FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Tehnică din Cluj-Napoca, Centrul Universitar Nord Baia Mare
1.2 Facultatea	Ştiinţe
1.3 Departamentul	Matematică şi Informatică
1.4 Domeniul de studii	Matematică
1.5 Ciclul de studii	**Licenţă**
1.6 Specializarea / Programul de studii	**Matematica Informatica**
1.7 Forma de învăţământ	IF – învăţământ cu frecvenţă
1.8 Codul disciplinei	SMAIL511

2. Date despre disciplină

2.1 Denumirea disciplinei	**Programare Orientata pe Obiecte I**						
2.2 Aria de conţinut							
2.3 Responsabil de curs	Conf. dr. ing. Cosma Ovidiu						
2.4 Titularul activităţilor de seminar / laborator / proiect	Asist. Univ. drd. Ţicală Cristina						
2.4 Anul de studii	**1**	2.5 Semestrul	**2**	2.6 Tipul de evaluare	**C**	2.7 Regimul disciplinei	**Op**

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 învăţământ	56	din care: 3.5 curs	28	3.6 seminar / laborator	28
Distribuţia fondului de timp					ore
Studiul după manual, suport de curs, bibliografie şi notiţe					18
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren					18
Pregătire seminarii / laboratoare, teme, referate, portofolii şi eseuri					18
Tutoriat					
Examinări					2
Alte activităţi elaborare de programe					18

3.7 Total ore studiu individual	**74**
3.8 Total ore pe semestru	**130**
3.9 Numărul de credite	**5**

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	Promovarea examenului la disciplina Programarea Procedurală II
4.2 de competenţe	Rezultă din 4.1.

5. Condiţii (acolo unde este cazul)

5.1. de desfăşurare a cursului	Sală de curs cu tablă, videoproiector.

5.2. de desfăşurare a seminarului / laboratorului / proiectului	Laborator cu calculatoare PC conectate la Internet, 4GB memorie RAM, medii de programare Code::Blocks, Visual Studio 2015, videoproiector.

6. Competenţele specifice acumulate

Competenţe profesionale	Definirea conceptelor, metodelor şi instrumentelor aplicate în ingineria software, Cunoaşterea paradigmelor programării orientate pe obiecte, Elaborarea şi analiza unor algoritmi în manieră obiectuală, Programarea în limbaje orientate pe obiecte (C++), Cunoaşterea unor medii de programare moderne, Conceperea unor modelelor simple, descrierea şi implementarea lor într-un limbaj orientat pe obiecte.
Competenţe transversle	Manifestarea unei atitudini responsabile faţă de domeniul ştiinţific, Respectarea regulilor de munca organizată şi eficientă, Selectarea eficienta a resurselor informaţionale, Utilizarea eficientă a surselor de formare profesională, Valorificarea creativă a propriului potenţial, Utilizarea unor metode şi tehnici eficiente de învăţare, Dezvoltare a capacităţilor de valorificare a cunoştinţelor, Respectarea principiilor şi a normelor de etică profesională, Elaborarea proiectului propriu de dezvoltare profesională, de adaptare la cerinţele unei societăţi dinamice.

7. Obiectivele disciplinei (reieşind din grila competenţelor specifice acumulate)

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

8. Conţinuturi

8.1 Curs	Metode de predare	Observaţii
8.1.1 Paradigmele programării orientate pe obiecte.	expunere, algoritmizare, explicatie, problematizare, exemple, demonstraţie didactică.	
8.1.2 Trecerea de la C la C++.Variabile referinţă, transferul prin referinţă. Supradefinirea funcţiilor. Funcţii inline. Parametri cu valori implicite. Operatorii new şi delete.		
8.1.3 Clase şi obiecte. Încapsularea datelor. Constructori şi destructori. Autoreferinţa. Operaţii cu obiecte. Structuri şi uniuni.		
8.1.4 Constructorul de copiere. Membri statici. Funcţii şi 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 si 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 excepţiilor.		
Bibliografie 1. Ovidiu Cosma, Programare Orientată pe Obiecte în Limbajul C++, Risoprint 2015; 2. Claudia Spircu, Analiza, proiectarea şi 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	Observaţii
8.2.1 Prezentarea laboratorului şi a mediului de programare care va fi folosit pe parcursul semestrului.	explicaţie, justificare, dialog, exemplificare, dezbatere, evaluare.	
8.2.2 Alocarea dinamică a memoriei. Transferul prin referinţă. 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 şi o clasă pentru reprezentarea datelor personale.		
8.2.4 Definirea unui set de constructori şi a unui destructor pentru fiecare dintre clasele de studiu. Constructorul implicit.		
8.2.5 Se vor demonstra situaţiile în care nu este corectă iniţializarea 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 si cate o metoda statica în fiecare clasa. Se vor revizui metodele existente.		
8.2.7 Încapsularea datelor. Funcţii 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 constuctorilor in cazul transferului prin valoare. Se va scrie o varianta bazata pe transfer prin referinţă. Se va scrie o varianta bazata pe funcţii prietene, unde e posibil.		
8.2.9 Supradefinirea operatorilor [ ], new, delete, ++.		
8.2.10 Conversii de tip definite prin program prin constructori şi prin supradefinirea operatorului cast.		
8.2.11 Derivarea claselor. Înlănţuirea constructorilor şi a destructorilor. Derivarea claselor din setul de studiu. Realizarea unei aplicaţii 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 Şabloane şi funcţii generice. Definirea unei variante generice pentru clasele din setul de studiu. Crearea unei liste FIFO de date personale si a unei stive de obiecte geometrice.		
Bibliografie 1. Ovidiu Cosma, Programare Orientată pe Obiecte în Limbajul C++, Risoprint 2015; 2. Claudia Spircu, Analiza, proiectarea şi 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

9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatorilor reprezentativi din domeniul aferent programului

- Se asigură cunoştinţele necesare de programare orientată pe obiecte, într-un limbaj solicitat de către majoritatea ofertanţilor de locuri de muncă în domeniul programarii calculatoarelor.
- Competenţele dobândite la disciplină permit absolvenţilor să lucreze ca:
  Programator, Analist, Dezvoltator software de sistem, Dezvoltator de aplicaţii, Inginer specialist în asigurarea calităţii software şi testare, Inginer de sisteme software, Profesor în învăţământul liceal şi postliceal.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Prezenţa la curs		10%
	Lucrare scrisă	Examen scris	50%
10.5 Laborator	Activitatea la laborator	Evaluare continuă prin observare sistematică, proiecte individuale.	20%
	Proba practică, realizarea unor aplicaţii folosind mediul de dezvoltare de la laborator	Evaluare practică.	20%
10.6 Standard minim de performanţă			
• Cunoaşterea noţiunilor fundamentale prezentate la curs, care este echivalentă cu promovarea examenului scris. • Realizarea activităţilor de la laborator la un nivel satisfăcător.			

Data completării
09.2016

Titular de curs
Conf. dr. ing. Cosma Ovidiu

Titular de laborator
Asist. Univ. drd. Ţicală Cristina

Data avizării în Departament
09.2016

Director Departament
Prof. dr. Vasile Berinde