

FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea „Tibiscus” din Timișoara
1.2. Facultatea	Calculatoare și Informatică Aplicată
1.3. Departamentul	Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii/Calificarea	Informatică / Asistent de cercetare în informatică - 214918; Proiectant sisteme informatice - 25110; Analist - 251201; Administrator baze de date - 252101; Administrator de rețea de calculatoare - 252301; Profesor în învățământul gimnazial - 233002; Programator - 251202

2. Date despre disciplină

2.1. Denumirea disciplinei	TEHNICI AVANSATE DE PROGRAMARE (TAP) – LIN121						
2.2. Titularul activității de curs	Lect.univ.dr. Simona Angela Apostol						
2.3. Titularul activității de laborator	Asist.univ. Ligia Tabita Pintea						
2.4. Anul de studiu	1	2.5. Semestrul	2	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	DS

3. Timpul total estimat

3.1. Numărul 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
Studiul după manual, suport de curs, bibliografie și notițe					28
Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate					32
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					34
Tutoriat					6
Examinări					4
Alte activități					
3.7. Total ore studiu individual					94
3.8. Total ore pe semestru					150
3.9. Numărul de credite					6

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Programare procedurală
4.2. de competențe	-

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Online: Zoom
5.2. de desfășurare a seminarului/laboratorului	Online: Google Classroom (code: o34isgg) Zoom

6. Competențe specifice acumulate

6.1. Competențe profesionale	C1. Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice C2. Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date.
------------------------------	--

	C3. Descrierea de concepte, teorii si modele folosite in domeniul de aplicare. C4. Analiza datelor si a modelelor.
6.2. Competențe transversale	CT1. Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatice de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse.

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

7.1. Obiectivul general al disciplinei	O1. Valorificarea cunoștințelor acumulate în cadrul cursurilor de Programare Procedurală, în vederea programării unor algoritmi avansați elaborați cu metode specifice. O2. Crearea deprinderilor de a gândi și programa folosind anumiți algoritmi. O3. Cunoașterea modului de valorificare a structurilor de date, liste, cozi, programarea dinamică în limbajul C, precum și însușirea unor algoritmi avansați în domeniul prelucrării șirurilor.
7.2. Obiectivele specifice	Cunoaștere și înțelegere (<i>cunoașterea și utilizarea adecvată a noțiunilor specifice disciplinei</i>) <ul style="list-style-type: none"> ▪ Însușirea de către studenți a principalelor metode de programare și tehnici de elaborare a algoritmilor și programelor. ▪ Se urmărește valorificarea cunoștințelor dobândite la cursul de “Programare Procedurală” în implementarea algoritmilor și pregătirea pentru potențiali programatori. Explicare și interpretare (<i>explicarea și interpretarea unor idei, proiecte, procese, precum și a conținuturilor teoretice și practice ale disciplinei</i>) <ul style="list-style-type: none"> ▪ Explicarea diferitelor tipuri de metode de programare și a unor tehnici de rezolvare a unor clase de probleme. ▪ Explicarea diferiților algoritmi de căutare și sortare. Instrumental – aplicative (<i>proiectarea, conducerea și evaluarea activităților practice specifice; utilizarea unor metode, tehnici și instrumente de investigare și de aplicare</i>) <ul style="list-style-type: none"> ▪ Aplicații în limbajul C, pentru testarea algoritmilor de căutare și sortare și tehnicilor Backtracking, Greedy, etc.

