



FIȘA DISCIPLINEI
ANUL UNIVERSITAR 2020 – 2021

1. Date despre program

1.1. Instituția de învățământ superior	UNIVERSITATEA DIN CRAIOVA
1.2. Facultatea	FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ
1.3. Departamentul	AUTOMATICĂ ȘI ELECTRONICĂ (D28)
1.4. Domeniul de studii	INGINERIA SISTEMELOR
1.5. Ciclul de studii ¹⁾	LICENȚĂ
1.6. Specializarea/ Programul de studii	AUTOMATICĂ ȘI INFORMATICĂ APLICATĂ (cod L20601022010)
1.7. Forma de învățământ	CU FRECVENȚĂ

2. Date despre disciplină

2.1. Denumirea disciplinei	Programare orientată pe obiecte							
2.2. Titularul activităților de curs	Ș.l. dr. ing. Lavinia Aurelian Bădulescu							
2.3. Titularul activităților de seminar/ laborator/ proiect	Asist. dr. ing. Bogdan Popa							
2.4. Anul de studiu	2	2.5. Semestrul	3	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Continent ²	DS
							Obligativitate ³	DI

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. laborator	2
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6. laborator	28
Distribuția fondului de timp					ore
3.4.1. Studiul după manual, suport de curs, bibliografie și notițe					24
3.4.2. Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					21
3.4.3. Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					14
3.4.4. Tutoriala					-
3.4.5. Examinări					3
3.4.6. Alte activități					7
3.7. Total ore studiu individual	69				
3.8. Total ore pe semestru	125				
3.9. Numărul de credite ⁴	5				

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Studentul trebuie să posede cunoștințe de specialitate dobândite la următoarele discipline: Programarea calculatoarelor și limbaje de programare, Algebră liniară, geometrie analitică și diferențială, Analiză matematică, Informatică aplicată I, Metode numerice, Ingineria sistemelor de programe, Matematici speciale.
4.2. de competențe	Studentul trebuie să aibă cunoștințe/competențe referitoare la programare în limbajul C.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Predarea cursului se face folosind videoprojectorul. Pentru unele exemple, explicații și răspunsuri la întrebări adresate de studenți se folosește tabla. În cazul predării online se folosesc facilitățile aplicației Google Classroom. Se asigură suport de curs în format electronic și acces la documentații actualizate. Procesul de predare are următoarea structură: <ul style="list-style-type: none">▪ 80% prezentare teoretică, pe baza suportului de curs (slide-uri)▪ 20% activitate interactivă (discuții cu studenții)
5.2. de desfășurare a seminarului/ laboratorului/ proiectului	Laboratorul utilizează o rețea de calculatoare. În cazul predării online se folosesc facilitățile aplicației Google Classroom. Sunt implementate și testate o serie de programe având drept scop însușirea și fixarea informațiilor prezentate la curs.



ROMÂNIA
MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
UNIVERSITATEA DIN CRAIOVA

FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ
Blvd. Decebal nr.107, Craiova, RO-200440, Tel./Fax +(4)-0251-438.198, <http://ace.ucv.ro>



6. Competențe specifice acumulate

Competențe profesionale	<p>Prin cunoștințele predate la curs, prin exemplele prezentate și prin aplicațiile practice efectuate în cadrul laboratorului, cursul „Programare orientată pe obiecte” contribuie la formarea competențelor profesionale:</p> <p>C2: Operarea cu concepte fundamentale din știința calculatoarelor, tehnologia informației și comunicațiilor.</p> <p>C2.1 Descrierea funcționării și a structurii sistemelor de calcul, rețelelor de comunicații și aplicațiilor acestora în ingineria sistemelor folosind cunoștințe referitoare la limbaje, medii și tehnologii de programare, ingineria programării și instrumente specifice (algoritmi, scheme, modele, protocoale etc.).</p> <p>C2.2 Utilizarea argumentată a conceptelor din informatică și tehnologia calculatoarelor în rezolvarea de probleme bine definite din ingineria sistemelor și în aplicații ce impun utilizarea de hardware și software în sisteme industriale sau în sisteme informatice.</p>
Competențe transversale	

7. Obiectivele disciplinei

7.1. Obiectivul general al disciplinei	Contribuie la formarea viitorilor ingineri automatiști, specialiști în conducerea proceselor și informatică aplicată, asigurându-le cunoștințe referitoare la limbaje, medii și tehnologii de programare, ingineria programării și instrumente specifice (algoritmi, scheme, modele, protocoale etc.).
7.2. Obiectivele specifice	Cursul urmărește introducerea conceptele de bază privind problematica programării orientate pe obiecte, precum și caracteristicile și conceptele introduse de limbajele C++ și Java. Sunt prezentate apoi detaliile sintactice ale limbajului C++ și apoi Java. Laboratorul are rolul de a fixa cunoștințele teoretice și de a crea deprinderi de programare prin aplicații practice, exerciții și probleme.

