

FIȘA DISCIPLINEI

PROGRAMAREA CALCULATOARELOR II

1. Date despre program

1.1. Instituția de învățământ	Universitatea „1 Decembrie 1918”
1.2. Facultatea	de Științe
1.3. Departamentul	de Științe Exacte și Inginerești
1.4. Domeniul de studii	Inginerie electronica si telecomunicatii
1.5. Ciclul de studii	Licență
1.6. Programul de studii	Electronica aplicata

2. Date despre disciplină

2.1. Denumirea disciplinei	Programarea Calculatoarelor II		2.2. Cod disciplină	EA2102			
2.3. Titularul activității de curs	Rotar Corina						
2.4. Titularul activității de seminar	Incze Arpad						
2.5. Anul de studiu	II	2.6. Semestrul	I	2.7. Tipul de evaluare (E/C/VP)	C	2.8. Regimul disciplinei (O – obligatorie, Op – opțională, F – facultativă)	O

3. Timpul total estimat

3.1. Numar ore pe saptamana	3	din care: 3.2. curs	2	3.3. seminar/laborator	1
3.4. Total ore din planul de învățământ	42	din care: 3.5. curs	28	3.6. seminar/laborator	14
Distribuția fondului de timp					Ore
Studiul după manual, suport de curs, bibliografie și notițe					50
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					25
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					20
Tutoriat					-
Examinări					28
Alte activități					-

3.7 Total ore studiu individual	81
3.8 Total ore din planul de învățământ	42
3.9 Total ore pe semestru	123
3.10 Numărul de credite	3

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	1. Programare imperativa si procedurala
4.2. de competențe	<ul style="list-style-type: none"> - C1.1 Descrierea adecvată a - paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic - C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	<i>Sala dotata cu videoproiector/tabla</i>
5.2. de desfășurarea a seminarului/laboratorului	<i>Laboratoare – calculatoare dotate cu: Visual Studio 2010, BorlandC, acces Internet.</i>

6. Competențe specifice acumulate

Competențe profesionale	<p>C3. Aplicarea cunoștințelor, conceptelor și metodelor de bază privitoare la arhitectura sistemelor de calcul, microprocesoare, microcontrolere, limbaje și tehnici de programare</p> <p>C1.1 Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic.</p> <p>C3.2 Utilizarea unor limbaje de programare de uz general și specifice aplicațiilor cu microprocesoare și microcontrolere; explicarea funcționării unor sisteme de control automat care folosesc aceste arhitecturi și interpretarea rezultatelor experimentale</p> <p>C3.3 Rezolvarea problemelor practice concrete care includ elemente de structuri de date și algoritmi, programare și utilizare de microprocesoare sau microcontrolere</p> <p>C3.4 Elaborarea de programe într-un limbaj de programare general și/sau specific, pornind de la specificarea cerințelor și până la execuție, depanare și interpretarea rezultatelor în corelație cu procesorul utilizat</p> <p>C3.5 Realizarea de proiecte care implică componente hardware (procesoare) și software (programare)</p>
Competențe transversale	Nu e cazul

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

7.1 Obiectivul general al disciplinei	<i>Dezvoltarea capacității studentului de a dezvolta aplicații software dedicate rezolvării problemelor. Dezvoltarea abilităților de a concepe tipuri abstracte de date și bibliotecile aferente. Crearea unui stil de programare orientat obiect riguros și eficient</i>
7.2 Obiectivele specifice	<i>Dezvoltarea abilității studentului de a gestiona eficient informațiile prin tipuri abstracte de date și de a concepe în mod riguros metode de prelucrare a acestora. Intocmirea unei documentații coerente pe marginea aplicațiilor de complexitate medie.</i>

