

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea „Dunărea de Jos” Galați
1.2 Facultatea	Economie și Administrarea Afacerilor
1.3 Departamentul	Administrarea Afacerilor
1.4 Domeniul de studii	Cibernetică, Statistică și Informatică economică
1.5 Ciclul de studii	Licență
1.6 Programul de studii	Informatică economică

2. Date despre disciplină

2.1 Denumirea disciplinei	Bazele programării calculatoarelor						
2.2 Codul disciplinei	1107.1OB12F						
2.3 Titularul activităților de curs							
2.4 Titularul activităților de seminar							
2.5 Anul de studiu	I	2.6 Semestrul	II	2.7 Tipul de evaluare	E	2.8 Regimul disciplinei	OB

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	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					69 ore
Studiul după manual, suport de curs, bibliografie și notițe					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					20
Tutoriat					5
Examinări					4
Alte activități.....					
3.7 Total ore studiu individual	69				
3.9 Total ore pe semestru	125				
3.10 Numărul de credite	5				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Nu este cazul
4.2 de competențe	• Nu este cazul



5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs, laptop, videoproiector
5.2. de desfășurare a seminarului/laboratorului	- Calculatoare, - C/C++ instalat pe fiecare calculator, - conexiune la Internet pentru fiecare calculator.

6. Competențele specifice acumulate

Competențe profesionale	- C4.1 Definirea cerințelor de parcurgere a etapelor ciclului de dezvoltare pentru a obține componente software performante, folosind tehnologii moderne. - C4.2 Explicarea structurilor de date, a instrucțiunilor și claselor de probleme pentru a construi componente integrabile în sisteme software complexe. - C4.4 Actualizarea limbajelor, tehnicilor și metodelor de programare, astfel încât componentele software construite să reflecte stadiul dezvoltării IT&C. - C4.5 Dezvoltarea activităților specifice ciclului de realizare software, urmărind aspectele cantitative, calitative și de eficiență economică.
Competențe transversale	CT3 Identificarea oportunităților de formare continuă și valorificarea eficientă a resurselor și tehnicilor de învățare pentru propria dezvoltare.

7. Rezultatele învățării

7.1 Cunoștințe	7.2 Aptitudini	7.3 Responsabilitate și autonomie
Studentul/Absolventul identifică, recunoaște, descrie și explică, conceptele fundamentale privind programarea calculatoarelor și a tehnicilor de programare de bază, exemplificate printr-un limbaj popular de programare. Studentul/Absolventul descrie și explică structuri de date, algoritmi, limbaje de programare utilizate în domeniul economic. Studentul/Absolventul descrie structurile de date fundamentale. Studentul/Absolventul analizează complexitatea algoritmilor de căutare și sortare.	Studentul/Absolventul utilizează instrumente pentru proiectarea, implementarea, testarea și depanarea unui program ce folosește construcțiile fundamentale specifice programării. Studentul/Absolventul efectuează calcule de bază pentru operații de: de intrare/ieșire simple, structuri condiționale și iterative, subprograme, modularizarea programelor, lucrul cu date stocate extern în structuri simple, algoritmi de bază pentru rezolvarea problemelor, implementarea algoritmilor studiați, folosind un limbaj de programare. Studentul/Absolventul scrie cod pentru prelucrarea datelor economice, utilizând limbaje de programare, în context economic; Studentul/Absolventul programează algoritmi eficienți de manipulare a structurilor de date. Studentul/Absolventul evaluează complexitatea algoritmică pentru a optimiza	Studentul/Absolventul proiectează, implementează, testează și depanează programe care folosesc construcțiile fundamentale specifice programării



performanța programelor.

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

8.1 Obiectivul general al disciplinei	Dobândirea cunoștințelor și abilităților necesare realizării de programe în limbajul de programare C/C++.
8.2 Obiectivele specifice	Cunoașterea unor algoritmi de bază în domeniul programării calculatoarelor, capacitatea de a elabora algoritmul de rezolvare a unor probleme, capacitatea de a implementa algoritmi și de a testa funcționalitatea aplicațiilor realizate.