8. Conținuturi

8.1 CURS (unități de conținut)	Nr. ore	Metode de predare
<p>1 C++ și programarea orientată pe obiecte</p> <p>1.1. Programarea procedurală</p> <p>1.2. Încapsularea datelor (modularizarea)</p> <p>1.3. Abstractizarea datelor</p> <p>1.4. Programarea orientată pe obiecte – caracteristici</p> <p>1.5. Extensii ale limbajului C în limbajul C++</p> <p>1.6. Elemente preliminare privind programarea orientată pe obiecte</p>	2	<p>Predarea cursului se face folosind videoproiectorul (când este cazul și tabla).</p> <p>În cazul predării online se folosesc facilitățile aplicației Google Classroom:</p> <ul style="list-style-type: none"> - 80% prezentare teoretică, pe baza suportului de curs (slide-uri). - 20% activitate interactivă (discuții cu studentii) <p>Materialele necesare vor fi puse la dispoziția studenților în format electronic și în formă tipărită.</p>
<p>2 Definirea și utilizarea claselor în limbajul C++</p> <p>2.1. Declararea claselor</p> <p>2.2. Accesul la datele și funcțiile membre ale unei clasei</p> <p>2.3. Implementarea claselor. Operatorul de rezoluție</p> <p>2.4. Utilizarea claselor</p> <p>2.5. Crearea mai multor obiecte</p> <p>2.6. Prevenirea redeclarării claselor</p> <p>2.7. Cuvântul cheie <code>this</code></p>	2	
<p>3 Utilizarea pointerilor și referințelor. Elemente preliminare despre funcții</p> <p>3.1. Pointeri și referințe. Pointeri la pointeri</p> <p>3.2. Elemente referitoare la aritmetica pointerilor</p> <p>3.3. Pointeri la variabile de tip <code>const</code>. Pointeri de tip <code>const</code></p> <p>3.4. Pointeri de tipul <code>void</code></p> <p>3.5. Masive de pointeri. Liste</p> <p>3.6. Diferențe dintre masiv și pointer</p> <p>3.7. Definirea unui pointer la o funcție</p> <p>3.8. Elemente preliminare despre funcții</p>	2	



4 Funcții de tip constructor și destructor 4.1. Constructori 4.2. Tipuri de constructori 4.3. Destructori 4.4. Restricții ale funcțiilor de tip constructor și destructor	2	
5 Compunerea obiectelor 5.1. Definierea și utilizarea obiectelor compuse 5.2. Crearea și distrugerea obiectelor simple 5.3. Crearea și distrugerea obiectelor compuse. Liste de inițializare a membrilor	2	
6 Mecanismul moștenirii. Construirea ierarhiilor de clase 6.1. Derivarea claselor. Specificatorii de acces 6.2. Utilizarea membrilor de tip protected 6.3. Constructori și destructori în ierarhii de clase 6.4. Mecanismul moștenirii multiple 6.5. Probleme ale moștenirii multiple	2	
7 Funcții și clase prietene. Clase imbricate 7.1. Funcții și clase prietene 7.2. Clase definite în interiorul altor clase (clase imbricate)	2	
8 Supraîncărcarea operatorilor 8.1 Generalități. Definierea și apelul operatorilor 8.2 Operatori unari 8.3 Operatorul de atribuire 8.4 Operatori binari 8.5 Conversia tipurilor	2	
9 Polimorfismul și funcții virtuale 9.1. Pointeri care punctează la clasele derivate 9.2. Declararea funcțiilor virtuale 9.3. Funcții virtuale de tip pure 9.4. Polimorfismul	2	
10 Clase și funcții parametrizate. Mecanismul template 10.1. Clase template 10.2. Funcții template	2	
11 Sistemul "stream" de I/E din C++ 11.1. Principiile de bază ale sistemului de I/E din C++ 11.2. Operații de I/E cu format 11.3. Utilizarea funcțiilor de I/E de tip manipulator 11.4. Redefinirea operatorilor << și >> 11.5. Crearea unor manipulatori definiți de utilizator	2	
12 Introducere în Java 12.1. Ce este Java ? 12.2. Structura lexicală a limbajului Java 12.3. Tipuri de date și variabile 12.4. Controlul execuției 12.5. Vectori 12.6. Șiruri de caractere 12.7. Folosirea argumentelor de la linia de comandă	2	
13 Obiecte și clase 13.1. Ciclul de viață al unui obiect 13.2. Crearea claselor 13.3. Implementarea metodelor 13.4. Modificatori de acces 13.5. Membri de instanță și membri de clasă 13.6. Clase imbricate 13.7. Clase și metode abstracte 13.8. Clasa Object 13.9. Conversii automate între tipuri 13.10. Tipul de date enumerare	2	



