



FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea din Craiova
1.2. Facultatea	<i>Automatică, Calculatoare și Electronică</i>
1.3. Departamentul	AUTOMATICĂ ȘI ELECTRONICĂ (D28)
1.4. Domeniul de studii	INGINERIA SISTEMELOR
1.5. Ciclu de studii universitare	LICENȚĂ
1.6. Forma de organizare	IF – Învățământ cu frecvență
1.7. Programul de studii	AUTOMATICĂ ȘI INFORMATICĂ APLICATĂ

2. Date despre disciplină

2.1. Denumirea disciplinei			Programarea orientată pe obiecte				
2.2. Titularul activităților de curs			Ș.l. dr. ing. Bogdan Popa				
2.3. Titularul activităților de seminar/ laborator			Ș.l. dr. ing. Anca Albița, Ș.l. dr. ing. Bogdan Popa				
2.4. Anul de studiu	2	2.5. Semestrul	1	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	DOB

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

3.1. Numărul 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/sapt.					ore
Studiul după manual, suport de curs, bibliografie și notițe					12
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					16
Tutoriat					1
Examinări					4
Alte activități: consultații, cercuri studențești					1
3.7. Total ore studiu individual					44
3.8. Total ore pe semestru					100
3.9. Numărul de credite					4

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 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: • 80% prezentare teoretică, pe baza suportului de curs (slide-uri) • 20% activitate interactivă (discuții cu studenții) Se asigură suport de curs și subiecte pentru examen, în format electronic, pe Classwork (Platforma Google Classroom), la clasa disciplinei.
5.2. de desfășurare a seminarului/laboratorului	Laboratorul se desfășoară <i>Varianta cu prezență fizică</i> (folosind aparate și machete de laborator, videoproiectorul și medii de dezvoltare C++ (Code Blocks, Visual Studio) sau Java (Eclipse, Netbeans)). <i>Varianta on-line</i> (primele 3 săptămâni) a laboratorului presupune meet-uri organizate folosind platforma Google Classroom. Activitatea de laborator constă în folosirea software dedicate pentru rezolvarea aplicațiilor în limbajul C++ (Code Blocks, Visual Studio, Dev C++). Studenții primesc teme de casă, activitatea didactică este interactivă și sunt evaluați pe parcursul. Se asigură suport pentru activitatea de laborator, în format electronic, pe Classwork, la clasa disciplinei.

6. Obiectivele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei

Cunoștințe	C1: Studentul/absolventul realizează operarea cu concepte fundamentale din știința calculatoarelor, tehnologia informației și comunicațiilor. C2: Studentul/absolventul descrie, identifică, sumarizează concepte precum arhitecturile orientate pe obiecte și metode elementare privitoare la limbaje de programare, medii de programare, tehnici de programare, baze de date, inteligență artificială și inginerie software și modul lor de aplicare în probleme concrete. C3: Studentul/absolventul identifică și descrie concepte, principii și metode de bază din matematică, fizică, chimie, desen tehnic, economie și informatică.
Aptitudini (Abilități)	A1. Studentul/absolventul alege și explică concepte proprii specifice proiectării algoritmilor, programării orientate pe obiecte, programării logice și funcționale. A2. Studentul/absolventul specifică cerințe, analizează, elaborează, dezvoltă și testează programe în limbaje de programare de uz general (C, etc.) și /sau obiect-orientate (C++, Java, etc.), aplicând elementele specifice ingineriei software. A3. Studentul/absolventul proiectează, implementează, dezvoltă și /sau gestionează aplicații ce includ diverse tipuri de baze de date. A5. Studentul/absolventul operează cu concepte, principii și metode de bază din matematică, fizică, chimie, desen tehnic, economie și informatică. A6. Studentul/absolventul efectuează calcule ingineresti și economice de complexitate medie și le asociază cu reprezentări grafice letrice sau specifice proiectării asistate de calculator.

