

## FIȘA DISCIPLINEI

### 1. Date despre program

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

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Programare orientată obiect</b>						
2.2 Titularul activităților de curs	<b>Schutten Markus</b>						
2.3 Titularul activităților de seminar	<b>Gaceanu Radu</b>						
2.4 Anul de studiu	<b>1</b>	2.5 Semestrul	<b>2</b>	2.6. Tipul de evaluare	<b>E</b>	2.7 Regimul disciplinei	<b>Obligatorie</b>

### 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	1se m+ 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					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					20
Tutoriat					14
Examinări					6
Alte activități: .....					-
3.7 Total ore studiu individual	80				
3.8 Total ore pe semestru	150				
3.9 Numărul de credite	6				

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

#### 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	<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>C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază</p> <p>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</p> <p>C1.4 Testarea unor aplicații pe baza unor planuri de test</p> <p>C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente</p>
<b>Competențe transversale</b>	<p>CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională</p> <p>CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională</p>

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

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"><li>Să deprindă studentul cu proiectare orientată obiect a problemelor de scară mică/mijlocie și învățarea limbajului de programare C++ și a bibliotecii QT.</li></ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"><li>Demonstrarea diferenței dintre proiectarea imperativă tradițională și proiectarea orientată obiect.</li><li>Explicarea structurilor de tip clasă ca fiind componente fundamentala, modulare.</li><li>Înțelegerea rolului moștenirii, polimorfismului, legării dinamice și a structurilor generice în realizarea codului reutilizabil.</li><li>Explicarea și utilizarea strategiilor de programare DEFENSIVE, utilizarea aserțiunilor formale și tratarea excepțiilor.</li></ul>

- 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
<b>1. Paradigma de programare orientată obiect.</b> <ul style="list-style-type: none"> <li>• Elemente de bază ale limbajului C..</li> <li>• Elemente lexicale. Operatori. Conversii.</li> <li>• Tipuri de date. Variabile. Constante.</li> <li>• Domeniul de vizibilitate și durata de viață a variabilelor. Spații de nume.</li> <li>• Declararea și definirea funcțiilor. Supraîncărcarea funcțiilor. Funcții inline.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Exemple</li> <li>• Demonstrația didactică</li> </ul>	
<b>2. Programare modulară în C++.</b> <ul style="list-style-type: none"> <li>• Funcții. Parametrii.</li> <li>• Fișiere header. Biblioteci.</li> <li>• Implementarea modulară a TAD-urilor..</li> <li>• Utilizarea pointerului void pentru obținerea genericității.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Exemple</li> <li>• Demonstrația didactică</li> </ul>	
<b>3. Tipuri de date derivate și tipuri definite de utilizator, alocare dinamică în C++.</b> <ul style="list-style-type: none"> <li>• Tipuri de date: vectori și structuri.</li> <li>• Tipuri de date: pointeri și referințe.</li> <li>• Alocarea și dealocarea memoriei.</li> <li>• Pointeri la funcții și pointeri spre void.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>4. Programare orientată obiect în C++.</b> <ul style="list-style-type: none"> <li>• Clase și obiecte.</li> <li>• Membri unei clase. Modificatori de acces.</li> <li>• Constructori/destructori.</li> <li>• Diagrame UML pentru clase (membri, acces).</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>5. Moștenire</b> <ul style="list-style-type: none"> <li>• Moștenire simplă. Clase derivate.</li> <li>• Principiul substituției.</li> <li>• Supraîncărcarea metodelor.</li> <li>• Moștenire multiplă.</li> <li>• Relații de specializare/generalizare – reprezentări UML.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>6. Operații de intrare/ieșire.</b> <ul style="list-style-type: none"> <li>• Stream-uri I/O. Ierarhia de clase I/O.</li> <li>• Formatare. Manipulatori.</li> <li>• Fișiere text.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația</li> </ul>	

	didactică	
<b>7. QT Toolkit.</b> <ul style="list-style-type: none"> <li>• Instrumente și module QT.</li> <li>• Instalare QT.</li> <li>• Exemple.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>8. QT</b> <ul style="list-style-type: none"> <li>• Semnale și slot-uri.</li> <li>• QWidget.</li> <li>• Exemple.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>9. Utilizarea designerului QT în Eclipse (1)</b> <ul style="list-style-type: none"> <li>• Proiectare GUI</li> <li>• Studiu de caz. Detalii comenzi – Produse.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>10. Utilizarea designerului QT în Eclipse (2)</b> <ul style="list-style-type: none"> <li>• Studiu de caz. Detalii comenzi – Produse.</li> <li>• Șablonul MVC.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>11. Șabloane de proiectare.</b> <ul style="list-style-type: none"> <li>• Șabloane de proiectare creaționale, structurale, comportamentale.</li> <li>• Exemple</li> <li>• Biblioteca STL</li> <li>• Clase de tip container.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>12. Biblioteca STL</b> <ul style="list-style-type: none"> <li>• Iteratorii STL.</li> <li>• Algoritmi STL.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>13. Aplicație POS (Point Of Sale)</b> <ul style="list-style-type: none"> <li>• Șabloane de proiectare Façade, Strategy.</li> <li>• Șablonul de proiectare Composite.</li> </ul>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Demonstrația didactică</li> </ul>	
<b>14. Recapitulare</b>	<ul style="list-style-type: none"> <li>• Expunerea interactivă</li> <li>• Conversația</li> </ul>	

## Bibliografie

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. Alabastra, Cluj-Napoca 1996.