ROMÂNIA
MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
UNIVERSITATEA DIN CRAIOVA

FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ
Blvd. Decebal nr.107, Craiova, RO-200440, Tel./Fax +(4)-0251-438.198, http://ace.ucv.ro



14. Excepții 14.1. Ce sunt excepțiile ? 14.2. "Prinderea" și tratarea excepțiilor 14.3. "Aruncarea" excepțiilor	1	
15 Interfețe 15.1. Folosirea interfețelor 15.2. Interfețe și clase abstracte 15.3. Moștenire multiplă prin interfețe 15.4. Utilitatea interfețelor 15.5. Compararea obiectelor	1	
Bibliografie 1. Holzner, S., <i>Borland C++ Programming</i> , Brady Books, New York, 1992. 2. Brezovan, M., <i>Programare orientata pe obiecte in limbajul C++</i> , Editura SITECH, Craiova, 2008. 3. Ionete C., Petre E., Sendrescu D., <i>Programarea în C</i> , Editura SITECH, Craiova, 2003. 4. Ionita, A. D., <i>Modelarea UML in ingineria sistemelor de programe</i> , Ed. ALL, 2002. 5. Ionita, A. D., Saru D., <i>Sisteme de programe orientate pe obiecte</i> , 328 pag. Ed. ALL, 2000. 6. Jamsa, K., Klander, L., <i>Totul despre C si C++</i> , Ed. Teora, 2000. 7. Oprea, M., <i>Programare orientata pe obiecte. Exemple in limbajul C++</i> , Ed. Matrixrom, 2004. 8. Petre, E., <i>Programare orientata pe obiecte. Notite de curs</i> , 2018. 9. Prata, S., <i>Manual de programare in C++</i> , Ed. Teora, 2001. 10. Popa, I., <i>Inginerie software pentru conducerea proceselor industriale</i> , Ed. ALL, 2001. 11. Schild, H., <i>Manual complet C++</i> , Ed. Teora, 2003. 12. Somea, D., Turturea, D., <i>Introducere in C++</i> , <i>Programarea orientata pe obiecte</i> , Ed. Tehnica, Bucuresti, 1993. 13. Frăsineanu, C., Curs practic de Java, https://profs.info.uaic.ro/~acf/java/Cristian_Frasinaru-Curs_practic_de_Java.pdf , 2011. 14. Șerbănați, D. L., Bogdan C., <i>Programare orientată spre obiecte cu exemplificări în limbajul Java</i> , București : Politehnica Press, 2010.		
8.2. ACTIVITAȚI APLICATIVE (subiecte/ teme)	Nr. ore	Metode de predare
Sistemul "stream" de I/E din C++. Exemple. Aplicații	2	Efectuarea lucrărilor de laborator se face pe o rețea de calculatoare pe care se pot dezvolta aplicații C++/Java. Mediile de dezvoltare sunt Borland C++ 5.02, CodeBlocks, Visual Studio. Eclipse. În cazul predării online se folosesc facilitățile aplicației Google Classroom. Sunt puse la dispoziția studenților platforme de laborator care conțin un breviar teoretic și modul de desfășurare al lucrării.
Funcții și structuri în limbajul C++. Exemple. Aplicații	2	
Clase și obiecte. Exemple. Aplicații	2	
Utilizarea pointerilor și referințelor. Exemple. Aplicații	2	
Funcții de tip constructor și destructor. Exemple. Aplicații	2	
Mecanismul moștenirii. Exemple. Aplicații	2	
Proprietăți ale mecanismului de moștenire. Exemple. Aplicații	2	
Clase definite în interiorul altor clase (clase imbricate). Exemple. Aplicații	2	
Supraîncărcarea operatorilor. Exemple. Aplicații	2	
Funcții virtuale. Moduri de utilizare. Exemple. Aplicații	2	
Sistemul "stream" de I/E din C++. Formatarea datelor. Exemple. Aplicații	2	
Introducere în limbajul Java. Structura lexicală a limbajului Java. Controlul execuției. Vectori. Șiruri de caractere. Aplicații introductive.	2	
Crearea claselor în limbajul Java. Implementarea metodelor. Metode și clase abstracte. Clasa Object. Excepții.	2	
Interfețe. Moștenire multiplă prin interfețe. Compararea obiectelor. Exemple.	2	<ul style="list-style-type: none"> ▪ 80% desfășurarea lucrării ▪ 20% interpretarea rezultatelor și discuții cu studenții
Bibliografie 1. Holzner, S., <i>Borland C++ Programming</i> , Brady Books, New York, 1992. 2. Brezovan, M., <i>Programare orientata pe obiecte in limbajul C++</i> , Editura SITECH, Craiova, 2008. 3. Ionete, C., Petre, E., Sendrescu D., <i>Programarea în C</i> , Editura SITECH, Craiova, 2003. 4. Ionita, A. D., <i>Modelarea UML in ingineria sistemelor de programe</i> , Ed. ALL, 2002. 5. Ionita, A. D., Saru, D., <i>Sisteme de programe orientate pe obiecte</i> , 328 pag. Ed. ALL, 2000. 6. Jamsa, K., Klander, L., <i>Totul despre C si C++</i> , Ed. Teora, 2000. 7. Oprea, M., <i>Programare orientata pe obiecte. Exemple in limbajul C++</i> , Ed. Matrixrom, 2004. 8. Petre, E., <i>Programare orientata pe obiecte. Notite de curs</i> , 2014. 9. Prata, S., <i>Manual de programare in C++</i> , Ed. Teora, 2001. 10. Popa, I., <i>Inginerie software pentru conducerea proceselor industriale</i> , Ed. ALL, 2001.		



