

FIȘA DISCIPLINEI

PROGRAMARE ORIENTATĂ OBIECT

1. Date despre program

1.1. Instituția de învățământ superior	UNIVERSITATEA „ATHENAEUM” DIN BUCUREȘTI
1.2. Facultatea	ȘTIINȚE ECONOMICE
1.3. Departamentul	ȘTIINȚE ECONOMICE
1.4. Domeniul de licență	Cibernetică, statistică și informatică economică
1.5. Ciclul de studii	LICENȚĂ
1.6. Programul de studii/ Calificarea	Informatica Economica
1.7. Limba de studiu	ROMÂNĂ

2. Date despre disciplină

2.1 Denumirea disciplinei	PROGRAMARE ORIENTATĂ OBIECT						
2.2 Titularul/titularii activităților de curs						Email (adresa instituțională):	
2.3 Titularul/titularii activităților de seminar						Email (adresa instituțională):	
2.4 Anul de studiu	II	2.5 Semestrul	IV	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	O
2.8. Numărul de credite ECTS	5						

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

3.1 Număr de ore pe săptămână	4	din care:	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	44	din care:	22	3.6 seminar/laborator	22
3.7. Total ore de studiu pe semestru (număr ECTS * 25 ore)					125
3.8. Total ore studiu individual (3.7. – 3.4.)					81
<i>Distribuția fondului de timp de studiu individual</i>					<i>ore</i>
Studiul după manual, suport de curs, bibliografie și notițe					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					26
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					
Examinări					2
Alte activități.....					

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	- Tehnologia informației - Bazele informaticii - Programare I
4.2. de competențe	

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Săli dotate cu aparatură multimedia și conexiune la Internet/Acces Platforma online sau alte mijloace de comunicare online*
5.2. de desfășurare a	Laborator dotat cu calculatoare legate în rețea și tablă

seminarului/laboratorului	interactivă/Acces Platforma online sau alte mijloace de comunicare online*
---------------------------	--

***Pe perioada desfasurarii activitatilor didactice in sistem online**

6. Competențele specifice acumulate

Competențe profesionale	- C4.1 Definirea cerintelor de parcurgere a etapelor ciclului de dezvoltare pentru a obtine componente software performante, folosind tehnologii moderne - C4.2 Explicarea structurilor de date, a instrucțiunilor și claselor de probleme pentru a construi componente integrabile in sisteme software complexe - C4.3 Rezolvarea de probleme bine definite din economie prin aplicarea de proceduri integrabile in sisteme software complexe - C4.4 Actualizarea limbajelor, tehnicilor si metodelor de programare, astfel incat componentele software construite sa reflecte stadiul dezvoltarii IT&C - C4.5 Dezvoltarea activitatilor specifice ciclului de realizare software, urmarind aspectele cantitative, calitative si de eficienta economica
Competențe transversale	

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

7.1 Obiectivul general al disciplinei	Formarea de competențe specifice profesiilor de programator și de analist-programator Formarea de competențe specifice programării orientate obiect
7.2 Obiectivele specifice	Cunoașterea unor concepte, teorii, modele specifice informaticii; Dezvoltarea, instalarea și întreținerea sistemelor de programe Configurarea și întreținerea rețelelor de calculatoare - Utilizarea sistemelor open source - Configurarea sistemelor de calcul pentru funcționarea optimală a sistemelor de programe - Medii integrate de dezvoltare a programelor ce includ exploratoare de cod sursă, de sisteme de control a versiunilor - Elaborarea de programe extensibile și care înglobează descrieri abstracte - Tehnologiile orientate pe obiecte de reutilizare a sistemelor software - Conceptele de bază referitoare la arhitectura unei rețele de calculatoare, protocoale de comunicare, servicii etc. Promovarea abilităților studentului în operarea pe calculator; Promovarea lucrului în echipă pentru elaborarea temelor de laborator.