9. Conținuturi

9. 1 Curs	Metode de predare	Observații
Date, operatori, expresii, algoritm. Instrucțiuni în C/C++.	Prelegere, video-proiecție, discuții interactive	2 ore
Algoritmi elementari: - interschimbare, - par-impar, - pătrat perfect, - prelucrarea cifrelor unui număr, - divizibilitate, - exemple de aplicații implementate în C/C++.		6 ore
Fișiere text: - deschidere, închidere fișier, - citirea datelor din fișier, - scrierea datelor în fișier, - exemple de aplicații implementate în C/C++.		2 ore
Tablouri unidimensionale: - declarare, inițializare, - parcuregere, - determinare element minim/maxim, - căutare secvențială, - sortare, - verificare elemente distincte, - operații cu mulțimi (reuniune, intersecție, diferență), - interclasare vectori, - liste, - exemple de aplicații implementate în C/C++.		4 ore
Tablouri bidimensionale: - declarare, inițializare, - parcuregere, - diagonala principală și secundară, - exemple de aplicații implementate în C/C++.		4 ore
Șiruri de caractere: - codul ASCII, - declarare, inițializare,		4 ore



• parcurgere, • funcții pentru prelucrare: strlen, strcpy, strncpy, strcat, strncat, strcmp, stricmp, strchr, strrchr, strtstr, strstr, strrev,strupr, strtolr, strtok, • funcții pentru conversii: atoi, atof, itoa, ecvt, • exemple de aplicații implementate în C/C++.		
Pointeri. Alocarea dinamică a memoriei: • adresa unei variabile, pointer, operații cu pointeri, • pointeri și tablouri, • alocarea dinamică a memoriei, • exemple de aplicații implementate în C/C++.		2 ore
Subprograme: • tip II, • tip I, • recursive, • exemple de aplicații implementate în C/C++.		2 ore
Structuri: • declarare, inițializare, accesare, • exemple de aplicații implementate în C/C++.		2 ore
Bibliografie		
1. Dr. Kris Jamsa, Lars Klander - <i>Totul despre C și C++ - manualul fundamental de programare in C și C++</i>, Ed. Teora, 2011. 2. Oana-Carmen Niculescu-Faida, Tehnici de programare C++, Editura Matrixrom, 2023 3. Diana Ștefănescu, Programarea in limbajele C/C++. Noțiuni de baza, Editura Matrix		
9. 2 Seminar/laborator	Metode de predare	Observații
Aplicații pentru utilizarea instrucțiunilor de decizie și a instrucțiunilor repetitive	Teme, aplicații, documentare online	2 ore
Aplicații pentru utilizarea algoritmilor elementari		4 ore
Aplicații pentru citirea / scrierea datelor din / în fișiere text		2 ore
Aplicații cu tablouri unidimensionale		4 ore
Aplicații cu tablouri bidimensionale		4 ore
Aplicații cu șiruri de caractere		2 ore
Aplicații cu alocarea dinamică a memoriei		2 ore
Aplicații cu subprograme		4 ore
Aplicații cu tipul struct		4 ore
Bibliografie		
1. Doina Logofatu - <i>Algoritmi fundamentali in C++ - Aplicații</i>, Editura Polirom, 2007. 2. Liviana Tudor - <i>Bazele programării in limbajul C++</i>, Editura: Matrix Rom, 2010. 3. Doru Anastasiu Popescu, Bazele programării – Limbajul C++, ISBN: 978-606-95233-2-2.		

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

- Tematica disciplinei este în concordanță cu programa analitică din alte centre universitare și asigură studenților asimilarea conceptelor de bază din domeniul programării



11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
11.4 Curs	Notele obținute la testele periodice sau parțiale		
	Nota acordată la examinarea finală	Lucrare scrisă	50%
11.5 Seminar/laborator	Teme	Practic	10%
	Două teste pe parcursul semestrului	Practic	40%
11.6 Standard minim de performanță			
Obținerea minim a notei 5 la fiecare din testele de verificare			

