

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea „Lucian Blaga” din Sibiu
1.2 Facultatea / Departamentul	Științe / Departamentul de Matematică și Informatică
1.3 Catedra	Matematică
1.4 Domeniul de studii	Matematică
1.5 Ciclul de studii	Licență
1.6 Programul de studii/Calificarea	Matematică – Informatică

2. Date despre disciplină

2.1 Denumirea disciplinei				Programare orientată pe obiecte			
2.2 Titularul activităților de curs				Lector univ. dr. Mircea-Adrian Mușan			
2.3 Titularul activităților de seminar				Lector univ. dr. Costel Maniu			
2.4 Anul de studiu	1	2.5 Semestrul	2	2.6 Tipul de evaluare	Ex	2.7 Regimul disciplinei	0

3. Timpul total estimat (ore pe semestru al activităților 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: 3.2 curs	28	3.3 seminar/laborator	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					5
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					24
Tutoriat					28
Examinări					2
Alte activități					
3.7 Total ore studiu individual					84
3.9 Total ore pe semestru					140
3.10 Numărul de credite					5

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Parcursarea disciplinei “Programare procedural”
4.2 de competențe	–

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	-
5.2. de desfășurare a seminarului/laboratorului	-

6. Competențele specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> • Cunoașterea și utilizarea adecvată a noțiunilor specifice tehnicii OOP. • Capacitatea de a interpreta și înțelege programe și aplicații realizate, pe baza codului sursă. • Capacitatea de a proiecta și realiza programe folosind tehnica OOP, pe baza specificațiilor.
Competențe transversale	<ul style="list-style-type: none"> • Dezvoltarea atitudinii pozitive față de muncă și responsabilitate pentru propria pregătire profesională. • Dezvoltarea spiritului de muncă în echipă.

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

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Însușirea cunoștințelor generale despre paradigma programării orientate pe obiecte.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Însușirea unui limbaj care suportă paradigma de programare, ca instrument efectiv de lucru (limbajul C++). • Elemente primare și fundamentale de proiectare. • Introducerea în elementele programării vizuale prin intermediul limbajului Visual C++.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
Metoda programării orientată pe obiecte. (1 sapt) 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. Limbajul C++, superset al limbajului C. Tipul referință. Funcții in-line. Pointeri. Structs. Pointeri Smart.	Expunerea, explicația, exemplificarea și conversația frontală	
Introducere în programarea obiectuală. Clase și obiecte. (2 sapt) Tipul abstract de date. Încapsulare. Declararea și definirea unei clase. Declararea obiectelor și referirea membrilor. Structura spațiului obiectelor. Constructori generali. Constructori de inițializare prin copiere. Destructorii.	Expunerea, explicația, exemplificarea și conversația frontală	
Facilități avansate ale programării OPP în limbajul	Expunerea, explicația, exemplificarea și	

<p>C++. (1 sapt) Membri statici. Funcții membru constante. Obiecte constante. Array-uri de obiecte. Supraîncărcarea operatorilor. Funcții și clase prietene.</p>	conversația frontală	
<p>Realizarea operațiilor de intrare și ieșire prin intermediul obiectelor. (1 sapt) 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.</p>	Expunerea, explicația, exemplificarea și conversația frontală	
<p>Supraîncărcarea operatorilor remarcabili în C++. (1 sapt) Supraîncărcarea operatorului de atribuire. Supraîncărcarea operatorilor de relație. Supraîncărcarea operatorului de apel de funcție. Supraîncărcarea operatorului de indexare. Conversii. Supraîncărcarea operatorului de <i>cast</i>.</p>	Expunerea, explicația, exemplificarea și conversația frontală	
<p>Principiul moștenirii în OOP. (3 sapt) 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.</p>	Expunerea, explicația, exemplificarea și conversația frontală	
<p>Obiecte fișier în programarea orientată spre obiecte. (1 sapt) 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.</p>	Expunerea, explicația, exemplificarea și conversația frontală	
<p>Genericitatea în programarea orientată obiect. (1 sapt) Funcții generice. Clase generice. Biblioteca STL de clase generice: principalele clase container. Algoritmi generici, funcții generice.</p>	Expunerea, explicația, exemplificarea și conversația frontală	
<p>Tratarea excepțiilor și portabilitatea tipurilor. (1 sapt) Tratarea excepțiilor. Instrucțiunea <i>throw</i>. Lansarea excepțiilor cu o funcție din cadrul blocului <i>try</i>. Aplicații ale tratării excepțiilor. Restricționarea și relansarea excepțiilor. Spații de nume.</p>	Expunerea, explicația, exemplificarea și conversația frontală	