## In limba germana:

1. G. Goos, W. Zimmermann, Objektorientiertes Programmieren und Algorithmen, Springer, Berlin,

Heidelberg, New York, 2006.

2. Pöttsch-Heffter, A., Konzepte objektorientierter Programmierung, Springer, Berlin, Heidelberg, 2009.
3. Küchlin, W, Weber, A., Einführung in die Informatik, Objektorientiertes Programmieren mit Java, Springer, Berlin, Heidelberg, New York, 2004.

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"><li>• Expunerea interactivă</li><li>• Explicația</li><li>• Conversația</li><li>• Demonstrația didactică</li><li>• Exemple</li></ul>	
2. TAD de tip container cu elemente generice (void*): reprezentări vizibile și ascunse.	<ul style="list-style-type: none"><li>• Expunerea interactivă</li><li>• Explicația</li><li>• Conversația</li><li>• Demonstrația didactică</li><li>• Exemple</li></ul>	
3. Clase. Clase simple. Supraîncărcarea operatorilor. Clase cu obiecte ca date membre.	<ul style="list-style-type: none"><li>• Expunerea interactivă</li><li>• Explicația</li><li>• Conversația</li><li>• Demonstrația didactică</li><li>• Exemple</li></ul>	
4. Clase de tip vector dinamic și iteratori. Moștenire.	<ul style="list-style-type: none"><li>• Expunerea interactivă</li><li>• Explicația</li><li>• Conversația</li><li>• Demonstrația didactică</li><li>• Exemple</li></ul>	
5. Clase abstracte și interfețe. Polimorfism.	<ul style="list-style-type: none"><li>• Expunerea interactivă</li><li>• Explicația</li><li>• Conversația</li><li>• Demonstrația didactică</li><li>• Exemple</li></ul>	
6. Clase: template și excepții	<ul style="list-style-type: none"><li>• Expunerea interactivă</li><li>• Explicația</li><li>• Conversația</li><li>• Demonstrația didactică</li><li>• Exemple</li></ul>	
7. Probleme complexe implementate pe baza diagramelor UML. Șabloane de proiectare. Pregătire pentru examenul scris.	<ul style="list-style-type: none"><li>• Expunerea interactivă</li><li>• Explicația</li><li>• Conversația</li><li>• Demonstrația didactică</li><li>• Exemple</li></ul>	

8.3 Laborator	Metode de predare	Observații
		<ul style="list-style-type: none"> <li>• Laboratorul este structurat sub forma a 2 ore săptămânal.</li> <li>• Documentațiile de laborator se predau în săptămâna următoare celei în care s-a dat tema, iar programele se predau după două săptămâni.</li> </ul>
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"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
2. Programare modulară în C++	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
3. Proces de dezvoltare “feature driven”	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
4. Proces de dezvoltare “feature driven”	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
5. Proces de dezvoltare “feature driven”	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
6. Arhitecturi stratificate	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
7. Arhitecturi stratificate	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
8. Arhitecturi stratificate	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
9. Fișiere text	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
10. GUI folosind QT	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> </ul>	

	<ul style="list-style-type: none"> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
11. Repository.	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
12. Containere, iteratori și algoritmi STL	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
13. Predare laboratoare (a se vedea observația)	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	
14. Predare laboratoare (a se vedea observația)	<ul style="list-style-type: none"> <li>• Lucrare de laborator</li> <li>• Explicația</li> <li>• Conversația</li> <li>• Modelarea</li> </ul>	

### Bibliografie

1. B. Stroustup, The C++ Programming Language, Addison Wesley, 1998.
2. Bruce Eckel, Thinking in C++, www.bruceekel.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 des tudy 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"> <li>• Corectitudinea și completitudinea cunoștințelor acumulate și capacitatea de a proiecta și implementa programe C++</li> </ul>	Examen scris (în sesiunea regulată)	40%
10.5 Seminar/ Laborator	<ul style="list-style-type: none"> <li>• Capacitatea de a proiecta, testa și depana programe C++ folosind QT</li> </ul>	Evaluare practică (în sesiunea regulată)	30%

	<ul style="list-style-type: none"> <li>• Corectitudinea programelor C++ și a documentațiilor de laborator</li> </ul>	-documentații -portofoliu -observare continuă	30%
<b>10.6 Standard minim de performanță</b>			
<ul style="list-style-type: none"> <li>• 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++.</li> <li>• Pentru promovare este necesar ca nota finală să fie minim 5.</li> </ul>			

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

13.12.2013

Markus Schutten

Gaceanu Radu

Data avizării în departament

Semnătura directorului de departament

20.12.2013

Prof. dr. Bazil Pârv