8. Conținuturi

8. 1 Curs	Metode de predare	Recomandări pentru studenți
C1. Elemente ale programarii procedurale:	Prelegere	Tabla interactivă,

1.1. Functii. 1.2. Transferul parametrilor. 1.3. Pointeri la date si functii. 1.4. Clase de memorie. – 2 ore	participativă; Dezbateri; Problematizare; Exercițiu; Conversație	Familiarizarea cu site-urile recomandate /Acces Platforma online sau alte mijloace de comunicare online*
C2. Concepte programare OO 2.1. Conceptele de clasa, obiect, constructor, destructor, metode de acces, pointerul this. 2.2. Declararea si implementarea metodelor in clasa si in afara clasei; 2.3. C3. Concepte de constructor 3.1. 3 Asimilarea conceptelor de constructor de copiere, supraincarcare operator =, supraincarcare operatorilor unari si binari (+, -, ++, etc). 3.2. Obiecte cu extensii in memoria dinamica si domenii de nume (namespace).- 3 ore	Idem	Idem
C4. Conversii intre diferite tipuri de obiecte. 4.1. Operatorul cast, operatorul= si constructor de copiere), vector de obiecte, modifierul const. 4.2. Tipologia membrilor statici (static), obiecte constante, pointeri constanti la obiecte si pointeri la obiecte constante. 2 ore	Idem	Idem
C.5. Mecanismul try-catch in C++. 5.1. Descriere mecanism try-catch in C++. 5.2. Exemplificare mecanism try-catch in C++. C.6. Supraincarcarea operatorilor 6.1. Descriere supraincarcarea operatorilor, 6.2. Exemplificare supraincarcarea operatorilor– 3ore	Idem	Idem
C.7. Clase derivate, mostenire. Polimorfism. 7.1. Descriere clase derivate, mostenire. Polimorfism. 7.2. Exemplificare clase derivate, mostenire. Polimorfism. – 2 ore		
C.8. Functii virtuale, supradefinire, mostenire multipla. 8.1. Descriere functii virtuale, supradefinire, mostenire multipla. 8.2. Exemplificare functii virtuale, supradefinire, mostenire multipla. C.9. Mecanisme de tip RTTI, mostenire multipla si dynamic cast. 9.1. Descriere mecanisme de tip RTTI, mostenire multipla si dynamic cast. 9.2. Exemplificare mecanisme de tip RTTI, mostenire multipla si dynamic cast. – 2 ore	Idem	Idem
C.10. Functii si clase template. 10.1. Descriere functii si clase template. 10.2 Modalitati de incapsulare a functiilor si claselor sablon in biblioteci statice/dinamice. – 2 ore	Idem	Idem
C.11. Operatii I/O orientate pe stream-uri. 11.1. Descriere operatii I/O orientate pe stream-uri mplate. 11.2. Implementarea obiectuala a structurilor dinamice	Idem	Idem

de date (liste, arbori, etc.). 11.3. Conceptele de serializare/deserializare obiecte – 2 ore		
C.12. Standard Template Library - STL. 12.1. Descriere Standard Template Library - STL – containere, iteratori si algoritmi. 12.2. Clasa string, map, list, vector, etc.–2 ore	Idem	Idem
C.13. Lucru cu biblioteci dinamice in C++. 13.1. Descriere lucru cu biblioteci dinamice in C++. 13.2. Exemplificare lucru cu biblioteci dinamice in C++ – 2 ore	Idem	Idem
Bibliografie Suport de curs in format electronic 1. Ion Smeureanu, Marian Dardala , Programarea orientata obiect in limbajul C++, CISON, Bucuresti, 2002, România 2. Ion Smeureanu, Programarea in limbajul C/C++, CISON, Bucuresti, 2001, România 3. Bjarne Stroustrup , The C++ Programming Language, 3rd Edition, Addison-Wesley, http://www.research.att.com/~bs/3rd.html , Statele Unite 4. Herbert Schildt, C++ manual complet, Teora, Bucuresti, România 5. M. Oprea, Programare orientată pe obiecte – Exemple în limbajul C ++, Editura Matrix Rom, București. 6. H. M. Deitel, P.J. Deitel (2001), C++ How to program, Prentice Hall. 7. S. Reiss (1999), A practical introduction to software design in C++, John Wiley & Sons. 8. T. Budd (1991), An introduction to object-oriented programming, Addison Wesley. 9. C. Spircu, I. Lopătan (1995), Analiza, proiectarea și programarea orientate spre obiecte, Editura Teora, București. 10. B. Stroustrup (1986), The C++ programming language, Addison Wesley.		
8. 2 Seminar/laborator	Metode de lucru	Recomandări pentru studenți
1. Dezvoltarea si rularea programelor sub Microsoft Visual Studio .NET in C++. Exemplu de structura cu pointer la functie (pentru comportament - subliniata legatura intre structura si clasa). (2 ore)	Conversație, Exerciții, Problematizare, Studiu de caz	Utilizarea aplicațiilor IT recomandate
2. Exemplu practic pentru clasa Student reliefând conceptele de clasa, obiect, constructor, destructor, metode de acces, pointer this. Declararea metodelor in clasa si in afara clasei. (2 ore)	Idem	Idem
3. Exemplu practic de clasa pentru asimilarea conceptelor de constructor de copiere, supraincarcare operator =, supraincarcare operatori unari si binari (+, -, ++, etc), obiecte cu extensii in memoria dinamica si domenii de nume (namespace). (2 ore)	Idem	Idem
4. Conversii intre diferite tipuri de obiecte (operatorul cast, operator = si constructor de copiere), vector de obiecte, modificadorul const, tipologia membrilor static (static), obiecte constante, pointeri constanti la obiecte si pointeri la obiecte constante. (2 ore)	Idem	Idem
5. Analiza algoritmului de planificare a proceselor, round-robin. Studiul influenței marimii cuantei de timp si a duratei de comutare a contextului. Prezentarea si comentarea rezultatelor. (2 ore)	Idem	Idem
6. Mecanismul try-catch. (2 ore)	Idem	Idem
7. Clase derivate, mostenire si polimorfism utilizand pointerul la tabela pointerilor catre functii virtuale.	Idem	Idem

