

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 / Automatică și Informatică Aplicată
1.3 Catedra	-
1.4 Domeniul de studii	Ingineria sistemelor
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Automatică și Informatică aplicată / Inginer

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare orientată pe obiecte						
2.2 Titularul activităților de curs	Prof. dr. ing. Vasile Stoicu-Tivadar						
2.3 Titularul activităților de seminar	Asist. dr. ing. Mihaela Vida, asist. dr. ing. Norbert Gal						
2.4 Anul de studiu	2	2.5 Semestrul	1	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie

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

3.1 Număr de ore pe săptămână	4,5	din care:3.2 curs	2,5	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	115	din care:3.5 curs	35	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					12
Tutoriat					7
Examinări					3
Alte activități					
3.7 Total ore studiu individual	52				
3.8 Total ore pe semestru	125				
3.9 Numărul de credite	5				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> Programarea calculatoarelor
4.2 de competențe	<ul style="list-style-type: none"> Cunoștințe elementare de programare (limbajul C)

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none"> Sală mare, Materiale suport: laptop, video-proiector, tablă, eventual sistem de sonorizare
5.2 de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> Laborator cu 17-25 calculatoare – Mediu de programare pentru limbajul C++ (Visual Studio), tablă

6. Competențe specifice acumulate

Competențe profesionale ⁴	<ul style="list-style-type: none"> Operarea cu concepte fundamentale din știința calculatoarelor, tehnologia informației și comunicațiilor. Proiectarea, implementarea, testarea, utilizarea și mentenanța sistemelor cu echipamente de uz general și dedicat, inclusiv rețele de calculatoare, pentru aplicații de automatică și informatică aplicată. Dezvoltarea de aplicații și implementarea algoritmilor și structurilor de conducere automată, utilizând principii
--------------------------------------	--

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;

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

	de management de proiect, medii de programare și tehnologii bazate pe microcontrolere, procesoare de semnal, automate programabile, sisteme încorporate
Competențe transversale	<ul style="list-style-type: none"> • Aplicarea, în contextul respectării legislației, a drepturilor de proprietate intelectuală (inclusiv transfer tehnologic), a metodologiei de certificare a produselor, a principiilor, normelor și valorilor codului de etică profesională în cadrul propriei strategii de muncă riguroasă, eficientă și responsabilă • Identificarea rolurilor și responsabilităților într-o echipă plurispecializată, luarea deciziilor și atribuirea de sarcini, cu aplicarea de tehnici de relaționare și muncă eficientă în cadrul echipei. • Identificarea oportunităților de formare continuă și valorificarea eficientă a resurselor și tehnicilor de învățare pentru propria dezvoltare.

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

7.1 Obiectivul general al disciplinei	Crearea deprinderilor teoretice și practice necesare analizei, proiectării, codificării și testării aplicațiilor de complexitate medie în limbajul C++.
7.2 Obiectivele specifice	<p>Transmiterea cunoștințelor de bază despre paradigma programării orientate pe obiecte.</p> <p>Crearea deprinderilor teoretice și practice pentru realizarea aplicațiilor de complexitate medie în limbajul C++, inclusiv utilizarea tipare de programare.</p> <p>Înșușirea unor elemente de analiză și proiectare orientate pe obiecte, inclusiv noțiuni elementare de UML și șabloane de proiectare.</p> <p>Înțelegerea modului în care cunoștințele despre programarea orientată pe obiecte sînt utilizabile în mediile de programare moderne; studiu de caz pentru programarea Windows.</p>

8. Conținuturi

8.1 Curs	Număr de ore	Metode de predare
1. Paradigma programării orientate pe obiecte 1.1. Necesitatea unei noi paradigme 1.2. Caracteristici 1.3. Conținutul cursului. Obiective, mod de desfășurare, condiții de examinare, bibliografie	3	Prelegere susținută de prezentări PPT/Prezi conversații, explicații, exemplificări, studii de caz
2. Limbajul C++. Generalități 2.1 Programarea în stilul C++ O 2.2. Operatori de I/E, C 2.3. Clase 2.4. Supraîncărcare 2.5. Moștenire 2.6 Constructori/destructori).	4	
3. Clase și obiecte 3.1. Clase 3.2. Structuri, uniuni, 3.3. Funcții prietene 3.4. Funcții inline 3.5. Constructori 3.6. Membri statici 3.7. Manipularea obiectelor	3	
4. Matrice, pointeri și referințe 4.1. Matrice de obiecte 4.2. Pointeri către obiecte 4.3. Referințe 4.4. Alocarea dinamică	3	
5. Supraîncărcarea 5.1. Supraîncărcarea funcțiilor 5.2. Supraîncărcarea operatorilor	4	
6. Moștenirea. 6.1. Controlul accesului la clasa de bază 6.2. Moștenirea din clase de bază multiple 6.3. Constructori, destructori și moștenire 6.4. Declarații de acces 6.5. Clase virtuale	2	
7. Operații de I/E în C++ 7.1. Stream-uri	3	

