

FIȘA DISCIPLINEI

Course title/ Titlul cursului:	Programare orientată obiect				
Course code/ Codul cursului:	Type of course/ Tipul cursului:	Level of course/ Nivelul cursului:	Year of study/ An de studiu:	Semester/ Semestru:	Number of credits/ Număr de credite:
38060209	Obligatoriu	fundamental	1	2	5
Name of lecturer/ Numele profesorului:	Titular curs/seminar/laborator/proiect: Lector univ. dr. Mușan Mircea				
Department/ Departament (ce coordonează disciplina):	DEPARTAMENTUL DE MATEMATICĂ ȘI INFORMATICĂ				
Extinderea disciplinei în planul de învățământ					
Lecture/Curs	Seminar	Lab/Laborator	Project/Proiect	Total (NOAD_{sem})	
28	–	28	–	56	
Bugetul de timp pentru studiu individual					
Type of activities / Denumirea activității	Hours/Nr. Ore	Type of activities / Denumirea activității		Hours/Nr. Ore	
1. Studiarea notițelor de curs	10	5. Pregătirea seminariilor/laboratoarelor		10	
2. Studiul după suport de curs	14	6. Elaborarea referatelor, temelor individuale		10	
3. Studiarea bibliografiei minimale	15	7. Pregătirea pentru evaluările periodice		10	
4. Documentare suplimentară (internet, bibliotecă ș.a.)	10	8. Participarea la consultații		5	
Total ore alocate studiului individual NOSI_{sem}				84	
Objectives of the course/ Obiectivele cursului:	<ul style="list-style-type: none"> - Însușirea cunoștințelor generale despre paradigma programării orientate pe obiecte. - Însușirea unui limbaj care suportă paradigma de programare, ca instrument efectiv de lucru (limbajul C++). - Introducerea în elementele programării vizuale prin intermediul limbajului Visual C++. 				
Prerequisites/ Discipline precursoare obligatorii:	Programare procedurală				
Course contents/ Conținutul cursului:	Cursul 1 <i>sapt. 1</i>	Metoda programării orientată pe obiecte. 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++, supraset al limbajului C. Tipul referință. Funcții in-line. Pointeri. Structs. Pointeri Smart.			
	Cursul 2	Introducere în programarea obiectuală. Clase și obiecte. Tipul abstract de date. Încapsulare. Declarația și definirea unei			

	<i>săpt. 2 – 3</i>	clase. Declararea obiectelor și referirea membrilor. Structura spațiului obiectelor. Constructori generali. Constructori de inițializare prin copiere. Destructorii.
	Cursul 3 <i>săpt. 4</i>	Facilități avansate ale programării OPP în limbajul C++. Membri statici. Funcții membru constante. Obiecte constante. Array-uri de obiecte. Supraîncărcarea operatorilor. Funcții și clase prietene.
	Cursul 4 <i>săpt. 5</i>	Realizarea operațiilor de intrare și ieșire prin intermediul obiectelor. 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.
	Cursul 5 <i>săpt. 6</i>	Supraîncărcarea operatorilor remarcabili în C++. 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> .
	Cursul 6 <i>săpt. 7 – 9</i>	Principiul moștenirii în OOP. 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.
	Cursul 7 <i>săpt. 10</i>	Obiecte fișier în programarea orientată spre obiecte. 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.
	Cursul 8 <i>săpt. 11</i>	Genericitatea în programarea orientată obiect. Funcții generice. Clase generice. Biblioteca STL de clase generice: principalele clase container. Algoritmi generici, funcții generice.
	Cursul 9 <i>săpt. 12</i>	Tratarea excepțiilor și portabilitatea tipurilor. 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.
	Cursul 10 <i>săpt. 13</i>	Elemente primare și fundamentale de proiectare. 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.
	Cursul 11 <i>săpt. 14</i>	Inițiere în programarea concurentă. Procese. Fire de execuție.
Lab/Seminary contents/ Conținutul laboratorului/seminarului:	Cursul 1 <i>săpt. 1</i>	Utilizarea îmbunătățirilor limbajului C++ în programarea procedurală. Realizarea unor programe care utilizează referințe în transmiterea parametrilor. Realizarea de programe de lucru cu memoria dinamică utilizând operatorii <i>new</i> și <i>delete</i> . Programe care utilizează funcții cu argumente implicite și realizează supraîncărcarea funcțiilor. Realizarea programelor ce utilizează funcții <i>in-line</i> și operatorul de rezoluție.
	Cursul 2 <i>săpt. 2 – 3</i>	Introducere în conceptul de OOP. Aplicații ce construiesc clase, declară și alocă memorie pentru obiecte. Realizarea unor aplicații pentru exemplificarea

