

Universitatea „Lucian Blaga” din Sibiu  
 Facultatea de Științe  
 Catedra de Informatică  
 Domeniul de studii de licență: Informatică  
 Specializarea: Informatică

## PROGRAMA ANALITICĂ

<b>Denumirea disciplinei: Programare orientată obiect</b>
<b>Codul disciplinei: 3906S02I009</b>
<b>Anul de studiu și semestrul în care se studiază disciplina: I/2</b>
<b>Regimul disciplinei (obligatorie O, opțională A sau facultativă L): O</b>
<b>Categoria formativă (fundamentală Fd, de specialitate Sp, generală Gen):Sp</b>
<b>Discipline anterioare cerute *: Programare procedurală</b>
<b>Forma de evaluare (examen E, verificare V, colocviu C):E</b>
<b>Catedra care coordonează disciplina: Catedra de Informatică</b>
<b>Titularul / titularii disciplinei: Prof.univ.dr. Valer Roșca</b>

\* disciplinele studiate anterior a căror cunoaștere este necesară pentru însușirea disciplinei

<b>Extinderea disciplinei în planul de învățământ *:</b>				
Curs	Seminar	Laborator	Proiect	Total ( <i>NOAD<sub>sem</sub></i> )
2 x 14 = 28 ore		2 x 14 = 28 ore		56

\* numărul semestrial de ore de activități didactice directe

<b>Bugetul de timp și creditele alocate disciplinei</b>			
<i>NOAD<sub>sem</sub></i>	<i>NOSI<sub>sem</sub></i>	<i>NOT<sub>sem</sub> = NOAD<sub>sem</sub> + NOSI<sub>sem</sub></i>	Numărul de credite
56	84	140	5

<b>Obiectivele disciplinei</b>
<p><b>Obiectivele cursului</b></p> <ul style="list-style-type: none"> <li>-Însușirea cunoștințelor generale despre paradigma programării orientate pe obiecte.</li> <li>-Însușirea unui limbaj care suportă paradigma de programare, ca instrument efectiv de lucru (limbajul C++).</li> </ul>
<p><b>Obiectivele activităților aplicative (seminar, laborator, proiect)</b></p> <ul style="list-style-type: none"> <li>-Formarea unor deprinderi de lucru pentru realizarea de programe în tehnica programării orientate obiect.</li> </ul>

<b>Conținutul disciplinei</b> (capitolele cursului / tematica seminarului / lucrărilor practice / etapele proiectului)			
<b>CURS</b>			
<b>Nr. crt.</b>	<b>Tema</b>	<b>Nr.ore</b>	<b>Săptămâna</b>
1	<b>Metoda programării orientată pe obiecte (OOP – Object Oriented Programming).</b> Contextul apariției metodei .Principii ale metodei de programare. Calitatea produselor program și noua metodă. Limbajele de programare care suportă metoda de programare orientată pe obiecte. Îmbunătățiri în limbajul C++. Programul în C++ în metoda OOP.	2	1
2	<b>Clase și obiecte în C++.</b> Declararea unei clase. Definierea unei clase. Declararea obiectelor. Referirea membrilor obiectelor. Structura spațiului obiectelor. Constructori generali. Constructori de copiere. Destructorii.	4	2-3
3	<b>Facilități suplimentare ale metodei OOP în C++.</b> Membri statici.Funcții membru constante. Obiecte constante. Array-uri de obiecte. Supraîncărcarea operatorilor. Funcții și clase prietene.	2	4
4.	<b>Operații de intrare și ieșire standard cu obiecte în C++.</b> Clase și obiecte de intrare și ieșire. operații standard cu format. Controlul formatului în operații. Manipulatori. Operații fără format. Supraîncărcarea operatorilor de extracție și inserție pentru obiectele unei clase.	4	5-6
5.	<b>Supraîncărcarea unor operatori remarcabili în C++.</b> Supraîncărcarea operatorului de atribuire. Supraîncărcarea operatorilor de relație. Supraîncărcarea operatorului de apel de funcție. Supraîncărcarea operatorului de indexare. Conversii. Supraîncărcarea operatorului de cast.	4	7-8
6.	<b>Moștenirea în C++.</b> Moștenirea simplă: controlul membrilor. Moștenirea multiplă: moștenirea repetată. Ierarhii de clase: constructorii și destructorii, supraîncărcarea funcțiilor, operatorul de atribuire, conversia obiectelor și a pointerilor la obiecte, redefinirea funcțiilor. Polimorfism și funcții virtuale în ierarhii de clase. Interapelarea funcțiilor în ierarhii de clase. Virtualizarea destructorilor în ierarhii de clase. Clase abstracte.	6	9-11
7.	<b>Genericitatea în C++.</b> Funcții generice. Clase generice. Biblioteca STL de clase generice: principalele clase container. Algoritmi generici, funcții generice.	2	12
8.	<b>Lucrul cu fișiere în metoda OOP C++.</b> Modul de lucru cu obiecte fișier. Tratarea erorilor în lucrul cu fișier. Prelucrarea fișierelor de tip text. Realizarea lucrului cu fișiere binare : accesul secvențial și accesul direct.	2	13
9.	<b>Introducere în proiectare.</b> Modelarea obiectelor: modelul obiectelor, modelul dinamic al obiectelor, modelul funcțional al obiectelor. Etapele activității de realizare a proiectului: analiza, proiectarea de ansamblu, proiectarea de detaliu.	2	14