8. Conținuturi

8.1. Curs	Metode de predare	Observații
1. Metode de programare Programare ascendentă și programare descendentă, Programare modular, Programare structurată	Expunerea interactivă, problematizarea, conversația euristică, documentarea pe web, exemplificarea	1 săptămână – 2 ore
1. Metode speciale de rezolvare a unor clase de probleme Tehnica Backtracking (elementar, fără schemă, cu schemă, recursiv), Tehnica Greedy, Tehnica Divide et Impera, Metoda programării dinamice		8 săptămâni – 16 ore
3. Clase de algoritmi Analiza algoritmilor, Algoritmi de căutare, Algoritmi de sortare		4 săptămâni – 8 ore
4. Gestiunea ecranului în mod text, mod grafic		1 săptămână – 2 ore
Bibliografie :		
<ol style="list-style-type: none"> 1. Liviu Negrescu – Limbajele C si C++ pentru începatori, vol 1., partea a 2a, Ed.Albastra, 2009. 2. H.Schildt – C – manual complet, Ed.Teora, actualizata 2011. 3. Dr.K.Jamsa, L.Klander – Totul despre C si C++ - manualul fundamental de programare in C si C++, Ed.Teora, actualizata 2011. 4. Ionică Alexandru, Iordan Victoria – Tehnici de programare, Ed. Mirton, Timișoara, 2002 5. Iordan Victoria - Algoritmi și programare în C, Ed. Eurostampa, Timișoara, 2008 		
8.2. Seminar/laborator	Metode de seminarizare	Observații
1.Recapitulare. Structuri de date. Programare ascendentă și descendentă. Programare modulară. Programare structurată	Realizarea programelor prevăzute în lucrarea de laborator. Operare pe calculator.	Studenții vor primi materialul pe Classroom
2.Metoda Backtracking elementar		
3. Metoda Backtracking cu schemă, fără schemă		
4. Metoda Backtracking recursiv		
5.Recapitulare metoda Backtracking diverse probleme		
6.Tehnica Greedy		

7. Tehnica Divide et Impera	Realizarea programelor prevăzute în lucrarea de laborator. Operare pe calculator.	Studentii vor primi materialul pe Classroom
8. Metoda programării dinamice		
9. Rezolvarea problemelor utilizând diverse tehnici		
10. Analiza algoritmilor. Programe elementare rezolvate cu algoritmi		
11. Algoritmi de căutare		
12. Algoritmi de sortare		
13. Gestiunea ecranului în mod text, mod grafic		
14. Examen practic		
Bibliografie:		
1. S. Apostol, O. Crista – Tehnici Avansate de Programare, note de laborator- material destinat studentilor.		
2. Dr.K.Jamsa, L.Klander – Totul despre C si C++ - manualul fundamental de programare in C si C++, Ed.Teora, actualizata 2011.		
3. Liviu Negrescu – Limbajele C si C++ pentru incepatori, vol 1., partea a 2a, Ed.Albastra,2009.		
4. H.Schildt – C – manual complet, Ed.Teora, actualizata 2011.		
5. Scott Meyers - <i>Effective Modern C++</i> , 2014		

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

Conținutul disciplinei corespunde curriculei din alte centre universitare, din țară sau Uniunea Europeană.
Conținuturile practice (lucrări de laborator) corespund cerințelor de pe piața muncii locală.


10. Evaluare

Tip de activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Evaluarea are în vedere următoarele categorii de cunoștințe: <ul style="list-style-type: none"> ● Test cuprinzând modalități de testare a cunostintelor: Expunerea unor subiecte desemnate. Intrebari grila cu variante multiple de raspuns pe baza unor secvente de cod din programe. Rezolvarea unor probleme propuse. 	Examen scris	70 %
10.5. Seminar / laborator	Evaluarea are în vedere următoarele categorii de cunoștințe: <ul style="list-style-type: none"> ● Cunoștințe generale: utilizarea limbajului de programare `C` de bază. ● Cunoștințe de detaliu: utilizarea algoritmilor și realizarea de programe simple de utilizare a limbajului C pentru rezolvarea unor sarcini de dificultate medie. ● Cunoștințe avansate: rezolvarea de probleme complexe, de dificultate sporită. 	Evaluarea activității la laborator; Rezolvarea problemelor propuse ca teme la laboratoare. Examinare printr-un test practic pe calculator cuprinzând probleme specifice limbajului cu diferite grade de dificultate.	30 %
10.6. Standard minim de performanță			
Examinare scrisă: <ul style="list-style-type: none"> ● Pentru nota 5 este necesară tratarea minimală a fiecăreia din cele 4 categorii de subiecte. Probe practice și activitate de laborator: <ul style="list-style-type: none"> ● Pentru nota 5 este necesară obținerea unui nivel superior (min. 60%) pentru cunoștințe generale, precum și a unui nivel minim de înțelegere și utilizare a cunoștințelor de detaliu prezentate anterior (minim rezolvarea unei probleme din cele 3 propuse pe biletul de examen). 			

Data completării

28.09.2020

Semnătura titularului de curs



Semnătura titularului de laborator



Data avizării în departament

05.10.2020

Semnătura directorului de departament