



FIȘA DISCIPLINEI FUNDAMENTELE PROGRAMĂRII

1. Date despre program

| | |
|---------------------------------------|---|
| 1.1 Instituția de învățământ superior | UNIVERSITATEA „OVIDIUS” DIN CONSTANȚA |
| 1.2 Facultatea | Facultatea de Matematica și Informatica |
| 1.3 Departamentul | Matematica și Informatica |
| 1.4 Domeniul de studii | Informatica |
| 1.5 Ciclul de studii | Licenta |
| 1.6 Programul de studii | Informatica |
| 1.7 Anul universitar | 2025-2026 |

2. Date despre disciplină

| | | | | | | | |
|--|---|---------------------------|---|-----------------------|---|---------------------------------|--------|
| 2.1 Denumirea disciplinei | | Fundamentele programării | | | | | |
| 2.2 Cod disciplină | | Info.1.1.03 | | | | | |
| 2.3 Titularul activităților de curs | | Conf.univ.dr. Elena BĂUTU | | | | | |
| 2.4 Titularul activităților aplicative | | Conf.univ.dr. Elena BĂUTU | | | | | |
| 2.5 Anul de studii | 1 | 2.6 Semestrul | 1 | 2.7 Tipul de evaluare | E | 2.8 Regimul disciplinei */** | DF/DOB |

* DF – disciplină fundamentală, DS – disciplină de specializare, DC – disciplină complementară

** DOB – disciplină obligatorie; DOP – disciplină opțională; DFA – Disciplină facultativă

3. Timpul total (ore pe semestru)

| | | | | | |
|---|-------------|--------------------|----|------------------|-------|
| 3.1 Număr de ore activități directe pe săptămână | 4 | din care: 3.2 curs | 2 | 3.3 aplicații*** | 2 |
| 3.4 Total ore activități directe pe semestru | 56 | din care: 3.5 curs | 28 | 3.6 aplicații | 28 |
| 3.7 Total ore de studiu individual | | | | | 69 |
| Distribuția fondului de timp | | | | | [ore] |
| Studiul cărților, manualelor, suportului de curs,, notițelor, bibliografie minimală recomandată | | | | | 10 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren | | | | | 10 |
| Pregătire seminar / laborator / proiect, teme, referate, portofolii și eseuri | | | | | 20 |
| Pregătire pentru prezentări sau verificări | | | | | 12 |
| Pregătire pentru examinarea finală | | | | | 13 |
| Alte activități: consultații | | | | | 4 |
| 3.8 Total ore pe semestru | 125 (56+69) | | | | |
| 3.9 Numărul de credite | 5 | | | | |

*** S - seminar; L - laborator; P - proiect

4. Precondiții

| | |
|--------------------------------|------------|
| 4.1 de curriculum | Nu e cazul |
| 4.2 de rezultate ale învățării | Nu e cazul |



5. Condiții necesare pentru desfășurarea optimă a activităților didactice

| | |
|---|---|
| 5.1. de desfășurare a cursului | Sala de curs dotata cu videoproiector |
| 5.2. de desfășurare a seminarului/ laboratorului / proiectului* | Sala de laborator dotata cu calculatoare si conexiune la internet |

*Se alege tipul de aplicație aferent disciplinei

6. Obiectivele disciplinei

| | |
|---------------------------------------|---|
| 6.1 Obiectivul general al disciplinei | Familiarizarea studentului cu principiile programarii procedurale Intelegerea si implementarea si unor algoritmi de baza |
| 6.2 Obiectivele specifice | Studentul va fi familiarizat cu elemente fundamentale ale programarii procedurale cum ar fi variabila, tip de data, proceduri si functii, transferul parametrilor, precum si cu elemente minimale de ingineria software-ului. Studentii isi vor imbunatati cunostintele si abilitatile de programare, inclusiv prin folosirea unui mediu de dezvoltare software integrat. |

7. Rezultatele învățării

| | |
|------------|---|
| Cunoștințe | Studentul / Absolventul - identifică, explică și argumentează concepte fundamentale de paradigme de programare, precum și ale arhitecturii calculatoarelor. - identifică și explică conceptele fundamentale ale programării procedurale, exemplificând folosind limbajul de programare C - înțelege structura unui program, tipurile de date, operatorii, expresiile și instrucțiunile pentru controlul fluxului de execuție - cunoaște mecanismele de lucru cu funcții, pointeri, structuri, fișiere și memoria dinamică - cunoaște bune practici de programare, debugging și gestionare a erorilor - explică principiile de baza de programare in C++ |
| Aptitudini | Studentul / Absolventul - elaborează, dezvoltă și demonstrează soluții software complexe utilizând algoritmi eficienți și paradigme diverse de programare - dezvoltă programe modulare și eficiente în C, aplicând concepte de bază și avansate - utilizează pointeri, alocare dinamică și funcții pentru gestionarea corectă a memoriei - testează și depanează programe, identificând erori de compilare, runtime și logice - elaborează mici proiecte software integrate |



| | |
|-------------------------------|--|
| Responsabilitate și autonomie | Studentul / Absolventul: |
| | <ul style="list-style-type: none">- coordonează echipe tehnice pentru dezvoltarea de aplicații informatice, asumând decizii responsabile legate de optimizarea și integrarea acestora;- dezvoltă programe corecte și eficiente respectând principiile de bune practici și standardele de codare- gestionează responsabil resursele de calcul (memorie, fișiere) și evită scurgerile de memorie- documentează și structurează proiectele software, facilitând colaborarea în echipă- ia decizii autonome privind proiectarea modulară și reutilizabilă a codului- se adaptează la noi concepte și tehnologii, aplicând în mod independent cunoștințele de programare |

