

FIȘA DISCIPLINEI

1. Date despre program

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

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare orientată obiect						
2.2 Titularul activităților de curs	Diosan Laura						
2.3 Titularul activităților de seminar	Diosan Laura						
2.4 Anul de studii	1	2.5 Semestrul	2	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatoriu

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

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 învățământ	56	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					35
Tutoriat					7
Examinări					7
Alte activități:					
3.7 Total ore studiu individual		94			
3.8 Total ore pe semestru		150			
3.9 Numărul de credite		6			

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> Algoritmica, structuri de date, statistica
4.2 de competențe	<ul style="list-style-type: none"> Abilități medii de programare într-un limbaj de nivel înalt (orientat obiect)

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none">
5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> Pentru activitatea de laborator este nevoie de calculatoare cu medii de programare avansate

6. Competențele specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> • Cunoasterea, intelegerea si utilizarea conceptelor de baza din Informatica • Abilitatea de a lucra independent si/sau in echipa pentru rezolvarea unor probleme complexe • Capacitatea de a dezvolta programe in limbaje de programare de nivel inalt
Competențe transversale	<ul style="list-style-type: none"> • Cunoasterea si intelegerea principiilor de programare orientate obiect • Modelarea si rezolvarea problemelor din viata reala astfel incat ele sa poata fi rezolvate cu ajutorul programarii orientate obiect • Imbunatatirea abilitatilor de programare: cresterea eficientei aplicatiilor dezvoltate

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Programarea orientata obiect are drept obiectiv insusirea princiilor de baza a programarii cu ajutorul tipurilor abstracte de date (obiectelor).
7.2 Obiectivele specifice	<p>Dupa insusirea materialului prezentat la aceasta disciplina studentii ar trebui:</p> <ul style="list-style-type: none"> • sa poata rezolva probleme de dimensiuni mici si medii intr-o maniera orientata pe obiecte • sa poata evidentia diferenta intre proiectarea functionala traditionala si proiectarea orientata pe obiecte • sa intealega rolul mostenirii, polimorfismului, legarii dinamice si a structurilor generice in dezvoltarea unor programe reutilizabile • sa explice si sa foloseasca diferite strategii de programare referitor la tratarea exceptiilor si asertiuni formale • sa poata scrie programe C++ de dimensiuni mici/medii • in timpul rezolvarii unei probleme sa foloseasca clase scrise de alti programatori • sa intealega si sa foloseasca structurile de date fundamentale: colectii, multimi, tabele, liste, stive, cozi, arbori, grafe

8. Conținuturi

8.1 Curs	Metode de predare	Observații
In prima parte a cursului sunt introduse gradat conceptele programarii orientate pe obiecte. Pentru exemplificari este folosit limbajul C++. Sunt supuse dezbaterilor diferite structuri de date ce au fost prezentate in cursul de introducere in informatica. In partea a doua sunt prezentate subiecte mai avansate de programare C++: ierarhii de clase standard, programarea dirijata de evenimente, componente C++ pentru interfata cu utilizatorul, tratarea exceptiilor.	Expunerea Conversația Problematizarea	