Responsabilitate și autonomie	AR1: Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei. AR2: Studentul/absolventul aplică valorile eticii și deontologiei profesiei de inginer. AR3: Studentul/absolventul practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor. AR4: Studentul/absolventul comunică eficient despre activitățile de inginerie cu o gamă largă de public. AR5: Studentul/absolventul este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate. AR6: Studentul/absolventul promovează dialogul, cooperarea, respectul față de ceilalți și interculturalitatea. AR7: Studentul/absolventul lucrează eficient ca membru în echipă sau lider al acesteia.
--------------------------------------	---

7. Conținuturi

7.1. CURS	Modalitatea de desfășurare	Metode de predare	Fond de timp alocat (ore)
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	Online (săptămâna 1)	Predarea cursului se face folosind videoproiectorul (când este cazul și tabla). - 80% prezentare teoretică, pe baza suportului de curs (slide-uri). - 20% activitate interactivă (discutii cu studentii)	2
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	Online (săptămâna 2)	Materialele necesare vor fi puse la dispoziția studenților în format electronic.	2
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	Online (săptămâna 3)		2

3.7. Definirea unui pointer la o funcție 3.8. Elemente preliminare despre funcții			
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	Online (săptămâna 4)		2
5. Compunerea obiectelor 5.1. Definirea și utilizarea obiectelor compuse 5.2. Crearea și distrugerea obiectelor simple 5.3. Crearea și distrugerea obiectelor compuse. Liste de inițializare a membrilor	Online (săptămâna 5)		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	Online (săptămâna 6)		2
7. Funcții și clase prietene. Clase imbricate 7.1. Funcții și clase prietene 7.2. Clase definite în interiorul altor clase (clase imbricate)	față în față (săptămâna 7)		2
8. Supraîncărcarea operatorilor 8.1 Generalități. Definirea și apelul operatorilor 8.2 Operatori unari 8.3 Operatorul de atribuire 8.4 Operatori binari 8.5 Conversia tipurilor	față în față (săptămâna 8)		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	față în față (săptămâna 9)		2
10. Clase și funcții parametrizate. Mecanismul template	față în față (săptămâna 10)		2

10.1. Clase template 10.2. Funcții template			
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	față în față (săptămâna 11)		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	față în față (săptămâna 12)		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	față în față (săptămâna 13)		2
14. Excepții 14.1. Ce sunt excepțiile ? 14.2. "Prinderea" și tratarea excepțiilor 14.3. "Aruncarea" excepțiilor	față în față (săptămâna 14)		1
15. Interfețe 15.1. Folosirea interfetelor 15.2. Interfețe și clase abstracte 15.3. Moștenire multiplă prin interfete 15.4. Utilitatea interfetelor 15.5. Compararea obiectelor	față în față (săptămâna 14)		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. 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., Programare orientată spre obiecte cu exemplificări în limbajul Java, București : Politehnica Press, 2010.
15. Jana, Debasish. C++ and object-oriented programming paradigm. PHI Learning Pvt. Ltd., 2014.
16. Malik, D. S. <i>C++ programming</i> . Cengage Learning, EMEA, 2023
17. Cyganek, Boguslaw. <i>Introduction to Programming with C++ for Engineers</i> . John Wiley & Sons, 2020
18. Bancila, Marius. <i>Modern C++ Programming Cookbook: Master modern C++ including the latest features of C++ 23 with 140+ practical recipes</i> . Packt Publishing, 2024.
19. Hubbard, John R. "Programming with C++." (2021).
20. Horton, Ivor, and Peter Van Weert. <i>Beginning C++ 23: from beginner to pro</i> . Apress, 2023.
21. Iglberger, Klaus. <i>C++ Software Design</i> . " O'Reilly Media, Inc.", 2022.
22. Saxena, Supriya, et al. <i>C++ Programming</i> . Chyren Publication, 2025.
23. Bhatt, Ashish, Rashmi Kuksal, and Praveen Shah. "Programming Principles and Practice Using C++." ASP Books (2025): 01-93.