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Paradigma programării orientate obiect. Concepte de baza.	<i>Prelegere, discutii</i>	
2. Programarea prin abstractizarea datelor. Particularități C++.	<i>Prelegere, discutii</i>	
3. Clase și obiecte. Date și funcții membre. Specificatori de acces	<i>Prelegere, discutii</i>	
4. Constructori. Destructorii. Constructorul de copiere	<i>Prelegere, discutii</i>	
5. Elemente statice și const ale claselor.	<i>Prelegere, discutii</i>	
6. Elemente friend. Supraincercarea operatorilor binari.	<i>Prelegere, discutii</i>	
7. Supraincercarea operatorilor (II).	<i>Prelegere, discutii</i>	
8. Conversii.	<i>Prelegere, discutii</i>	
9. Clase derivate, Clase de baza.	<i>Prelegere, discutii</i>	

10. Mostenirea. Mostenirea multipla. Constructori, destructori si mostenire	<i>Prelegere, discutii</i>	
11. Funcții virtuale	<i>Prelegere, discutii</i>	
12. Polimorfismul.	<i>Prelegere, discutii</i>	
13. Clase generice.		
14. Exceptii. Intrari iesiri standard.		
8.2 Bibliografie		
<ol style="list-style-type: none"> 1. Bruce Eckel, Thinking in C++, manual online. 2. Bjarne Stroustrup, The C++ Programming Language, Addison Wesley, 1997. 3. H. Schildt: C++ manual complet, Teora, 2000. 4. Peter Muller: Introduction to Object-Oriented Programming Using C++, resurse electronice. 		
Seminar-laborator		
Instruire NTSM. Considerații de specificare și implementare a POO	Lucrare practica de laborator	
Clasa ca tip abstract de date. Implementarea clasei în C++.	Lucrare practica de laborator	
Clasa. Structură. Componente: attribute, metode. Exemple	Lucrare practica de laborator	
Domeniul public, private, protected. Exemple de aplicații.	Lucrare practica de laborator	
Constructorii și destructorii. Aplicații.	Lucrare practica de laborator	
Operatori. Supraîncărcarea operatorilor.	Lucrare practica de laborator	
Utilizarea mediilor vizuale în POO. Visual Studio .NET, C#. Aplicații consolă.	Lucrare practica de laborator	
Clase standard și clase utilizator. Definierea claselor în C#.	Lucrare practica de laborator	
Mostenirea. Clase friend. Exemple în C++ comparativ cu C#.	Lucrare practica de laborator	
Metode statice și virtuale. Legare statică și dinamică. Proiectarea și implementarea metodelor virtuale.	Lucrare practica de laborator	
Programarea aplicațiilor Windows folosind clase predefinite în C#.	Lucrare practica de laborator	
Polimorfism. Exemple.	Lucrare practica de laborator	
Bibliografie		
<ol style="list-style-type: none"> 1. Bruce Eckel, Thinking in C++, manual online. 2. Bjarne Stroustrup, The C++ Programming Language, Addison Wesley, 1997. 3. H. Schildt: C++ manual complet, Teora, 2000. 4. Peter Muller: Introduction to Object-Oriented Programming Using C++, resurse electronice. 		

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

- Nu e cazul. Disciplina Programare orientata obiect este o disciplina fundamentala obligatorie in planul de invatamant al specializarii Informatica. Conținutul disciplinei este conceput in scopul formarii si dezvoltării gândirii algoritmice a studentului din anul II.

2. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<i>Evaluare finala</i>	<i>Examen scris</i>	60%
	-	-	-
10.5 Seminar/laborator	<i>Verificare pe parcurs</i>	<i>Portofoliu de lucrări practice de laborator; colocviu; activitate laborator</i>	40%
	-	-	-
10.6 Standard minim de performanță:			
Implementarea și documentarea de unități de program în limbaje de programare orientate obiect și folosirea eficientă a mediilor de programare			

Observatii: Recuperarea laboratoarelor se poate face in regim de consultații in timpul semestrului. De asemenea, in cazuri bine motivate, recuperarea orelor de laborator se mai poate face prin prezentarea de către student a portofoliului complet de lucrari practice - in ultima saptamana din semestrul II, in orele de consultații ale cadrului didactic titular.

Data completării

.....

Semnătura titularului de curs

.....

Semnătura titularului de seminar

.....

Data avizării în catedră

.....

Semnătura director de departament

.....