FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA
1.2 Facultatea	Autovehicule Rutiere, Mecatronică si Mecanică
1.3 Departamentul	Mecatronică și Dinamica Mașinilor
1.4 Domeniul de studii	Mecatronica si Robotica -
1.5 Ciclul de studii	licenta
1.6 Programul de studii / Calificarea	Mecatronica / Mecatronica
1.7 Forma de învăţământ	IF – învăţământ cu frecvenţă

2. Date despre disciplină

2.1 Denumirea disciplinei			Informatica aplicata I	Codul disciplinei	4.00
2.2 Titularul de curs			Sl.dr.ing. Radu Donca - Radu.Donca@mdm.utcluj.ro		
2.3 Titularul activităţilor de seminar / laborator / proiect			Sl.dr.ing. Radu Donca - Radu.Donca@mdm.utcluj.ro		
2.4 Anul de studiu	1	2.5 Semestrul	1	2.6 Tipul de evaluare	examen
2.7 Regimul disciplinei	Categoria formativă				DF
	Opționalitate				DOB

3. Timpul total estimate

3.1 Număr de ore pe săptămână	4	din care:	3.2 Curs	2	3.3 Seminar	0	3.3 Laborator	2	3.3 Proiect	0
3.4 Număr de ore pe semestru	56	din care:	3.5 Curs	28	3.6 Seminar	0	3.6 Laborator	28	3.6 Proiect	0
3.7 Distribuţia fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie şi notiţe										26
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate şi pe teren										5
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii şi eseuri										35
(d) Tutorat										0
(e) Examinări										3
(f) Alte activităţi:										0

3.8 Total ore studiu individual (suma (3.7(a)...3.7(f)))	69
3.9 Total ore pe semestru (3.4+3.8)	125
3.10 Numărul de credite	5

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	
4.2 de competenţe	

5. Condiţii (acolo unde este cazul)

5.1. de desfăşurare a cursului	Proiector, calculator, tablă
5.2. de desfăşurare a seminarului / laboratorului / proiectului	Calculatoare, echipamente, software specific

6. Competenţele specifice acumulate

Competenţe profesionale	După parcurgerea disciplinei studenţii vor cunoaşte: Notiuni fundamentale de informatica aplicata utilizand software dedicat acestui scop. Vor cunoaste elementele de baza ale limbajului si sintaxa acestuia. Structuri de control alternative si repetitive. Definirea de functii. Algoritmi bazati pe recursivitate. Algoritmi de optimizare. Diviziunea problemelor complexe prin metoda divide-et-impera. După parcurgerea disciplinei studenţii vor fi capabili: Sa dezvolte programe de complexitate mica spre medie in limbajul C++. Sa identifice diversele elemente de limbaj. Sa corecteze si sa imbunatatesasca diverse coduri sursa disponibile pe internet. Sa scrie programe adaptate cerintelor date. Sa rezolve probleme de cod specifice sistemelor mecatronice. După parcurgerea disciplinei studenţii vor fi capabili: - Dezvolte diverse aplicatii folosind limbajul C/C++ - Sa utilizeze diferite medii de dezvoltare specifice - Sa simuleze mai multe tipuri de sisteme mecatronice (roboti industriali, platforme mobile, roboti umanoizi etc.) si sa proiecteze software-ul necesar controlului acestora
Competenţe transversale	CT1. Îndeplinirea sarcinilor profesionale cu identificare exactă a obiectivelor de realizat, a resurselor disponibile, condiţiilor de finalizare a acestora, etapelor de lucru, timpului de lucru şi termenelor de realizare aferente CT2. Executarea responsabilă a unor sarcini de lucru în echipă pluridisciplinară cu asumarea de roluri pe diferite paliere ierarhice CT3. Identificarea nevoii de formare continuă şi utilizarea eficientă a surselor informaţionale şi a resurselor de comunicare şi formare profesională asistată (portaluri Internet, aplicaţii software de specialitate, baze de date, cursuri on-line etc.) atât în limba română cât şi într-o limbă de circulaţie internaţională