7.2. Seminar/laborator	Modalitatea de desfășurare	Metode de predare	Fond de timp alocat (ore)
1. Sistemul "stream" de I/E din C++. Exemple. Aplicații	online (săptămâna 1)	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. 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	2
2. Funcții și structuri în limbajul C++. Exemple. Aplicații	online (săptămâna 2)		2
3. Clase și obiecte. Exemple. Aplicații	online (săptămâna 3)		2
4. Utilizarea pointerilor și referințelor. Exemple. Aplicații	față în față (săptămâna 4)		2
5. Funcții de tip constructor și destructor. Exemple. Aplicații	față în față (săptămâna 5)		2
6. Mecanismul moștenirii. Exemple. Aplicații	față în față (săptămâna 6)		2
7. Proprietăți ale mecanismului de moștenire. Exemple. Aplicații	față în față (săptămâna 7)		2
8. Clase definite în	față în față		2

interiorul altor clase (clase imbricate). Exemple. Aplicatii	(săptămâna 8)		
9. Supraîncărcarea operatorilor. Exemple. Aplicatii	față în față (săptămâna 9)		2
10. Funcții virtuale. Moduri de utilizare. Exemple. Aplicatii	față în față (săptămâna 10)		2
11. Sistemul "stream" de I/E din C++. Formatarea datelor. Exemple. Aplicații	față în față (săptămâna 11)		2
12. Introducere în limbajul Java. Structura lexicală a limbajului Java. Controlul execuției. Vectori. Șiruri de caractere. Aplicații introductive.	față în față (săptămâna 12)		2
13. Crearea claselor în limbajul Java. Implementarea metodelor. Metode și clase abstracte. Clasa Object. Excepții.	față în față (săptămâna 13)		2
14. Interfețe. Moștenire multiplă prin interfețe. Compararea obiectelor. Exemple.	față în față (săptămâna 14)		2

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. 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. Jana, Debasish. C++ and object-oriented programming paradigm. PHI Learning Pvt. Ltd., 2014.
16. Malik, D. S. <i>C++ programming</i> . Cengage Learning, EMEA, 2023
17. Cyganek, Boguslaw. <i>Introduction to Programming with C++ for Engineers</i> . John Wiley & Sons,

2020
18. Bancila, Marius. <i>Modern C++ Programming Cookbook: Master modern C++ including the latest features of C++ 23 with 140+ practical recipes</i> . Packt Publishing, 2024.
19. Hubbard, John R. "Programming with C++." (2021).
20. Horton, Ivor, and Peter Van Weert. <i>Beginning C++ 23: from beginner to pro</i> . Apress, 2023.
21. Iglberger, Klaus. <i>C++ Software Design</i> . " O'Reilly Media, Inc.", 2022.
22. Saxena, Supriya, et al. <i>C++ Programming</i> . Chyren Publication, 2025.
23. Bhatt, Ashish, Rashmi Kuksal, and Praveen Shah. "Programming Principles and Practice Using C++." ASP Books (2025): 01-93.

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

Conținutul cursului a fost discutat cu reprezentanții:
Forvia Craiova, SUGAR CRM Craiova, Caphyon S.R.L, QuEST Global Craiova, Cs Romania

9. Evaluare

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	- Înțelegerea fundamentelor teoretice corespunzătoare programării orientate pe obiecte.	Examen scris final Examen grilă final	30% 30%
	- Capacitatea de a realiza realiza programe obiectuale.		
	- Capacitatea de analiză și sinteză într-o situație concretă.		
9.5. Seminar/laborator	- Analiza și interpretarea rezultatelor.	Verificare pe parcurs și testare finală	40%
	- Soluțiile aplicațiilor se prezintă și se discută în cadrul grupei de lucru		
9.6. Standard minim de performanță			
Standardul minim de performanță este atins dacă studentul:			
• Obține 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.			

Data completării
30.09.2025

Titular de disciplină,
Ș.l. dr. ing. Bogdan Popa
Semnătura titularului

Data avizării în departament
30.09.2025

Director de departament,
Conf. dr. ing. Mircea Cătălin Constantinescu
Semnătura directorului de departament,

Decan,
Prof. dr. ing. Dorin Gheorghe Șendrescu
Semnătura decanului,