

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență
1.6 Programul de studiu / Calificarea	Informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare orientată obiect						
2.2 Titularul activităților de curs	Lect. dr. CZIBULA Istvan Gergely						
2.3 Titularul activităților de seminar							
2.4 Anul de studiu	1	2.5 Semestrul	4	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	5	Din care: 3.2 curs	2	3.3 seminar/laborator	1sem + 2 lab
3.4 Total ore din planul de învățământ	70	Din care: 3.5 curs	28	3.6 seminar/laborator	14+ 28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					19
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					24
Tutoriat					14
Examinări					18
Alte activități:					-
3.7 Total ore studiu individual			105		
3.8 Total ore pe semestru			175		
3.9 Numărul de credite			7		

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Fundamentele Programării, Structuri de date și algoritmi	
4.2 de competențe	Cunoștințe medii de programare într-un limbaj de programare de nivel înalt	

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none"> Sală de curs cu videoproiector
-------------------------------	--

5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> Sală de laborator cu calculatoare dotate cu limbajul de programare C++ și biblioteca QT
--	---

6. Competențele specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> Cunoașterea și înțelegerea conceptelor specifice programării orientate obiect Înțelegerea rolului moștenirii, polimorfismului, legării dinamice și structurilor dinamice în realizarea de cod reutilizabil. Abilități de programare în limbajul de programare C++ și crearea de interfețe grafice utilizator folosind QT .
Competențe transversale	<ul style="list-style-type: none"> Abilitatea de a aplica conceptele, principiile și tehnicile însușite în rezolvarea problemelor din lumea reală. Executarea responsabilă a lucrărilor de laborator. Aplicarea regulilor de muncă riguroasă și eficientă. Manifestarea unor atitudini responsabile față de domeniul științific și didactic. Respectarea principiilor și a normelor de etică profesională.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> Să deprindă studentul cu proiectare orientată obiect a problemelor de scară mică/mijlocie și învățarea limbajului de programare C++ și crearea de interfețe grafice utilizator în QT.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> Demonstrarea diferenței dintre proiectarea imperativă tradițională și proiectarea orientată obiect. Explicarea structurilor de tip clasă ca fiind componente fundamentala, modulare. Înțelegerea rolului moștenirii, polimorfismului, legării dinamice și a structurilor generice în realizarea codului reutilizabil. Explicarea și utilizarea dezvoltării bazate pe funcționalități, dezvoltarea bazată pe testare, utilizarea aserțiunilor formale și tratarea excepțiilor. Scrierea de programe de scară mică/mijlocie folosind C++ și QT. Utilizarea claselor scrise de alți programatori în dezvoltarea sistemelor.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Pradigma de programare orientată obiect. <ul style="list-style-type: none"> Elemente de bază ale limbajului C.. Elemente lexicale. Operatori. Conversii. Tipuri de date. Variabile. Constante. Domeniul de vizibilitate și durata de viață a 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Exemple Demonstrația didactică 	

<p>variabilelor. Spații de nume.</p> <ul style="list-style-type: none"> • Declararea și definirea funcțiilor. Supraîncărcarea funcțiilor. Funcții inline. 		
<p>2. Programare modulară în C++.</p> <ul style="list-style-type: none"> • Funcții. Parametri. • Fișiere header. Biblioteci. • Implementarea modulară a TAD-urilor.. • Utilizarea pointerului void pentru obținerea genericității. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Exemple • Demonstrația didactică 	
<p>3. Tipuri de date derivate și tipuri definite de utilizator, alocare dinamică în C++.</p> <ul style="list-style-type: none"> • Tipuri de date: vectori și structuri. • Tipuri de date: pointeri și referințe. • Alocarea și dealocarea memoriei. • Pointeri la funcții și pointeri spre void. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>4. Programare orientată obiect în C++.</p> <ul style="list-style-type: none"> • Clase și obiecte. • Membri unei clase. Modificatori de acces. • Constructori/destructori. • Diagrame UML pentru clase (membri, acces). 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>5. Moștenire</p> <ul style="list-style-type: none"> • Moștenire simplă. Clase derivate. • Principiul substituției. • Supraîncărcarea metodelor. • Moștenire multiplă. • Relații de specializare/generalizare – reprezentări UML. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>6. Operații de intrare/ieșire.</p> <ul style="list-style-type: none"> • Stream-uri I/O. Ierarhia de clase I/O. • Formatare. Manipulatori. • Fișiere text. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>7. QT Toolkit.</p> <ul style="list-style-type: none"> • Instrumente și module QT. • Instalare QT. • Exemple. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>8. QT</p> <ul style="list-style-type: none"> • Semnale și slot-uri. • QWidget. • Exemple. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>9. Utilizarea designerului QT în Eclipse (1)</p> <ul style="list-style-type: none"> • Proiectare GUI • Studiu de caz. Detalii comenzi – Produse. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>10. Utilizarea designerului QT în Eclipse (2)</p> <ul style="list-style-type: none"> • Studiu de caz. Detalii comenzi – Produse. • Șablonul MVC. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>11. Șabloane de proiectare.</p> <ul style="list-style-type: none"> • Șabloane de proiectare creaționale, structurale, comportamentale. • Exemple • Biblioteca STL 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	

