							

4. Precondiții și recomandări (acolo unde este cazul)

4.1 precondiții de curriculum	<ul style="list-style-type: none"> Programarea calculatoarelor
4.2 precondiții de competențe	<ul style="list-style-type: none"> Cunostinte de matematica Cunostinte de programare
4.3. recomandări de discipline anterioare	<ul style="list-style-type: none"> Utilizarea calculatoarelor Programarea calculatoarelor

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none"> Sală mare, Materiale suport: laptop, proiector, tablă
5.2 de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> Laborator cu 15-20 calculatoare – Mediu de programare pentru limbajul C, tablă

¹ Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 (Anexa3);

² Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina;

³ Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului;

6. Competențe specifice acumulate

Competențe profesionale ⁴	<ul style="list-style-type: none"> • Programarea în limbaje de nivel înalt • Proiectarea și gestiunea bazelor de date • Proiectarea și administrarea rețelelor de calculatoare
Competențe transversale	Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională

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

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Completarea cunoștințelor de programare dobândite la cursul de inițiere în programare cu alte facilități ale limbajului C și cu elemente de tehnici de programare specifice unui curs de programare avansată
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Dobândirea tehnicilor de programare avansată • Proiectarea și implementarea unor programe C de complexitate medie și mare • Însușirea unui stil de programare corect

8. Conținuturi

8.1 Curs (SI – studiu individual)	Număr de ore	Metode de predare
1. Funcții definite de utilizator. Argumente 6.1 Bazele definirii și utilizării funcțiilor 6.2 Funcții ce returnează altfel de valori decât întregi 6.3 Variabile externe. Reguli pentru domeniile de vizibilitate 6.4 Variabile statice. Variabile registru 6.5 Structura de bloc. Inițializarea variabilelor	1	Prelegere susținută de prezentări PPT, conversații, explicații, exemplificări
2. Tablouri, șiruri de caractere 7.1 Tablouri unidimensionale 7.2 Tablouri bidimensionale 7.3 Tablouri de caractere	1	
3. Pointeri. Alocarea dinamică a memoriei 8.1 Pointeri și adrese. 8.2 Pointerii și tablourile 8.3 Aritmetica adreselor 8.3 Pointeri spre caractere 8.4 Descrierea și implementarea unor funcții de bibliotecă pentru prelucrarea șirurilor de caractere 8.5 Alocarea dinamică de memorie 8.6 Tablouri de pointeri. Tablouri multidimensionale	1	
4. Elemente de programare avansată în limbajul C 1.1 Structuri și uniuni. Alocare dinamică implicând structuri 1.2 Câmpuri de biți 1.3 Operații pe biți. Tabelul complet al precedentei operatorilor 1.4 Argumente în linia de comandă 1.5 Argumente de tip pointer 1.6 Pointeri la funcții 1.7 Preprocesorul 1.8 Funcții cu număr variabil de argumente	1	
5. Fișiere 2.1 Caracteristici ale fișierelor 2.2 Operații de bază asupra fișierelor 2.3 Prelucrarea fișierelor text 2.4 Prelucrarea fișierelor binare 2.5 Funcții speciale pentru tratarea erorilor 2.6 Actualizarea fișierelor în acces direct 2.7 Redirecționarea intrării și a ieșirii standard	1	

⁴ Aspectul competențelor profesionale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competențele care sunt precizate în Registrul Național al Calificărilor din Învățământul Superior RNCIS (http://www.rncis.ro/portal/page?_pageid=117,70218&_dad=portal&_schema=PORTAL) pentru domeniul de studiu de la pct. 1.4, programul de studii de la pct. 1.6 din această fișă și materia în cauză

