

FIŞA 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 română

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare orientată obiect						
2.2 Titularul activităților de curs	Prof. dr. CZIBULA Istvan Gergely						
2.3 Titularul activităților de seminar	Prof. dr. CZIBULA Istvan Gergely						
2.4 Anul de studiu	1	2.5 Semestrul	2	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie
2.8 Cod disciplină	MLR5006						

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					18
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					12
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					8
Tutoriat					7
Examinări					10
Alte activități:					-
3.7 Total ore studiu individual	55				
3.8 Total ore pe semestru	125				
3.9 Numărul de credite	5				

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	Sală de curs cu videoproiector
-------------------------------	--------------------------------

De desfășurare a seminarului/laboratorului	Sală de laborator cu calculatoare dotate cu limbajul de programare C++ și biblioteca QT
--	---

6. Competențele specifice acumulate

Competențe profesionale	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. C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază 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 C1.4 Testarea unor aplicații pe baza unor planuri de test C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente
Competențe transversale	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 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

7. Obiectivele disciplinei (reiesind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	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 fundamentală, 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 exceptiilor. Scrierea de programme 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. Elemente de bază ale limbajului C. <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 variabilelor. Spații de nume. Declararea și definirea funcțiilor. Supraîncărcarea funcțiilor. Funcții inline. 	<ul style="list-style-type: none"> Exponere interactivă Explicația Conversația Exemple Demonstrația didactică 	
2. Programare modulară în C/C++.	<ul style="list-style-type: none"> Exponere interactivă 	

<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"> • Explicația • Conversația • Exemple • Demonstrația didactică 	
3. Tipuri de date derivate și tipuri definite de utilizator, alocare dinamică în C++. <ul style="list-style-type: none"> • Tipuri de date: vectori și structuri. • Tipuri de date: pointeri și referințe. • Gestiuinea memorie in C/C++ • Pointeri la funcții și pointeri spre void. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
4. Programare orientată obiect în C++. <ul style="list-style-type: none"> • Clase și obiecte. • Membrii unei clase. Modificatori de acces. • Constructori/destructori. • Gestiuinea memoriei in C++ (RAII) • Implementarea TAD-urilor in C++ • Diagrame UML pentru clase (membri, acces). 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
5. Elemente de programare generică <ul style="list-style-type: none"> • Functii/clase parametrizate. Mecanismul de template din C++ • Implementare TAD-uri folosind clase parametrizate • Containere și iteratori – biblioteca STL 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
6 Moștenire <ul style="list-style-type: none"> • Moștenire simplă. Clase deriveate. • Prințipiul substituției. • Supraîncarcarea metodelor. • Moștenire multiplă. • Relații de specializare/generalizare – reprezentări UML. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
7 Polimorfism <ul style="list-style-type: none"> • Mostenire, polimorfism • 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 	
8 Ierarhii de clase <ul style="list-style-type: none"> • Exceptii, spatii de nume • Fișiere text. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
9 Interfețe grafice utilizator <ul style="list-style-type: none"> • QT Toolkit: instalare, instrumente si module Qt • Componente grafice utilizator • Layout management 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
10 Elemente de programare bazată pe evenimente <ul style="list-style-type: none"> • Evenimente: Semnale si sloturi Qt • Proiectare GUI • Callback/Observer 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
11 Elemente de programare bazată pe evenimente <ul style="list-style-type: none"> • Componente grafice cu modele 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația 	

<ul style="list-style-type: none"> • Şablonul MVC. • Studiu de caz. Detalii comenzi – Produse. 	<ul style="list-style-type: none"> • Conversaţia • Demonstraţia didactică 	
12. Şabloane de proiectare. <ul style="list-style-type: none"> • Şabloane de proiectare creaţionale, structurale, comportamentale. • Exemple 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică 	
13. Aplicaţie POS (Point Of Sale) <ul style="list-style-type: none"> • Şabloane de proiectare Façade, Strategy. • Şablonul de proiectare Composite. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică 	
14. Recapitulare	<ul style="list-style-type: none"> • Expunerea interactivă • Conversaţia 	

Bibliografie

1. B. Stroustrup, The C++ Programming Language, Addison Wesley, 1998.
2. Bruce Eckel, Thinking in C++, www.bruceeckel.com
3. Alexandrescu, Programarea modernă în C++. Programare generică și modele de proiectare aplicate, Editura Teora, 2002
4. M. Frentiu, B. Parv, Elaborarea programelor. Metode și 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	Observaţii
		Seminarul este structurat sub forma a 2 ore din 2 în 2 săptămâni.
1. Probleme simple în C. Funcţii şi parametri. Variabile locale şi globale. Vizibilitate. Vectori (uni şi multidimensionali) şi structuri.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică • Exemple 	
2. TAD de tip container cu elemente generice (void*): reprezentări vizibile şi ascunse.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică • Exemple 	
3. Clase. Clase simple. Supraîncărcarea operatorilor. Clase cu obiecte ca date membre.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică • Exemple 	
4. Clase de tip vector dinamic şi iteratori. Moştenire. Polimorfism	<ul style="list-style-type: none"> • Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică • Exemple 	
5. Containere, iteratori, algoritmi STL	<ul style="list-style-type: none"> • Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică 	

	<ul style="list-style-type: none"> • Exemple 	
6. Interfete grafice utilizator	<ul style="list-style-type: none"> • Expunerea interactivă • 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. Visual Studio. 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 simple în C	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
3. Programare modulară C/C++	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
4. Proces de dezvoltare “feature driven”, arhitectura stratificată, test driven development	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
5. Proces de dezvoltare “feature driven”, arhitectura stratificată, test driven development	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
6. Programare orientată obiect	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
7. Programare orientată obiect	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
8. Containere, iteratori și algoritmi STL	<ul style="list-style-type: none"> • Lucrare de laborator 	

	<ul style="list-style-type: none"> • Explicația • 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. GUI folosind QT	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
12 Componente cu modele, MVC, Observer	<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

1. B. Stroustrup, The C++ Programming Language, Addison Wesley, 1998.
2. Bruce Eckel, Thinking in C++, www.bruceeckel.com
3. Alexandrescu, Programarea modernă în C++. Programare generică și modele de proiectare aplicată, Editura Teora, 2002
4. M. Frentiu, B. Parv, Elaborarea programelor. Metode și 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 angajaților reprezentativi din domeniul aferent programului

- Conținutul disciplinei este în concordanță cu recomandările ACM pentru domeniul informatică.
- Cursul există în programul de studiu 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 sesiune)	40%
10.5 Seminar/ Laborator	<ul style="list-style-type: none"> Capacitatea de a proiecta, testa și depana programe C++ folosind QT 	Evaluare practică (în sesiune)	30%
	<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 obligatorie prezența la minimum 5 seminarii și 12 laboratoare . Pentru promovare este necesar ca notele pe toate activitățile să fie cel puțin 5 și nota finală să fie cel puțin 5. 			

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

Prof. dr. Czibula Istvan Gergely

Prof. dr. Czibula Istvan Gergely

Data avizării în departament

Semnătura directorului de departament

Conf. dr. Sterca Adrian