

FIȘA DISCIPLINEI
ANUL UNIVERSITAR 2023-2024

1. DATE DESPRE PROGRAM

1.1 Instituția de învățământ superior	UNIVERSITATEA DIN CRAIOVA
1.2 Facultatea	Automatică, Calculatoare și Electronică
1.3 Departamentul	AUTOMATICĂ ȘI ELECTRONICĂ (D28)
1.4 Domeniul de studii	INGINERIA SISTEMELOR
1.5 Ciclu de studii ¹	LICENȚĂ
1.6 Programul de studii (denumire/cod) ² /Calificarea	INGINERIA SISTEMELOR MULTIMEDIA / L20602022030

2. DATE DESPRE DISCIPLINĂ

2.1 Denumirea disciplinei	Programarea orientată pe obiecte								
2.2 Titularul activităților de curs	Ș.I. dr. ing. Bogdan Popa								
2.3 Titularul activităților aplicative	Asist. drd. ing. Anca Albița								
2.4 Anul de studiu	2	2.5 Semestrul	3	2.6 Tipul disciplinei (conținut) ³	E	2.7 Regimul disciplinei (obligativitate) ⁴	DI	2.8 Tipul de evaluare	E

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/proiect	2
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 seminar/laborator/proiect	28
3.7 Distribuția fondului de timp					ore
▪ Studiul după manual, suport de curs, bibliografie și notițe					24
▪ Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					27
▪ Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					25
▪ Alte activități: consultații, cercuri studențești					18
Total ore activități individuale	94				
3.8 Total ore pe semestru ⁵	150				
3.9 Numărul de credite ⁶	6				

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 ⁹	Predarea cursului se face folosind videoproiectorul. 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 ⁹	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 înșușirea și fixarea informațiilor prezentate la curs.

6. COMPETENȚELE SPECIFICE ACUMULATE ⁷

C o m p e t e n ț e p r o f e s i o n a l e	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: C2: Operarea cu concepte fundamentale din știința calculatoarelor, tehnologia informației și comunicațiilor.
C o m p e t e n ț e t r a n s v e r s a l e	

7. OBIECTIVELE DISCIPLINEI (reieșind din grila competențelor specifice acumulate)

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
1 C++ și programarea orientată pe obiecte 1.1. Programarea procedurală 1.2. Încapsularea datelor (modularizarea) 1.3. Abstractizarea datelor 1.4. Programarea orientată pe obiecte – caracteristici 1.5. Extensii ale limbajului C în limbajul C++ 1.6. Elemente preliminare privind programarea orientată pe obiecte	2	Predarea cursului se face folosind videoproiectorul (când este cazul și tabla). În cazul predării online se folosesc facilitățile aplicației Google Classroom:

2 Definirea și utilizarea claselor în limbajul C++ 2.1. Declararea claselor 2.2. Accesul la datele și funcțiile membre ale unei clasei 2.3. Implementarea claselor. Operatorul de rezoluție 2.4. Utilizarea claselor 2.5. Crearea mai multor obiecte 2.6. Prevenirea redeclarării claselor 2.7. Cuvântul cheie this	2	- 80% prezentare teoretică, pe baza suportului de curs (slide-uri). - 20% activitate interactivă (discuții cu studenții) Materialele necesare vor fi puse la
3 Utilizarea pointerilor și referințelor. Elemente preliminare despre funcții 3.1. Pointeri și referințe. Pointeri la pointeri 3.2. Elemente referitoare la aritmetica pointerilor 3.3. Pointeri la variabile de tip const. Pointeri de tip const 3.4. Pointeri de tipul void 3.5. Masive de pointeri. Liste 3.6. Diferențe dintre masiv și pointer 3.7. Definierea unui pointer la o funcție 3.8. Elemente preliminare despre funcții	2	dispoziția studenților în format electronic și în formă tipărită.
4 Funcții de tip constructor și destructor 4.1. Constructori 4.2. Tipuri de constructori 4.3. Destructorii 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 puntează 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 << 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	
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^s	
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.	Somnea, D., Turturea, D., <i>Introducere in C++</i> , <i>Programarea orientata pe obiecte</i> , Ed. Tehnica, Bucuresti, 1993.
13.	Frăsineanu, C., <i>Curs practic de Java</i> , https://profs.info.uaic.ro/~acf/java/Cristian_Frasinaru-Curs_practic_de_Java.pdf , 2011.
14.	Șerbănași, 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. Activități: ▪ 80% desfășurarea lucrării ▪ 20% interpretarea rezultatelor și discuții cu studenții
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	

Bibliografie^s	
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.
11. Schild, H., <i>Manual complet C++</i> , Ed. Teora, 2003.
12. Somnea, D., Turturea, D., <i>Introducere in C++, 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.
15. Joshua Bloch, <i>Effective Java</i> , The Java Series, Addison-Wesley Professional, First Edition June 01, 2001
16. Bruce E. Wampler, <i>The Essence of Object-Oriented Programming with Java and UML</i> , ISBN-13: 978-0201734102
17. Reges, S., Stepp, M., <i>Building Java Programs: A Back to Basics Approach</i> , Pearson; 5 edition (March 28, 2019).