6. Proiectarea și dezvoltarea sistematică a programelor de mari dimensiuni 3.1 Programarea calculatoarelor – de la teorie la practică 3.2 Stilul de programare 3.3 Etica programării 3.4 Metoda detaliilor în pași succesivi 3.5 Exemplu de program 3.6 Concluzii	1	
7. Recursivitatea în C 4.1 Conceptul de recursivitate 4.2 Recursivitate directă 4.3 Înregistrarea de activare 4.4 Relația dintre recursivitate și iterație 4.5 Exemple de programe recursive	1	
8. Metode generale de proiectare a algoritmilor și programelor 5.1 Metoda Divide and Conquer (Tehnică divizării) 5.2 Metoda Backtracking (Algoritmi cu revenire)	1	
Bibliografie 1. Brian W. Kernighan, Denis Ritchie, <i>Limbajul C</i> , Ed. Teora, 2003 2. Horia Ciocârlie, Rodica Ciocârlie, <i>Tehnici de programare și structuri de date</i> , Ed. Eurostampa, 2012 3. Dr. Kris Jamsa & Lars Klander, <i>Totul despre C și C++. Manualul fundamental de programare în C și C++</i> , Ed. Teora, 2009 4. Liviu Negrescu, <i>Limbajele C și C++ pentru începători, vol. I, Limbajul C</i> , Ed. MicroInformatica, 1994 5. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, <i>Introducere în algoritmi</i> , Ed. Computer LIBRIS Agora, (MIT Press, 1990) 6. Donald E. Knuth, <i>Arta programării calculatoarelor (3 volume), Algoritmi fundamentali, Algoritmi seminumerici, Sortare și căutare</i> , Ed. Teora, 2000, (Addison Wesley, 1962) 7. Robert Sedgewick, <i>Algorithms</i> , Ed. Addison Wesley, 1983. 8. V.Iorga, P.Chiriță, C. Stratan, C.Opincaru, <i>Programare în C/C++. Culegere de probleme</i> , Ed. Niculescu, 2003 9. Vladimir Crețu, <i>Structuri de date și algoritmi, vol. I, Structuri de date fundamentale</i> , Ed. Orizonturi Universitare, 2000 10. Ioana Șora, Doru Todinca, <i>Introducere în programarea calculatoarelor</i> , Ed. Politehnica, 2004		
8.2 Seminar/laborator	Număr de ore	Metode de predare
1. Funcții definite de utilizator		Expunere temă, discuții, întrebări, rezolvare pe calculator, a 1-2 probleme.
2. Tablouri, șiruri de caractere		
3. Pointeri. Alocarea dinamică a memoriei		
4. Operații pe biți		
5. Structuri		
6. Uniuni și câmpuri pe biți		
7. Argumente în linia de comandă. Pointeri la funcții		
8. Proiectarea și dezvoltarea sistematică a programelor de mari dimensiuni		
9. Prelucrări de fișiere		
10. Aplicații ale metodei Divide and Conquer		
11. Algoritmi cu revenire (Backtracking)		
12. Recuperări		

9. Corelarea conținutului disciplinei cu cerințele specialiștilor din domeniu și cu așteptările angajatorilor reprezentativi

<ul style="list-style-type: none"> Cunoștințele de programare sunt importante pentru toate materiile cu specific software care fac parte din planul de învățământ al specializării: Tehnici de programare, Programare orientată pe obiecte, Structuri de date și algoritmi, Proiectarea și analiza algoritmilor, Fundamentele ingineriei software, Sisteme de operare ș. a. Majoritatea angajatorilor reprezentativi din domeniul aferent programului solicită atât cunoștințe de programare în general cât și cunoașterea limbajului de programare C.
--

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4. Evaluare finală	Rezolvarea a două probleme din prima parte și respectiv din a doua parte a materiei	Examen scris	60 %
10.5. AA – activități asistate (seminar/laborator)		Nu e cazul	
10.6. Teste pe parcursul semestrului		Nu e cazul	
10.7. Lucrări de verificare	Teme de casă	Prezentarea rezolvărilor, răspunsuri la întrebări	40 %
10.8 Standard minim de performanță (volumul de cunoștințe minim necesar pentru promovarea disciplinei și modul în care se			

verifică stăpânirea lui)

- Proiectarea, testarea și executarea unui program de complexitate medie din conținutul materiei
- Cele două programe de pe biletul de examen trebuie să fie funcționale și să rezolve minimul de cerințe solicitat

11. Compatibilitate internațională

- Carnegie Mellon University
- University of Michigan
- University of Wolverhampton

Data
completării
16.03.2015

Semnătura titularului de curs

Semnătura titularilor de seminar

.....

.....

Data avizării în departament

16.03.2015

Semnătura directorului de departament

.....