8. Conținuturi

| 8.1 Curs | Metode de predare | Număr ore alocate |
|--|---|-------------------|
| 1. Introducere in programarea procedurala. Operatori. Instructuni. Structura unui program, identificatori, variabile, constante, declaratii, tipuri fundamentale. Clasificarea operatorilor in functie de aritate, pozitie, tipul prelucrării efectuate. Reguli de precedenta si asociativitate. Expresii. | Medode de predare-invatare interactive. Problematizarea | 2 ore |
| 2. Instructiuni. Instructiuni de control al fluxului de executie al unui program: instructiuni decizionale (if, switch), instructiuni repetitive (for, while, do-while), instructiuni de salt (break, continue, return). | Metodele active și interactive cu multiple Metode care contribuie la dezvoltarea gândirii critice. | 4 ore |
| 3. Masive de date (tablouri). Siruri de caractere. Tablouri unidimensionale si multidimensionale. Manipularea sirurilor de caractere. | Interactiunea, problematizarea, argumentarea | 2 ore |
| 4. Functii. Declararea, implementarea si apelul unei functii, durata de viata vs. domeniu de accesibilitate, prototipul unei functii, transferul parametrilor prin valoare, redefinirea functiilor, parametri impliciti, pointeri la functii. Functii recursive. Functii din biblioteci standard (functii matematice, functii pentru siruri de caractere). Domeniul, vizibilitatea si tipul identificatorilor. | Dialogul Sintetizarea/ esențializarea informațiilor Problematizarea | 4 ore |



| | | |
|--|---|------------------------------|
| 5. Pointeri Definitie,exemple. Transferul parametrilor de tip pointer. Aritmetica pointerilor. Accesul direct la masive de date. Gestiunea dinamica a memoriei. | Invatarea independentă și prin cooperare | 4 ore |
| 6. Structuri, uniuni, enumerari. Structuri, typedef, functii cu structuri, masive de structuri, campuri, uniuni si enum. | Generalizarea Conversatia | 4 ore |
| 7. Lucrul cu fisiere in C. Functii pentru intrari/iesiri standard. Operatii cu fisiere (accesarea datelor, manevrarea datelor in format text si binar, acces aleator la fisier). | | 2 ore |
| 8. Notiuni de programare C++: Tipuri de date complexe definite de utilizator. Supraincercarea operatorilor pentru tipurile de date definite de utilizator. Redefinirea si reutilizarea functiilor. Fuxuri standard de intrare/iesire. | | 4 ore |
| 9. Recapitulare | | 2 ore |
| Bibliografie [1]. Slobodan Dmitrovic, Modern C for Absolute Beginners, Apress 2021, https://github.com/Apress/modern-c-absolute-beginners [2]. D.M.Popovici, I.M.Popovici, C++. Tehnologia orientata spre obiecte. Aplicatii, Ed. Teora, Bucuresti, 2000, 486pag, ISBN:973-20-0320-0. [3]. Suportul de curs, disponibil în format pdf si suportul de laborator, ambele disponibile pe platforma Moodle a universitatii: http://moodle.univ-ovidius.ro [4]. Limbajele C si C++ pentru incepatori, Liviu Negrescu, Editura Albastra, Cluj Napoca, 20005. [5]. The C Programming Language Book, Brian Kernighan and Dennis Ritchie, Prentice Hall, 1990, https://www.cs.sfu.ca/~ashriram/Courses/CS295/assets/books/C_Book_2nd.pdf [6]. Tutorial C online, https://www.tutorialspoint.com/cprogramming/ | | |
| 8.2 Aplicații (seminar/laborator/proiect)* <i>*Se alege tipul de aplicație aferent disciplinei</i> | Metode de predare | Număr ore alocate |
| 1. Introducere în programarea procedurala. Operatori. Instructuni. Exerciții practice de scriere, compilare și rulare a programelor simple în C. Definirea variabilelor și constantelor, utilizarea tipurilor fundamentale și realizarea de calcule aritmetice de bază. Probleme practice de calcul folosind operatori aritmetici, logici și relaționali. Construirea de expresii și verificarea rezultatelor prin programe scurte. | Problematizarea Exercitiul Metodele active și interactive cu multiple | 2 ore |