9. COROBORAREA CONȚINUTURILOR DISCIPLINEI CU AȘTEPTĂRILE REPREZENTANȚILOR COMUNITĂȚII EPISTEMICE, ASOCIAȚIILOR PROFESIONALE ȘI ANGAJATORI REPREZENTATIVI DIN DOMENIUL AFERENT PROGRAMULUI

<p>Conținutul cursului a fost discutat cu reprezentanții:</p> <ul style="list-style-type: none"> ▪HELLA Craiova ▪W-SYSTEMS DEVTEAM Craiova ▪Caphyon S.R.L ▪QuEST Global Craiova

10. EVALUARE ¹¹

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<p>C:</p> <ul style="list-style-type: none"> - Înțelegerea fundamentelor teoretice corespunzătoare programării orientate pe obiecte. - Capacitatea de a realiza realiza programe obiectuale. - Capacitatea de analiză și sinteză într-o situație concretă. 	<p>Examen scris final Examen grilă final</p>	<p>30% 30%</p>
10.5 Activități aplicative	<p>L:</p> <ul style="list-style-type: none"> - Analiza și interpretarea rezultatelor. - Soluțiile aplicațiilor se prezintă și se discută în cadrul grupei de lucru 	<p>Verificare pe parcurs și testare finală</p>	<p>40%</p>
10.6 Standard minim de performanță (volumul de	<ul style="list-style-type: none"> • Obținerea a minim 50 % din punctajul verificărilor pe parcurs, testărilor de laborator și examenului final. 		

cunoștințe minim necesar pentru promovarea disciplinei și modul în care se verifică stăpânirea lui)	<ul style="list-style-type: none"> • Calculul notei finale se face prin rotunjirea la notă întreagă a punctajului final.
-----------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------

Data completării: 30.09.2023

Titular curs
Ș.I. dr. ing. Bogdan Popa

Titular activități aplicative
Asist. drd. ing. Anca Albița

Data avizării în departament:

Director de departament
Prof. Dr. ing. Cosmin Ionete

Notă:

- 1) Ciclul de studii - se alege una din variantele: L (licență)/ M (master)/ D (doctorat).
- 2) Se înscrie codul prevăzut în HG nr. 493/17.07.2013.
- 3) Tip (conținut) - se alege una din variantele:
 - pentru nivelul de licență: DF (disciplină fundamentală)/ DD (disciplină din domeniu)/ DS (disciplină de specialitate)/ DC (disciplină complementară);
 - pentru nivelul de master: DA (disciplină de aprofundare)/ DS (disciplină de sinteză)/ DCA (disciplină de cunoaștere avansată).
- 4) Regimul disciplinei (obligativitate) - se alege una din variantele: DI (disciplină obligatorie)/ DO (disciplină opțională)/ FC (disciplină facultativă).
- 5) Se obține prin însumarea numărului de ore de la punctele 3.4 și 3.7.
- 6) **Un credit este echivalent cu 25 de ore de studiu (activități didactice și studiu individual).**
- 7) Aspectul competențelor profesionale și competențelor transversale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competențele care sunt precizate în Registrul Național al Calificărilor din Învățământul Superior RNCIS (http://www.rncis.ro/portal/page?_pageid=117,70218&_dad=portal&_schema=PORTAL) pentru domeniul de studiu de la pct. 1.4 și programul de studii de la pct. 1.6 din această fișă, la care participă disciplina.
- 8) Se recomandă ca cel puțin un titlu să aparțină colectivului disciplinei iar cel puțin 2-3 titluri să se refere la lucrări relevante pentru disciplină, de circulație națională și internațională, existente în biblioteca UCv.
- 9) În cazul situațiilor speciale, activitățile se vor desfășura conform regulamentelor și a reglementărilor specifice la nivelul Universității și ale facultății.
- 10) În cazul situațiilor speciale, metodele de predare se vor adapta conform regulamentelor și a reglementărilor specifice la nivelul Universității și ale facultății.
- 11) În cazul situațiilor speciale, metodele de evaluare se vor adapta conform regulamentelor și a reglementărilor specifice la nivelul Universității și ale facultății.