

**FIȘA DISCIPLINEI****1. Date despre program**

1.1 Instituția de învățământ superior	Universitatea “Alexandru Ioan Cuza” din Iași
1.2 Facultatea	Facultatea de Informatică
1.3 Departamentul	Departamentul de Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Informatică/Licențiat în Informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	PROGRAMARE ORIENTATĂ-OBIECT						
2.2 Titularul activităților de curs	PROF. DR. DOREL LUCANU Conf. DR. DRAGOȘ GAVRILUȚ						
2.3 Titularul activităților de seminar	Conf. DR. FRAGOȘ GAVRILUȚ						
2.4 An de studiu	I	2.5 Semestru	2	2.6 Tip de evaluare	E	2.7 Regimul disciplinei*	OB

* OB – Obligatoriu / OP – Opțional

3. Timpul total estimat (ore pe semestru și activități didactice)

3.1 Număr de ore pe săptămână	4	din care: 3.2 curs	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și altele					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					14
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					-
Examinări					4
Alte activități					-
3.7 Total ore studiu individual					56
3.8 Total ore pe semestru					116
3.9 Număr de credite					5

4. Precondiții (dacă este cazul)

4.1 De curriculum	Introducere în programare
4.2 De competențe	Programare procedurală, partea imperativă a limbajului C++

5. Condiții (dacă este cazul)

5.1 De desfășurare a cursului	--
5.2 De desfășurare a seminarului/ laboratorului	Prezența obligatorie la laborator



6. Competențe specifice acumulate

Competențe profesionale	La finalizarea cu succes a acestei discipline, studenții vor fi capabili să: C1. Explice sintaxa și semantica instrucțiunilor limbajului C++ C2. Descrie prin diagrame UML sisteme software de nivel elementar și mediu ca dificultate C3. Utilizeze conceptele de POO și șabloanelor în proiectarea sistemelor software C4. Analizeze cerințele unui sistem software de nivel elementar și mediu ca dificultate C5. Scrie cod C++ pornind de la specificația scrisă în UML a unui sistem software
Competențe transversale	CT1. Combinarea cunoștințelor de modelare OO cu cele de programare OO. CT2. Dezvoltarea capacității de a modela cu concepte OO diverse domenii de aplicații practice. CT3. Capacitatea de a utiliza modelele UML ca mijloc de comunicare cu clienții din aceste domenii practice.

7. Obiectivele disciplinei (din grila competențelor specifice acumulate)

7.1 Obiectivul general	Inițiere în programarea orientată-obiect la nivel abstract utilizând UML și la nivel concret utilizând limbajul de programare C++.
7.2 Obiectivele specifice	O1. Cunoașterea conceptelor de clase, obiecte, ierarhii de clase, polimorfism, clase abstracte, interfețe, clase parametrizate. O2. Modelarea în UML la nivel introductiv. O3. Cunoașterea limbajului C++ (ISO Standard), cu accent pe reprezentarea claselor, obiectelor, și a relațiilor dintre acestea. O4. Cunoașterea celor mai utilizate șabloane de proiectare. O5. Utilizarea Bibliotecii standard de tipuri generice (STL).

8. Conținut

8.1	Curs	Metode de predare	Observații (ore și referințe bibliografice)
1.	Introducere în C++, istoric, compilatoare, gramatica	Expunere	2
2.	Conceptele de clasă și obiect în C++	Expunere	2
3.	Operatori și metode	Expunere	2
4.	Ierarhii în C++	Expunere	2



5.	Macrouri și template-uri	Expunere	2
6.	STL I	Expunere	2
7.	STL II	Expunere	2
8.	Evaluare parțială	Portofolii, Interviu	6
9.	Exceptii in C++	Expunere	2
10.	Modelare OO	Expunere	2
11.	Proiectare OO I	Expunere	2
12.	Proiectare OO II	Expunere	2
13.	Proiectare OO III	Expunere	2
14.	MVC și și GUI	Expunere	2

Bibliografie

H. Schildt: C++ manual complet, Teora, 2000

D. Kaler, M.J. Tobler, J. Valter: C++, Teora, 2000

Bjarne Stroustrup: The C++ Programming Language, Addison-Wesley, 3rd edition, 1997

Bruce Eckel : Thinking in C++, 2nd Edition

*** : Online C++ tutorial

Donald Bell. UML basics: An introduction to the Unified Modeling Language

SGI Standard Template Library Programmer's Guide

8.2	Seminar / Laborator	Metode de predare	Observații (ore și referințe bibliografice)
1.	Recapitulare C - lucru cu pointeri, citire din fisiere.	Experiment, Lucru individual, Metode interactive	2
2.	Clase C++	Idem	2
3.	Supraincarcare. Operatori.	Idem	2
4.	Ierarhii de clase	Idem	2
5.	Polimorfism. Pluginuri. Librarii statice si dinamice	Idem	2



6.	STL 1	Idem	2
7.	STL 2	Idem	2
8.	Test	Exerciții rezolvate la calculator	2
9.	Excepții	Experiment, Lucru individual, Metode interactive	2
10.	Modelare și proiectare I	Idem	2
11.	Modelare și proiectare II	Idem	2
12.	Modelare și proiectare III	Idem	2
13.	Modelare și proiectare IV	Idem	2
14.	Test	Exerciții rezolvate la calculator	2

Bibliografie

H. Schildt: C++ manual complet, Teora, 2000

D. Kaler, M.J. Tobler, J. Valter: C++, Teora, 2000

Bjarne Stroustrup: The C++ Programming Language, Addison-Wesley, 3rd edition, 1997

Bruce Eckel : Thinking in C++, 2nd Edition

*** : Online C++ tutorial

Donald Bell. UML basics: An introduction to the Unified Modeling Language

SGI Standard Template Library Programmer's Guide

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

Conținutul disciplinei este coroborat cu tematica concursurilor pentru ocuparea posturilor în companiile IT.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere în nota finală (%)
10.4 Curs	<ul style="list-style-type: none">- abilitatea de a modela și proiecta OO un sistem simplu- abilitatea de a transforma o diagrama UML in cod C++- capacitatea de a înțelege și explica un cod C++- calitatea formulării răspunsurilor	Test scris	60%



10.5 Seminar/ Laborator	<ul style="list-style-type: none">- abilitatea de a construi programe POO care rezolvă probleme de nivel simplu și mediu- abilitatea de a aplica șabloane în găsirea soluțiilor- calitatea programelor scrise	2 teste de laborator (30%), activitate laborator (10%)	40%
10.6 Standard minim de performanță			
Punctajul se calculează în felul următor:			
<ul style="list-style-type: none">▪ Un prim test de laborator în săptămâna a 8-a (30 pct)▪ Un al doilea test de laborator la finalul perioadei de activitate didactică a semestrului (30 pct)▪ Un test final cu materia predată la curs (în sesiune, 30 pct)▪ Punctaj pentru participarea la laborator (18 pct) (1 punct se da pentru fiecare prezență = deci maxim 12 puncte se pot lua din prezente, restul de 6 puncte se iau în funcție de activitatea din timpul laboratorului)			
Sunt necesare următoarele pentru a putea promova materia:			
<ul style="list-style-type: none">▪ Minim 20 de puncte din cele două teste de laborator▪ Minim 10 puncte la testul final cu materia predată la curs▪ Minim 10 puncte din participare la laborator			
Notele se stabilesc conform cu criteriile ECTS			