Exemplu pe clasele Muncitor/Angajat/Profesor – polimorfism prin supraincarcare si polimorfism prin derivare/pointer la obiect/si functii virtuale. (2 ore)		
8. Exemple integrative pentru conceptele clasa, obiecte, supraincarcare operatori, transferul obiectelor in/din functii, metode friend, pointer this, etc. prin construirea claselor BigInt, Data, Vector si Matrice. Mostenire simpla, functii virtuale si polimorfism. (2 ore)	Idem	Idem
9. Generalizarea claselor prin obtinerea de sabloane (template) (2 ore)	Idem	Idem
10. Implementarea conceptului de serializare (2 ore)	Idem	Idem
11. Standard Template Library - STL – containere, ieratori si algoritmi. Clasa string, map, list, vector, etc. Exemple practice. (2 ore)	Idem	Idem
Bibliografie 1. Ion Smeureanu, Marian Dardala , Programarea orientata obiect in limbajul C++, CISON, Bucuresti, 2002, România 2. Ion Smeureanu, Programarea in limbajul C/C++, CISON, Bucuresti, 2001, România 3. Bjarne Stroustrup , The C++ Programming Language, 3rd Edition, Addison-Wesley, http://www.research.att.com/~bs/3rd.html , Statele Unite 4. Herbert Schildt, C++ manual complet, Teora, Bucuresti, România 5. M. Oprea, Programare orientată pe obiecte – Exemple în limbajul C ++, Editura Matrix Rom, București. 6. H. M. Deitel, P.J. Deitel (2001), C++ How to program, Prentice Hall. 7. S. Reiss (1999), A practical introduction to software design in C++, John Wiley & Sons. 8. T. Budd (1991), An introduction to object-oriented programming, Addison Wesley. 9. C. Spircu, I. Lopătan (1995), Analiza, proiectarea și programarea orientate spre obiecte, Editura Teora, București. 10. B. Stroustrup (1986), The C++ programming language, Addison Wesley. Suport de curs		

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

Discutarea conținutului disciplinei cu reprezentanți ai mediului academic și ai asociațiilor profesionale în domeniu.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- corectitudinea si completitudinea cunostințelor; - coerența logică; - gradul de asimilare a limbajului de specialitate;		20%
	- criterii ce vizeaza aspectele atitudinale: constiinciozitatea, interesul pentru studiu individual.		
10.5 Seminar/laborator	- participarea la grupuri de lucru - scriere specificații programe	Testare pe parcurs	20%

10.6. Evaluarea finală		Examen scris și lucrare practică de laborator//Evaluare scrisa prin aplicatii on-line	60%
10.7. Modalitatea de notare (calificativ sau notă): Notă			
10.8. Standard minim de performanță: - să cunoasca structurile fundamentale de programare orientată obiect - să dezvolte o aplicație utilizând un limbaj de programare orientat obiect			

Data completării

Semnătura titularului/titularilor
de curs

Semnătura titularului/titularilor
de seminar

27.09.2021

Data avizării în departament
28.09.2021

Semnătura directorului de departament