<ul style="list-style-type: none"> • Clase de tip container. 		
12. Biblioteca STL <ul style="list-style-type: none"> • Iteratori STL. • Algoritmi STL. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
13. Aplicație POS (Point Of Sale) <ul style="list-style-type: none"> • Șabloane de proiectare Façade, Strategy. • Șablonul de proiectare Composite. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
14. Recapitulare	<ul style="list-style-type: none"> • Expunerea interactivă • Conversația 	
Bibliografie		
<ol style="list-style-type: none"> 1. B. Stroustup, The C++ Programming Language, Addison Wesley, 1998. 2. Bruce Eckel, Thinking in C++, www.bruceeckel.com 3. Alexandrescu, Programarea moderna in C++. Programare generica si modele de proiectare aplicate, Editura Teora, 2002 4. M. Frentiu, B. Parv, Elaborarea programelor. Metode si tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994. 5. E. Horowitz, S. Sahni, D. Mehta, Fundamentals of Data Structures in C++, Computer Science Press, Oxford, 1995. 6. K.A. Lambert, D.W. Nance, T.L. Naps, Introduction to Computer Science with C++, West Publishing Co., New-York, 1996. 7. L. Negrescu, Limbajul C++, Ed. Albastra, Cluj-Napoca 1996. 		
8.2 Seminar	Metode de predare	Observații
		Seminarul este structurat sub forma a 2 ore din 2 în 2 săptămâni.
1. Probleme simple în C. Funcții și parametric. Variabile locale și globale. Vizibilitate. Vectori (uni și multidimensionali) și structuri.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
2. TAD de tip container cu elemente generice (void*): reprezentări vizibile și ascunse.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
3. Clase. Clase simple. Supraîncărcarea operatorilor. Clase cu obiecte ca date membre.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
4. Clase de tip vector dinamic și iteratori. Moștenire.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
5. Clase abstracte și interfețe. Polimorfism.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
6. Clase: template și excepții	<ul style="list-style-type: none"> • Expunerea interactivă 	

	<ul style="list-style-type: none"> • Explicația • Conversația • Demonstrația didactică • Exemple 	
7. Probleme complexe implementate pe baza diagramelor UML. Șabloane de proiectare. Pregătire pentru examenul scris.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
8.3 Laborator	Metode de predare	Observații
		<ul style="list-style-type: none"> • Laboratorul este structurat sub forma a 2 ore săptămânal. • Documentațiile de laborator și programele se predau în săptămâna următoare celei în care s-a dat tema.
1. Instalare MinGW și Eclipse CDT. Specificare, proiectare și implementare probleme simple în C/C++. Aspecte generale C/C++.	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
2. Programare modulară în C++	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
3. Proces de dezvoltare “feature driven”	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
4. Proces de dezvoltare “feature driven”	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
5. Proces de dezvoltare “feature driven”	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
6. Arhitecturi stratificate	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
7. Arhitecturi stratificate	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
8. Arhitecturi stratificate	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația 	

	<ul style="list-style-type: none"> • Conversația • Modelarea 	
9. Fișiere text	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
10. GUI folosind QT	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
11. Repository.	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
12. Containere, iteratori și algoritmi STL	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
13. Predare laboratoare (a se vedea observația)	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
14. Predare laboratoare (a se vedea observația)	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
Bibliografie		
<ol style="list-style-type: none"> 1. B. Stroustup, The C++ Programming Language, Addison Wesley, 1998. 2. Bruce Eckel, Thinking in C++, www.bruceeckel.com 3. Alexandrescu, Programarea moderna in C++. Programare generica si modele de proiectare aplicate, Editura Teora, 2002 4. M. Frentiu, B. Parv, Elaborarea programelor. Metode si tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994. 5. E. Horowitz, S. Sahni, D. Mehta, Fundamentals of Data Structures in C++, Computer Science Press, Oxford, 1995. 6. K.A. Lambert, D.W. Nance, T.L. Naps, Introduction to Computer Science with C++, West Publishing Co., New-York, 1996. 7. L. Negrescu, Limbajul C++, Ed. Albastra, Cluj-Napoca 1996. 		

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

- Conținutul disciplinei este în concordanță cu recomandările ACM pentru domeniul informatică.
- Cursul există în programul de studii al universităților importante din țară și străinătate.
- Conținutul disciplinei este considerat de companiile software ca important pentru a asigura cunoștințe medii de programare orientată obiect.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
----------------	---------------------------	-------------------------	------------------------------

10.4 Curs	<ul style="list-style-type: none"> Corectitudinea și completitudinea cunoștințelor acumulate și capacitatea de a proiecta și implementa programe C++ 	Examen scris (în sesiunea regulată)	30%
10.5 Seminar/ Laborator	<ul style="list-style-type: none"> Capacitatea de a proiecta, testa și depana programe C++ folosind QT 	Evaluare practică (în sesiunea regulată)	40%
	<ul style="list-style-type: none"> Corectitudinea programelor C++ și a documentațiilor de laborator 	-documentații -portofoliu -observare continuă	30%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> Fiecare student trebuie să demonstreze că a atins un nivel acceptabil de cunoaștere și înțelegere a domeniului, că este capabil să exprime cunoștințele într-o formă coerentă, că are capacitatea de a stabili anumite conexiuni și de a utiliza cunoștințele în rezolvarea unor probleme în limbajul de programare C++. Pentru promovare este necesar notele pe toate activitățile să fie minim 5 și nota finală să fie minim 5. 			

Data completării

30.04.2013

Semnătura titularului de curs

Lect. dr. Istvan Czibula

Semnătura titularului de seminar

Lect. dr. Istvan Czibula

Data avizării în departament

Semnătura directorului de departament

Prof. dr. Bazil Pârv