<p>Curs 1</p> <p>Elemente de baza ale limbajului C++.</p> <ul style="list-style-type: none"> - Elemente lexicale. Operatori. Conversii. - Tipuri de date. Variabile. Constante. - Domeniu de vizibilitate si durata de viata a variabilelor. - Spatii de nume. - Instructiuni C++. - Declararea si definitia functiilor in C++. - Supraincercarea functiilor. Functii inline. 		
<p>Curs 2</p> <p>Tipuri de date derivate si utilizator si alocare dinamica in C++.</p> <ul style="list-style-type: none"> - Tipurile tablou si structura. - Tipurile pointer si referinta. - Alocarea si dealocarea memoriei. - Pointeri la functii si pointeri void. <p>Programare modulara in C++.</p> <ul style="list-style-type: none"> - Fisiere header. Biblioteci. - Implementari modulare de tipuri abstracte de date. - Folosirea tipului pointer void pentru obtinerea genericitatii. 	<p>Expunerea Conversația Demonstrația didactică Algoritmizarea</p>	
<p>Curs 3</p> <p>Metoda programarii orientate-obiect in C++.</p> <ul style="list-style-type: none"> - Clase si obiecte. - Membrii unei clase. Specificatori de acces. - Constructori / destructori - Diagrame UML pentru clase (membrii, acces). 	<p>Expunerea Conversația Demonstrația didactică Algoritmizarea Problematizarea</p>	
<p>Curs 4</p> <ul style="list-style-type: none"> - Supraincercarea operatorilor. - Membrii statici. - Functii Friend. 	<p>Expunerea Algoritmizarea Problematizarea</p>	
<p>Curs 5</p> <ul style="list-style-type: none"> - Clase: containere si iteratori <ul style="list-style-type: none"> o Lista simplu inlantuita si lista dublu inlantuita o Stiva, Coada, Tabele de dispersie o Relatia de asociere/agregare intre clase - reprezentare UML 	<p>Expunerea Algoritmizarea Problematizarea</p>	
<p>Curs 6</p> <ul style="list-style-type: none"> - Relatia de mostenire <ul style="list-style-type: none"> o Mostenire simpla. Clase derivate. o Principiul substitutiei. o Suprascrierea metodelor. o Mostenire multipla. o Relatia de specializare/generalizare intre clase - reprezentare UML. 	<p>Expunerea Conversația Algoritmizarea Problematizarea</p>	
<p>Curs 7</p> <ul style="list-style-type: none"> - Clase abstracte si Polimorfism. <ul style="list-style-type: none"> o Metode virtuale. o Legare dinamica. o Mostenire virtuala. o Reutilizare cod (mostenire/compozitie). o Conversii (upcast/downcast). 	<p>Expunerea Demonstrația didactică Algoritmizarea Problematizarea</p>	
<p>Curs 8</p> <ul style="list-style-type: none"> - Proiectare orientata-obiect si proiectare bazata pe interfete. <ul style="list-style-type: none"> o Clase abstracte, interfete. o Reprezentare UML pentru interfete. o Proiectarea orientata-obiect a unei biblioteci de structuri de date. 	<p>Expunerea Conversația Demonstrația didactică Algoritmizarea Problematizarea</p>	

<p>Curs 9</p> <ul style="list-style-type: none"> - Sabloane (Template). <ul style="list-style-type: none"> o Functii template. Clase template. o Reutilizarea codului sursa. 	<p>Expunerea Conversația Algoritmizarea Problematizarea</p>	
<p>Curs 10</p> <ul style="list-style-type: none"> - Operatii de intrare/iesire. <ul style="list-style-type: none"> o Fluxuri de intrare/ iesire. Ierarhii de clase pentru I/O. o Formatare. Manipulatori. o Lucrul cu fisiere. 	<p>Expunerea Conversația Demonstrația didactică Algoritmizarea Problematizarea</p>	
<p>Curs 11</p> <ul style="list-style-type: none"> - Tratarea exceptiilor. <ul style="list-style-type: none"> o Notiunea de exceptie in programare o Tratarea exceptiilor in C++. 	<p>Expunerea Conversația Demonstrația didactică Algoritmizarea Problematizarea</p>	
<p>Curs 12</p> <ul style="list-style-type: none"> - Biblioteca STL - Clase container si iterator. - Folosirea algoritmilor din STL. 	<p>Expunerea Conversația Demonstrația didactică Algoritmizarea Problematizarea</p>	
<p>Curs 13</p> <ul style="list-style-type: none"> - Sabloane de proiectare <ul style="list-style-type: none"> o Dezvoltare bazata pe testare o Instrumente de testare 	<p>Expunerea Conversația Demonstrația didactică Algoritmizarea Problematizarea</p>	
<p>Curs 14</p> <ul style="list-style-type: none"> - Sabloane de proiectare <ul style="list-style-type: none"> o Singleton o Model View Controller o Repository 	<p>Expunerea Conversația Demonstrația didactică Algoritmizarea Problematizarea</p>	
<p>Bibliografie</p> <ol style="list-style-type: none"> 1. A.V. Aho, J.E. Hopcroft, J.D. Ullman, Data Structures and Algorithms, Addison-Wesley Publ., Massachusetts, 1983. 2. R. Andonie, I. Garbacea, Algoritmi fundamentali. O perspectiva C++, Editura Libris, 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. Dan Roman, Ingineria programarii obiectuale, Editura Albastra, Cluj_Napoca, 1996. 9. B. Stroustup, The C++ Programming Language, Addison Wesley, 1998. 10. Bruce Eckel, Thinking in C++, www.bruceeckel.com 		