ROMÂNIA
MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
UNIVERSITATEA DIN CRAIOVA

FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ
Blvd. Decebal nr.107, Craiova, RO-200440, Tel./Fax +(4)-0251-438.198, <http://ace.ucv.ro>



11. Schild, H., *Manual complet C++*, Ed. Teora, 2003.
12. Somnea, D., Turturea, D., *Introducere in C++, Programarea orientata pe obiecte*, Ed. Tehnica, Bucuresti, 1993.
13. Frăsineanu, C., Curs practic de Java, https://profs.info.uaic.ro/~acf/java/Cristian_Frasinaru-Curs_practic_de_Java.pdf, 2011.
14. Șerbănași, D. L., Bogdan C., Programare orientată spre obiecte cu exemplificări în limbajul Java, București : Politehnica Press, 2010.
15. Joshua Bloch, *Effective Java*, The Java Series, Addison-Wesley Professional, First Edition June 01, 2001
16. Bruce E. Wampler, *The Essence of Object-Oriented Programming with Java and UML*, ISBN-13: 978-0201734102
17. Reges, S., Stepp, M., *Building Java Programs: A Back to Basics Approach*, Pearson; 5 edition (March 28, 2019).

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

Conținutul cursului a fost discutat cu reprezentanții:

- HELLA Craiova
- W-SYSTEMS DEVTEAM Craiova
- Caphyon S.R.L
- QuEST Global Craiova

10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	- Înțelegerea fundamentelor teoretice corespunzătoare programării orientate pe obiecte. - Capacitatea de a realiza realiza programe obiectuale. - Capacitatea de analiză și sinteză într-o situație concretă.	Examen scris final	60%
10.5. Seminar/Laborator	- Analiza și interpretarea rezultatelor. - Soluțiile aplicațiilor se prezintă și se discută în cadrul grupei de lucru	Verificare pe parcurs și testare finală	30%
10.6. Standard minim de performanță			
<ul style="list-style-type: none"> ▪ Obținerea a minim 50 % din punctajul verificărilor pe parcurs, testărilor de laborator și examenului final. ▪ Calculul notei finale se face prin rotunjirea la notă întreagă a punctajului final. 			

¹ Ciclul de studii- se alege una din variantele- Licenta/Master/Doctorat

² Regimul disciplinei (continut) - pentru nivelul de licenta se alege una din variantele - **DF** (disciplina fundamentala), **DD** (disciplina din domeniu), **DS** (disciplina de specialitate), **DC** (disciplina complementara).

³ Regimul disciplinei (obligativitate) - se alege una din variantele – **DI** (disciplina obligatorie) **DO** (disciplina optionala) **DFac** (disciplina facultativa).

⁴ Un credit este echivalent cu 25-30 de ore de studiu (activitati didactice si studiu individual).

Data completării

25.09.2020

Titular curs

Ș.I. dr. ing. Lavinia Aurelian Bădulescu

Titular lucrari laborator/seminarii

Asist. dr. ing. Bogdan Popa

Data avizării în

departament

30.09.2020

Director de departament

Prof. Dr. ing. Cosmin Ionete