

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Facultatea de Inginerie
1.3 Departamentul	Inginerie Electrică, Electronică și Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	21.00

2. Date despre disciplină

2.1 Denumirea disciplinei	Programarea Orientată pe Obiecte 1 (C++)						
2.2 Aria de conținut							
2.3 Responsabil de curs	Conf dr. ing. Cosma Ovidiu – ovidiu.cosma@mi.utcluj.ro						
2.4 Titularul activităților de seminar / laborator / proiect	Lector dr. ing. Gâta Marieta - marieta.gata@ieec.utcluj.ro						
2.5 Anul de studiu	2	1	2	2.7 Tipul de evaluare	E	2.8 Regimul disciplinei	DOB

3. Timpul total estimat

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
Studiul după manual, suport de curs, bibliografie și notițe					16
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					14
Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri					14
Tutoriat					
Examinări					2
Alte activități: elaborare de programe					28
3.7 Total ore studiu individual	74				
3.8 Total ore pe semestru	130				
3.9 Numărul de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Promovarea examenului la disciplina Programarea Calculatoarelor și Limbaje de Programare 2
4.2 de competențe	Rezultă din 4.1.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	cursurile, laboratoarele și seminariile se vor desfășura online, prin platforma Microsoft Teams
5.2. de desfășurare a seminarului / laboratorului / proiectului	

6. Competențele specifice acumulate

Competențe profesionale	Definirea conceptelor, metodelor și instrumentelor aplicate în ingineria software, Cunoașterea paradigmei programării orientate pe obiecte, Elaborarea și analiza unor algoritmi în manieră obiectuală, Programarea în limbaje orientate pe obiecte (C++), Cunoașterea unor medii de programare moderne, Conceperea unor modele simple, descrierea și implementarea lor într-un limbaj orientat pe obiecte.
Competențe transversale	Manifestarea unei atitudini responsabile față de domeniul științific, Respectarea regulilor de munca organizată și eficientă, Selectarea eficientă a resurselor informaționale, Utilizarea eficientă a surselor de formare profesională, Valorificarea creativă a propriului potențial, Utilizarea unor metode și tehnici eficiente de învățare, Dezvoltare a capacităților de valorificare a cunoștințelor, Respectarea principiilor și a normelor de etică profesională, Elaborarea proiectului propriu de dezvoltare profesională, de adaptare la cerințele unei societăți dinamice.

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

7.1 Obiectivul general al disciplinei	Înțelegerea noțiunilor fundamentale privind programarea orientată pe obiecte și formarea deprinderilor necesare proiectării de aplicații performante.
7.2 Obiectivele specifice	Utilizarea unor medii de programare moderne. Realizarea și testarea unor aplicații în limbajul C++; Lucrul în echipă.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
8.1.1 Paradigmele programării orientate pe obiecte.	prelegeri online: expunere, algoritmizare, explicatie, problematizare, exemple, demonstrație didactică.	
8.1.2 Trecerea de la C la C++. Variabile referință, transferul prin referință. Supradefinirea funcțiilor. Funcții inline. Parametri cu valori implicite. Operatorii new și delete.		
8.1.3 Clase și obiecte. Încapsularea datelor. Constructori și destructori. Autoreferința. Operații cu obiecte. Structuri și uniuni.		
8.1.4 Constructorul de copiere. Membri statici. Funcții și clase prietene.		
8.1.5 Obiecte constante. Obiecte volatile. Tablouri de obiecte. Clase cu câmpuri obiecte. Pointeri către membrii unei clase.		
8.1.6 Supradefinirea operatorilor.		
8.1.7 Conversii de tip definite prin program.		
8.1.8 Derivarea claselor. Constructori și destructori pentru clasa derivată. Redefinirea metodelor. Limitările legăturii statice. Metode virtuale.		
8.1.9 Clase virtuale. Clase abstracte. Clase interioare.		
8.1.10 Programare generică.		
8.1.11 Tratarea excepțiilor.		
Bibliografie		
1. Ovidiu Cosma, Programare Orientată pe Obiecte în Limbajul C++, Risoprint 2015;		
2. Claudia Spiricu, Analiza, proiectarea și programarea orientate spre obiecte, Teora 1995;		
3. Herbert Schildt, C++ manual complet, Teora 1997;		
4. Ioan Jurca, Programarea orientată spre obiecte în limbajul C++, 1992.		
5. Resurse WWW		
8.2 Laborator	Metode de predare	Observații