Elemente primare și fundamentale de proiectare. (1 sapt) 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.	Expunerea, explicația, exemplificarea și conversația frontală	
Inițiere în programarea concurrentă. (1 sapt) Procese. Fire de execuție.	Expunerea, explicația, exemplificarea și conversația frontală	
Bibliografie <ol style="list-style-type: none"> 1. Valer Roșca, <i>Programarea orientată pe obiecte în C++</i>, Editura Universității "Lucian Blaga", Sibiu, 2005 2. Liviu Negrescu, <i>Limbajele C și C++ pentru începători. Limbajul C++</i>, Editura Albastră, Cluj-Napoca, 2002 3. Dr. Kris Jamsa, Lars Klander, <i>Totul despre C și C++. Manual fundamental de programare în C și C++</i>, Editura Teora, București, 2000 4. Herbert Schildt, <i>C++. Manual complet</i>, Editura Teora, București, 2000 		
8.2 Seminar/laborator	Metode de predare	Observații
Utilizarea îmbunătățirilor limbajului C++ în programarea procedurală. (1 sapt) 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. Realizarea programelor ce utilizează funcții in-line și operatorul de rezoluție.	Explicația, exemplificarea, învățarea prin descoperire	
Introducere în conceptul de OOP. (2 sapt) Aplicații ce construiesc clase, declară și alocă memorie pentru obiecte. Realizarea unor aplicații pentru exemplificarea instanțierii prin utilizarea constructorilor. Destructorii.	Explicația, exemplificarea, învățarea prin descoperire	
Supraîncărcarea operatorilor aritmetici. (1 sapt) Realizarea aplicațiilor cu dezvoltare de clase ce implică funcții prieten și operatori aritmetici supraîncărcați.	Explicația, exemplificarea, învățarea prin descoperire	
Realizarea de aplicații prin operații de intrare/ieșire în C++. (1 sapt) Clase de bază pentru streamuri. Utilizarea membrilor clasei IOS pentru formatarea op. de I/O. Manipulatori. Supraîncărcarea operatorilor "<<" și ">>".	Explicația, exemplificarea, învățarea prin descoperire	
Realizarea de aplicații prin operatori speciali supraîncărcați în C++. (1 sapt) Pointerul this. Supraîncărcarea operatorului "[]". Supraîncărcarea operatorului "()". Supraîncărcarea operatorului ";". Supraîncărcarea operatorului "=". Supraîncărcarea operatorilor "new" și "delete".	Explicația, exemplificarea, învățarea prin descoperire	
Construcția de aplicații folosind conceptul de moștenirea simplă în tehnica OOP. (1 sapt) Controlul accesului la membrii moșteniți. Constructorii și destructorii în ierarhii de clase. Alte probleme generate de procesul moștenirii. Aplicații complexe utilizând procesul de moștenire simplă.	Explicația, exemplificarea, învățarea prin descoperire	
Construcția de aplicații folosind moștenirea multiplă în OOP. (1 sapt) Clase virtuale. Funcții virtuale. Polimorfism.	Explicația, exemplificarea, învățarea prin descoperire	
Clase abstracte și clase prieten. (1 sapt) Realizarea de aplicații complexe ce folosesc concepte	Explicația, exemplificarea,	

precum: polimorfism, clase abstracte și clase prieten.	învățarea prin descoperire	
Prelucrarea fișierelor text și binare în OOP. (1 sapt) Ierarhia de clase pentru lucrul cu fișiere. Deschiderea fișierelor. Închiderea fișierelor. Intrări și ieșiri la nivel de fișier. Accesul la datele dintr-un fișier. Realizarea de aplicații prin folosirea obiectelor fișier în tehnica OOP.	Explicația, exemplificarea, învățarea prin descoperire	
Funcții generice. Clase generice. (1 sapt) Șabloane. Clase generice. Realizarea de exemple cu două clase generice.	Explicația, exemplificarea, învățarea prin descoperire	
Realizarea de aplicații pentru tratarea excepțiilor. (1 sapt) Tratarea excepțiilor. Instrucțiunea <i>throw</i> . Lansarea excepțiilor cu o funcție din cadrul blocului <i>try</i> . Aplicații ale tratării excepțiilor. Spații de nume.	Explicația, exemplificarea, învățarea prin descoperire	
Introducere în proiectare. (1 sapt) Studiu de caz.	Explicația, exemplificarea, învățarea prin descoperire	
Prezentarea proiectului final realizat individual de către studenți. (1 sapt)	Explicația, exemplificarea, învățarea prin descoperire	
Bibliografie		
1. Dr. Kris Jamsa, Lars Klander, <i>Totul despre C și C++. Manual fundamental de programare în C și C++</i> , Editura Teora, București, 2000		
2. Herbert Schildt, <i>C++. Manual complet</i> , Editura Teora, București, 2000		

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

Programarea orientată spre obiecte constituie fundament și bază în implementarea algoritmilor și în realizarea proiectelor reale în firmele de soft existente pe piața locală, națională și internațională

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Test de evaluare final	Testare scrisă	60%
10.5 Seminar/laborator	Testare de laborator	Teme de casă, Activitate, Test de laborator, Proiect final	40%
10.6 Standard minim de performanță			
Pentru promovarea examenului, trebuie obținută minim nota 5 la evaluările pe parcurs și la examenul de evaluare finală			

Data completării

Semnătura titular curs

Semnătura titular de seminar

26.09.2016

Data avizării în catedră

Semnătura directorului de departament

28.09.2016

Prof.univ.dr. Mugur Acu