		instanțierii prin utilizarea constructorilor. Destructori.
	Cursul 3 <i>săpt. 4</i>	Supraîncărcarea operatorilor aritmetici. Realizarea aplicațiilor cu dezvoltare de clase ce implică funcții prieten și operatori aritmetici supraîncărcați.
	Cursul 4 <i>săpt. 5</i>	Realizarea de aplicații prin operații de intrare/ieșire în C++. Clase de bază pentru streamuri. Utilizarea membrilor clasei IOS pentru formatarea op. de I/O. Manipulatori. Supraîncărcarea operatorilor "<<" și ">>".
	Cursul 5 <i>săpt. 6</i>	Realizarea de aplicații prin operatori speciali supraîncărcați în C++. Pointerul this. Supraîncărcarea operatorului "[]". Supraîncărcarea operatorului "()". Supraîncărcarea operatorului ";". Supraîncărcarea operatorului "=". Supraîncărcarea operatorilor "new" și "delete".
	Cursul 6 <i>săpt. 7</i>	Construcția de aplicații folosind conceptul de moștenirea simplă în tehnica OOP. 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.
	Cursul 7 <i>săpt. 8</i>	Construcția de aplicații folosind moștenirea multiplă în OOP. Clase virtuale. Funcții virtuale. Polimorfism.
	Cursul 8 <i>săpt. 9</i>	Clase abstracte și clase prieten. Realizarea de aplicații complexe ce folosesc concepte precum: polimorfism, clase abstracte și clase prieten.
	Cursul 9 <i>săpt. 10</i>	Prelucrarea fișierelor text și binare în OOP. 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.
	Cursul 10 <i>săpt. 11</i>	Funcții generice. Clase generice. Șabloane. Clase generice. Realizarea de exemple cu două clase generice.
	Cursul 11 <i>săpt. 12</i>	Realizarea de aplicații pentru tratarea excepțiilor. Tratarea excepțiilor. Instrucțiunea <i>throw</i> . Lansarea excepțiilor cu o funcție din cadrul blocului <i>try</i> . Aplicații ale tratării excepțiilor. Spații de nume.
	Cursul 12 <i>săpt. 13</i>	Introducere în proiectare. Studiu de caz.
	Cursul 13 <i>săpt. 14</i>	Prezentarea proiectului final realizat individual de către studenți.
Teaching methods/ Metode de predare: expunerea, explicația, exemplificarea și conversația frontală		Language of instruction/ Limba de predare: Româna
Assesment methods/ Sisteme de evaluare:	Activități aplicative - 10 %	1. Teme de curs/pondere = 70 %(nCPC) 2. Referate de disciplină = 0 %(nCPC) 3. Lucrări practice = 30 %(CPC)
	Proiect - 30 %	CPE (CPE – condiționează participarea la examen)
	Examen parțial - 0 %	(nCPE – nu condiționează participarea la examen)
	Examen de semestru - 60 %	(condiționează evaluarea finală)

Competențe specifice disciplinei	
1. Competențe privind cunoașterea și înțelegerea	Cunoașterea și utilizarea adecvată a noțiunilor specifice.
2. Competențe în domeniul explicării și interpretării	Capacitatea de a interpreta și înțelege programe realizate, pe baza codului sursă.
3. Competențe instrumental - aplicative	Capacitatea de a proiecta și realiza programe pe baza specificațiilor.
4. Competențe atitudinale	Dezvoltarea atitudinii pozitive față de muncă și responsabilitate pentru propria pregătire profesională.
Competențele generale sunt menționate în Fișa specializării	
Recommended reading/ Referințe bibliografice recomandate (max. 10):	<ol style="list-style-type: none"> 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
More references/ Referințe bibliografice suplimentare:	<ol style="list-style-type: none"> 1. Bruce Eckel – "Thinking in C++". Prentice Hall, 2000 2. Andrew Koenig and Barbara E. Moo – "Accelerated C++. Practical Programming by Example". Addison-Wesley, 2000

Data elaborării:
01.10.2013

Titulari disciplină:
Lect. univ. dr. Mușan Mircea