8.2 Seminar / laborator	Metode de predare	Observații
<p>S 1. Probleme simple in C++. Functii. Variabile (locale si globale) si vizibilitatea lor. Vectori (uni si multidimensionali) si structuri.</p> <p>L 1. Strategii și metode inteligente de rezolvare a jocurilor</p>	<p>Conversația Algoritmizarea Descoperirea Studiul individual Exercițiul</p>	<p>Fiecare seminar dureaza 2 ore si se desfasoara o data la 2 saptamani</p>
<p>S 2. TAD container cu elemente generice (void*): reprezentare vizibila si ascunsa. Citiri si scrieri din/in fisiere.</p> <p>L 2. Specificarea, proiectarea si implementarea unor probleme simple in C/C++. Aspecte generale ale limbajelor C si C++. Versiuni structurate si modulare ale aplicatiilor C/C++. Specificarea, proiectarea si implementarea unui TAD in C/C++. Reprezentarea unui TAD. Operatiile unui TAD. Utilizarea unui TAD.</p>	<p>Conversația Algoritmizarea Problematizarea Studiul de caz Cooperarea Studiul individual Exercițiul</p>	<p>Fiecare laborator dureaza 2 ore si se desfasoara o data la 2 saptamani</p>
<p>S 3. Clase. Clase simple. Supraincararea operatorilor. Clase cu date membre de tip obiect.</p> <p>L 3. Specificarea, proiectarea si implementarea unui TAD container in C/C++. Utilizarea elementelor generice pentru popularea containerului. Reprezentarea TAD vector cu elemente generice. Operatiile TAD vector cu elemente generice. Utilizarea TAD vector cu elemente generice. Specificarea, proiectarea si implementarea unui TAD container dinamic in C/C++. Utilizarea elementelor generice si a iteratorilor. Reprezentarea TAD container dinamic cu elemente generice. Operatiile TAD container cu elemente generice. Utilizarea TAD container cu elemente generice. Utilizarea iteratorilor.</p>	<p>Conversația Algoritmizarea Problematizarea Descoperirea Simularea Studiul individual Exercițiul</p>	<p>se desfasoara o data la 2 saptamani</p>
<p>S 4. Clase de tip lista dinamica si iteratori. Mostenire.</p> <p>L 4. Clase si obiecte. Specificarea, proiectarea si implementarea unei clase in C++. Constructori, destructor, accesori, mutatori. Supraincararea operatorilor. Obiecte statice si dinamice. Clase cu date membre de tip obiect. Specificarea, proiectarea si implementarea acestor clase. Obiecte dinamice.</p>	<p>Conversația Algoritmizarea Problematizarea Studiul de caz Brainstorming-ul Studiul individual Exercițiul</p>	
<p>S 5. Clase abstracte si interfete. Polimorfism.</p> <p>L 5. Clase si obiecte. Containere dinamice cu obiecte (liste dinamice, dictionare, tabele de dispersie). Iteratori pentru aceste containere. Specificarea, proiectarea si implementarea unei clase in C++. Constructori, destructor, accesori, mutatori. Clase cu date membre de tip obiect. Obiecte dinamice. Supraincaracarea operatorilor. Mostenire.</p>	<p>Conversația Algoritmizarea Problematizarea Descoperirea Studiul de caz Studiul individual Exercițiul</p>	
<p>S 6. Sabloane si exceptii.</p> <p>L 6. Clase si obiecte. Mostenire. Polimorfism. Specificarea, proiectarea si implementarea claselor derivate. Clase abstracte si interfete. Sabloane. Sabloane si exceptii. Specificarea, proiectarea si implementarea claselor in C++. Implementarea unei probleme respectand o diagrama UML. Sabloane de proiectare.</p>	<p>Conversația Algoritmizarea Studiul de caz Simularea Studiul individual Exercițiul</p>	
<p>S 7. Probleme complexe implementate pe baza unei diagrame UML. Sabloane de proiectare.</p> <p>L 7. -</p>	<p>Conversația Algoritmizarea Problematizarea Studiul de caz</p>	