7.2. Ierarhia de clase 7.3. Variabila de stare 7.4. Formatarea datelor 7.5. Supraîncărcarea operatorilor << și >> 7.6. Lucrul cu fișiere 7.7. Formatarea în memorie		
8. Polimorfismul cu funcții virtuale 8.1. Proprietăți 8.2. Utilizare	2	
9. Tipare 9.1. Utilizarea tiparelor 9.2. Containere, iteratori, algoritmi în Standard Template Library 9.3. Clase colecție în MFC	2	
10. Studiu de caz de ierarhie de clase: MFC 10.1. Introducere în programarea Windows. stilul windows, funcții API, bucle de mesaje. 10.2. Prezentarea generală a ierarhiei MFC. 10.3. Clasa CObject. Serializare și facilități de depanare. 10.4. Clasa CWnd și interfețele utilizator. 10.5. Arhitectura Document-View. 10.6. Grafica cu MFC.	3	
11. Dezvoltarea orientată pe obiecte a produselor-program 11.1. Modelarea orientată pe obiecte 11.2. Metode de dezvoltare orientate pe obiecte 11.3. Etape de dezvoltare: identificarea obiectelor, precizarea structurii statice, precizarea comportamentului dinamic, evaluarea modelului, implementarea claselor, nivele de abstractizare	2	
12. UML și șabloane de proiectare 12.1. Introducere în UML 12.2. Analiza cerințelor și Cazuri de utilizare 12.3. Definirea structurii obiectelor și diagramele de clasă, diagrame de secvență și de stare. 12.3. Definirea comportamentului obiectelor și diagramele de stare și de secvență. 12.4. Șabloane de proiectare. Definiție și clasificare. 12.5. Exemple de șabloane de proiectare.	4	
Bibliografie 1. Stoicu-Tivadar, V., <i>“Programare orientata pe obiecte”</i> , ediția a 2-a, Editura “Orizonturi Universitare”, Timisoara, 2010. 2. Schildt, H. , <i>“C++ manual complet”</i> , Editura Teora, București, 1998. 3. Jamsa, K., Klander, L., <i>Totul despre C și C++</i> . Manualul fundamental de programare în C și C++, Ed. Teora, 2001.		
8.2 Seminar/laborator	Număr de ore	Metode de predare
1. Completări aduse de C++ față de limbajul C.	2	Expunere temă, discuții, întrebări, rezolvare pe calculator a 1-3 probleme. Teste periodice tip grilă.
2. Încapsularea prin intermediul claselor.	2	
3. Pointeri la metode.	2	
4. Constructori, destructori.	2	
5. Funcții și clase friend.	2	
6. Derivarea claselor.	2	
7. Metode virtuale.	2	
8. Utilizarea listelor eterogene în POO.	3	
9. Moștenirea multiplă.	2	
10. Crearea unei ierarhii de clase pentru un editor grafic sau alte proiecte distribuite de asistenți (proiect cu susținere)	7	
11. Recuperări laboratoare	2	
Bibliografie 1. Stoicu-Tivadar, V., <i>“Programare orientată pe obiecte”</i> , ediția a 2-a, Editura “Orizonturi Universitare”, Timișoara, 2010. 2. Gal, N., Vida, M., Chirila, O., <i>Îndrumător de laborator pentru POO</i> , format electronic (disponibil in situ), 2013. 3. Schildt, H. , <i>“C++ manual complet”</i> , Editura Teora, București, 1998. 4. V.Iorğa, P.Chiriță, C. Stratan, C.Opincaru, <i>Programare în C/C++</i> . <i>Culegere de probleme</i> , Ed. Niculescu, 2003		

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

- Cunoștințele de programare orientate pe obiecte sunt importante pentru toate materiile aval cu specific software care fac parte din planul de învățământ al specializării: Medii și tehnologii de programare, Ingineria programării, discipline opționale diverse (Modelare software, Programare JAVA, Tehnologii.NET, etc.).
- Majoritatea angajatorilor reprezentativi din domeniul aferent programului solicită atât cunoștințe de programare în general cât și cunoașterea programării orientate pe obiecte (C++, JAVA, C#, care pot fi ușor învățate după parcurgerea cursului de POO).

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.1 Curs	Cunoștințe teoretice/practice	Test grilă cu 80 întrebări, notate cu 1 punct întrebările "teoretice" și cu 2 puncte întrebările referitoare la fragmente de cod sursă	30 %
	Rezolvarea unei probleme de complexitate medie	Examinare orală, pe calculator	30 %
	Stil de programare	Examinare orală, pe calculator	3 %
	Participare la curs	Evidența prezenței	3 %
10.2 Seminar /laborator	Rezolvarea problemelor corespunzătoare lucrărilor de laborator	Prezentarea rezolvărilor, răspunsuri la întrebări, răspunsuri la teste grilă periodice	25 %
	Teme de casă	Prezentarea rezolvărilor, răspunsuri la întrebări	5 %
	Prezența regulată la laborator	Evidența prezenței	3 %
10.3 Standard minim de performanță (volumul de cunoștințe minim necesar pentru promovarea disciplinei și modul în care se verifică stăpânirea lui)			
<ul style="list-style-type: none"> • Testul grilă cu cel puțin nota 5 (jumătate din punctaj) • Proiectarea, testarea și executarea unui program de complexitate medie: programul trebuie să fie compilat și executat, conține cel puțin o ierarhie de clase cu clasă de bază și două clase derivate și listă eterogenă, trebuie să funcționeze cel puțin inserarea și afișarea tuturor elementelor; programul trebuie să fie scris ordonat (să respecte cerințe minime de stil de programare) și studentul să poată răspunde satisfăcător la întrebări legate de soluțiile din program 			

11. Compatibilitate internațională

- Cornell University http://courses.cornell.edu/preview_course_nopop.php?catoid=14&coid=160280
- Technische Universität Wien <http://www.complang.tuwien.ac.at/franz/objektorientiert.html>
- University of Edinburgh <http://www.inf.ed.ac.uk/student-services/teaching-organisation/taught-course-information/year-guides/undergraduate-first-year-guide/inf1-op/inf1-op-object-oriented-programming>

Data
completării

Semnătura titularului de curs

Semnătura titularilor de seminar

Prof. dr. ing. Vasile STOICU-
TIVADAR

Asist. dr. ing. Mihaela VIDA, asist. dr. ing. Norbert GAL

.....

.....

Data avizării în departament

Semnătura directorului de departament

Prof. dr. ing. Ioan SILEA

.....