<b>SEMINAR / LABORATOR / PROIECT</b>			
<b>Nr. crt.</b>	<b>Tema</b>	<b>Nr.ore</b>	<b>Săptămâna</b>
1.	<b>Utilizare îmbunătățirilor limbajului C++ în programare procedurală.</b> Realizarea unor programe care utilizează referințe în transmiterea parametrilor. Realizarea de programe de lucru cu memoria dinamică utilizând operatorii new și delete. Programe care utilizează funcții cu argumente implicite și realizează supraîncărcarea funcțiilor.	4	1-2
2.	<b>Clase și obiecte de calcul numeric.</b> Realizare de programe care implementează clase numerice pentru lucrul cu numere raționale, numere complexe, matrice de numere standard și cu numere raționale sau complexe. Se utilizează supraîncărcarea operatorilor de inserție și extracție și a operatorilor de calcul.	6	3-5
3.	<b>Clase container proprii.</b> Realizare unui program care crează o clasă pentru array dinamic, container de elemente de tipul standard și/sau de tipurile definite la laboratorul anterior. Se utilizează spațiul dinamic.	4	6-7
4.	<b>Ierarhii de clase.</b> Se realizează o ierarhie de clase pentru figuri geometrice plane care să pună în evidență clase abstracte, să utilizeze moștenirea simplă și multiplă și să redefinească și să virtualizeze funcții.	6	8-10
5.	<b>Genericitate.</b> Realizarea unor programe care utilizează funcții și clase generice.	4	11-12
6.	<b>Lucrul cu fișiere.</b> Se realizează programe care să ofere facilități de creare, consultare aleatorie, actualizare și întocmire de rapoarte, utilizând fișiere.	4	12-13
7.	<b>Susținerea proiectului de disciplină</b>	2	14

#### **Descrierea metodelor de predare**

Prelegerea, dezbateră, învățarea prin cooperare, explicația, exemplificarea, munca individuală și de grup, dezbateră.

#### **Descrierea formelor și metodelor de evaluare a cunoștințelor**

Disciplina este prevăzută cu examen scris în semestrul I. Studenții trebuie să realizeze un proiect pe o tematică dinaintea aleasă. Proiectul și activitatea în laborator se notează și intră în nota finală. Nota finală se stabilește după cum urmează:

- |                            |     |
|----------------------------|-----|
| 1. Activitate în laborator | 10% |
| 2. Proiecte de semestru    | 30% |
| 3. Examen final            | 60% |

#### **Bibliografie obligatorie**

1. Roșca V – Programarea orientată pe obiecte în C++. Editura Universității "Lucian Blaga", Sibiu 2005.
2. Dr. Jamsa, ș.a.– *Totul despre C și C++. Manualul fundamental de programare în C și C++.*

Editura Teora 2000.

3. Stroustrup B. – C++. Editura Teora 2003.

**Bibliografie opțională**

1. Mușlea I.- Inițiere în C++. Programarea orientată pe obiecte. Microinformatica 1992.
2. Rumbaugh J., ș.a – Object Oriented Modelling and Design. Prentice Hall International Inc.1991.
3. Schildt H. – C++ Manual complet. Editura Teora 2000.

**Data elaborării:**

**Titularul / titularii disciplinei,  
Prof.univ.dr. Valer Roșca**