| | | |
|---|--|-------|
| 2. Instrucțiuni. Exerciții cu instrucțiuni decizionale (if, switch) și repetitive (for, while, do-while). Implementarea de programe care procesează condițional datele introduse de utilizator. | Lucrul în grup | 4 ore |
| 3. Masive de date (tablouri). Siruri de caractere. Probleme aplicative cu tablouri unidimensionale și bidimensionale (căutare, sortare, operații pe matrice). Exerciții de prelucrare a șirurilor de caractere (citire, parcurgere, transformare).. | Interacțiunea, problematizarea, argumentarea | 4 ore |
| 4. Funcții. Scrierea de programe cu funcții proprii. Exerciții de transmitere a parametrilor prin valoare și prin pointer. Implementarea și testarea de funcții recursive. | Învățarea independentă și prin cooperare | 4 ore |
| 5. Pointeri Probleme aplicative cu pointeri și aritmetică de pointeri. Crearea și manipularea de tablouri dinamice. Testarea programelor pentru scurgeri de memorie. | Generalizarea | 4 ore |
| 6. Structuri, uniuni, enumerari. Exerciții cu structuri pentru organizarea datelor complexe. Construirea de programe care folosesc masive de structuri și funcții care primesc structuri ca parametri. Aplicații cu uniuni și typedef. | Conversația | 2 ore |
| 7. Lucrul cu fișiere în C. Programe care citesc și scriu fișiere text și binare. Exerciții de citire secvențială și acces aleator. | Sintetizarea informațiilor | 2 ore |
| 8. Noțiuni de programare C++: Crearea unor tipuri complexe definite de utilizator care includ variabile și funcții. Exerciții de scriere și citire a datelor cu cin și cout. Exerciții de redefinire a funcțiilor pentru adaptarea comportamentului și testarea selecției dinamice a implementării în funcție de datele introduse. Aplicarea conceptului de redefinire a operatorilor (e.g. +, ==, <<, >>) pentru tipurile proprii. | | 6 ore |
| 9. Recapitulare | | 2 ore |



Bibliografie

- [1]. Slobodan Dmitrovic, Modern C for Absolute Beginners, Apress 2021, <https://github.com/Apress/modern-c-absolute-beginners>
- [2]. D.M.Popovici, I.M.Popovici, C++. Tehnologia orientata spre obiecte. Aplicatii, Ed. Teora, Bucuresti, 2000, 486pag, ISBN:973-20-0320-0.
- [3]. Suportul de curs, disponibil în format pdf si suportul de laborator, ambele disponibile pe platforma Moodle a universitatii: <http://moodle.univ-ovidius.ro>
- [4]. Limbajele C si C++ pentru incepatori, Liviu Negrescu, Editura Albastra, Cluj Napoca, 20005.
- [5]. The C Programming Language Book, Brian Kernighan and Dennis Ritchie, Prentice Hall, 1990, https://www.cs.sfu.ca/~ashriram/Courses/CS295/assets/books/C_Book_2nd.pdf
- [6]. Tutorial C online, <https://www.tutorialspoint.com/cprogramming/>

9. Evaluare

| Tip activitate | 9.1 Criterii de evaluare | 9.2 Metode de evaluare | 9.3 Pondere din nota finală |
|---|---|---|-----------------------------|
| 9.4 Curs | Participare activa la ore | Evaluare continuă orală, pe parcursul semestrului | 5% |
| 9.5 Aplicații* <i>*Se alege tipul de aplicație aferent disciplinei</i> | Capacitatea studentului de a rezolva autonom probleme de programare într-un timp limitat | Teste pe calculator în laborator, pe parcursul semestrului | 30% |
| | Interes și capacitate de lucru pentru studiu individual și în echipă | Efectuarea temelor de laborator evaluate pe parcursul semestrului | 5% |
| | Evaluarea nivelului de cunostinte si a capacitatii de a analiza si implementa solutii corecte si eficient pentru probleme de programare | Examen final in sesiune | 50% |
| Din oficiu | | | 10% |



9.6 Standard minim de performanță / Condiții de promovare: Nota 5 din 10.

Cunoasterea structurii unui program C. Cunoasterea operațiilor de intrare/iesire. Instrucțiuni de control a fluxului de execuție. Lucrul cu tablouri uni și bi-dimensionale. Realizarea și expunerea unor programe pentru probleme propuse simple.

Nota finală la disciplină trebuie să fie mai mare sau egală cu 5 pentru a promova.

Media finală se calculează ca medie ponderată (ponderile sunt precizate la 9.4 și 9.5).

Testele de laborator se dau în timpul semestrului.

Temele de laborator se lucrează și se evaluează în timpul semestrului.

Modalitatea de calcul a notei este aceeași, în toate sesiunile de examinare din cursul anului universitar.

Data completării,
15.09.2025

Titular activităților de curs,
Conf. dr. Elena BĂUTU

Titular aplicații,
Conf. dr. Elena BĂUTU

Data avizării în Departament,
19.09.2025

Director de Departament,
Conf. dr. Elena PELICAN

Decan,
Conf. dr. Aurelian NICOLA