8.2.1	Prezentarea laboratorului și a mediului de programare care va fi folosit pe parcursul semestrului.	laboratoare online: explicație, justificare, dialog, exemplificare, dezbateri, evaluare.	
8.2.2	Alocarea dinamică a memoriei. Transferul prin referință. Parametri cu valori implicite. Realizarea de programe pe calculator.		
8.2.3	Definirea unui set de clase, care vor fi actualizate la fiecare laborator pe baza elementelor prezentate la curs: O clasă pentru reprezentarea punctelor din plan, o listă FIFO, o listă LIFO, un arbore binar și o clasă pentru reprezentarea datelor personale.		
8.2.4	Definirea unui set de constructori și a unui destructor pentru fiecare dintre clasele de studiu. Constructorul implicit.		
8.2.5	Se vor demonstra situațiile în care nu este corectă inițializarea prin atribuire. Definirea constructorului de copiere pentru fiecare clasă din setul de studiu, dacă este necesar. Constructorul de copiere implicit.		
8.2.6	Membrii statici ai claselor. Se va adăuga câte un câmp static și câte o metoda statică în fiecare clasă. Se vor revizui metodele existente.		
8.2.7	Încapsularea datelor. Funcții prietene. Metode prietene. Clase prietene. Refacerea claselor din setul de studiu.		
8.2.8	Clase cu câmpuri obiecte. Supradefinirea operatorilor +, ==, = pentru clasele din setul de studiu. Se va vizualiza apelul constructorilor în cazul transferului prin valoare. Se va scrie o variantă bazată pe transfer prin referință. Se va scrie o variantă bazată pe funcții prietene, unde e posibil.		
8.2.9	Supradefinirea operatorilor [], new, delete, ++.		
8.2.10	Conversii de tip definite prin program prin constructori și prin supradefinirea operatorului cast.		
8.2.11	Derivarea claselor. Înlănțuirea constructorilor și a destructorilor. Derivarea claselor din setul de studiu. Realizarea unei aplicații de desenare a unor figuri geometrice simple.		
8.2.12	Clase abstracte. Redefinirea metodelor. Limitările legăturii statice. Metode virtuale.		
8.2.13	Pointeri către membrii claselor. Clase virtuale. Vizualizarea apelului constructorilor.		
8.2.14	Șabloane și funcții generice. Definirea unei variante generice pentru clasele din setul de studiu. Crearea unei liste FIFO de date personale și a unei stive de obiecte geometrice.		
Bibliografie			
<ol style="list-style-type: none"> Ovidiu Cosma, Programare Orientată pe Obiecte în Limbajul C++, Risoprint 2015; Claudia Spircu, Analiza, proiectarea și programarea orientate spre obiecte, Teora 1995; Herbert Schildt, C++ manual complet, Teora 1997; Ioan Jurca, Programarea orientată spre obiecte în limbajul C++, 1992. Resurse WWW 			

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

- Se asigură cunoștințele necesare de programare orientată pe obiecte, într-un limbaj solicitat de către majoritatea ofertanților de locuri de muncă în domeniul programării calculatoarelor.
- Competențele dobândite la disciplină permit absolvenților să lucreze ca:
Programator, Analist, Dezvoltator software de sistem, Dezvoltator de aplicații, Inginer specialist în asigurarea calității software și testare, Inginer de sisteme software, Profesor în învățământul liceal și postliceal.

10. Evaluare (prezența fizică / online)