	Brainstorming-ul Studiul individual Exercițiul	
Bibliografie <ol style="list-style-type: none"> 1. A.V. Aho, J.E. Hopcroft, J.D. Ullman, Data Structures and Algorithms, Addison-Wesley Publ., Massachusetts, 1983. 2. R. Andonie, I. Garbacea, Algoritmi fundamentali. O perspectiva C++, Editura Libris, 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. Dan Roman, Ingineria programarii obiectuale, Editura Albastra, Cluj_Napoca, 1996. 9. B. Stroustup, The C++ Programming Language, Addison Wesley, 1998. 10. Bruce Eckel, Thinking in C++, www.bruceeckel.com 11. 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

- Cursul respecta recomandările curriculare IEEE si ACM pentru studiile in informatica
- Cursul exista in programa de studiu a numeroase facultatilor de profil din intreaga lume
- Companiile de software considera continutul cursului ca fiind util in dezvoltarea abilitatilor de modelare si programare ale studentilor

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"> • Cunoasterea conceptelor de baza ale domeniului • Aplicarea principiilor de programare orientata obiect din continutul cursului pentru rezolvarea problemelor complexe si dificile 	Examen scris	50%
	<ul style="list-style-type: none"> • Aplicarea principiilor de programare orientata obiect din continutul cursului pentru rezolvarea problemelor complexe si dificile 	Proba practica	20%
10.5 Seminar/laborator	<ul style="list-style-type: none"> • Specificarea, proiectarea, implementarea si testarea aplicatiilor • Rezolvarea efectiva a problemelor cu ajutorul 	Observarea sistematică a studentului în timpul rezolvării temelor si o lucrare scrisa	10%

	metodelor anterior implementate		
	<ul style="list-style-type: none"> • Specificarea, proiectarea, implementarea si testarea aplicatiilor • Rezolvarea efectiva a problemelor cu ajutorul metodelor anterior implementate 	Observarea sistematică a studentului în timpul rezolvării temelor de laborator si realizarii proiectului	20%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • Fiecare student trebuie sa demonstreze ca a atins un nivel acceptabil de cunoastere si intelegere a domeniului, ca este capabil sa exprime cunostintele intr-o forma coerenta, ca are capacitatea de a stabili anumite conexiuni si de a utiliza cunostintele in rezolvarea unor probleme. • Pentru a promova examenul studentul trebuie să: <ul style="list-style-type: none"> ○ Realizeze cel putin 70% din temele de laborator. 			

Data completării

15 mai 2013

Semnătura titularului de curs

Conf. Dr. Diosan Laura

Semnătura titularului de seminar

Conf. Dr. Diosan Laura

Data avizării în departament

.....

Semnătura directorului de departament

Prof. Dr. Parv Bazil