7. Rezultatele aşteptate ale învăţării

Cunoștințe	- Studentul identifică și descrie principiile de funcționare și proiectare a sistemelor de reglare automată. - Studentul identifică și descrie sistemele și subsistemele mecatronice, a principiilor și metodelor de proiectare a acestora, respectiv a tehnicilor, instrumentelor specifice și practicilor moderne în concepția sistemelor și subsistemelor mecatronice. - Studentul identifică și descrie tehnici computaționale specifice în concepția sistemelor și subsistemelor automate.
Abilități	- Studentul alege și proiectează un sistem de reglare automată. - Studentul alege și aplică metode de evaluare a proiectelor de sisteme mecatronice. - Studentul alege și aplică tehnici computaționale specifice sistemelor mecatronice.
Responsabilitate și autonomie	- Studentul decide cu privire la alegerea tipului de sistem de reglare automată ce trebuie implementat în anumite aplicații și este responsabil de funcționarea corespunzătoare a acestuia. - Studentul elaborează schemele, diagramele structurale și de funcționare, a reprezentărilor grafice și a documentelor tehnice specifice. - Studentul decide cu privirea la alegerea algoritmilor utilizați și implementarea acestora folosind tehnici computaționale specifice.

8. Obiectivele disciplinei (reieşind din grila competenţelor specifice acumulate)

8.1 Obiectivul general al disciplinei	Proiectarea și implementarea aplicatiilor informatice specifice mecatronicii
8.2 Obiectivele specifice	Cunoașterea limbajului C/C++, Utilizarea platformelor de dezvoltare specifice

9. Conţinuturi

9.1 Curs	Nr. ore	Metode de predare	Observaţii
1. Prezentarea disciplinei, bibliografie curs, acces literatura. Notiunea de algoritm. Scheme logice.	2	Videoproiector, tablă, discuții, Internet,	
2. Principiile programarii structurate. Scheme logice. Variabile simple.	2		
3. Elemente de baza ale limbajului. Operatiuni de intrare-iesire. Afisarea datelor pe ecran. Citirea informatiilor de la tastatura. Structuri de control liniare. Tipuri numerice de date. Operatiuni aritmetice.	2		
4. Structura alternativa. Instructiunile If..Else si Switch..Case. Tipul boolean. Operatiuni logice.	2		
5. Structura repetitiva. Ciclul For. Siruri de date. Indexare. Operatiuni cu siruri.	2		
6. Structura repetitiva. Ciclul While si Do..While. Algoritmi de sortare. Matrici.	2		
7. Caractere. Codul ASCII. Operatiuni cu caractere. Siruri de caractere. Aplicatii specifice.	2		
8. Subrutine. Rezolvarea modulara a unei probleme. Variabile locale. Variabile globale.	2		
9. Functii. Returnarea rezultatului. Parametri. Transmitere prin valoare sau prin referinta.	2		
10. Recursivitate. Functii matematice rezolvate recursiv. Divide-et-impera. Turnurile din Hanoi.	2		
11. Fisiere. Acces. Citire date din fisier. Salvare rezultate pe disc.	2		
12. Grafica. Initializare driver grafic, pixeli, linii. cercuri. Animatii simple.	2		
13. Programare orientata obiect. Clase/obiecte. Metode. Constructori.	2		
14. Programare orientata obiect. Incapsulare. Mostenire. Polimorfism.	2		

Bibliografie

1. Stanley Lippman - C++ Primer (5th Edition) - Addison-Wesley Professional, 2012
2. Bjarne Stroustrup - Programming: Principles and Practice Using C++ (2nd Edition) - Addison-Wesley Professional; 2nd edition 2014
3. Scott Meyers - Effective C++: 55 Specific Ways to Improve Your Programs and Designs 3rd Edition - Addison-Wesley Professional; 3rd edition, 2005
4. Andrei Alexandrescu - Modern C++ Design: Generic Programming and Design Patterns Applied 1st Edition, 2001
5. Thomas H. Cormen - Introducere in Algoritmi – Byblos, 2004
6. Gabriel Braharu - Limbajul C++ Pentru Incepatori (Volumul I + Volumul II), 2020
7. w3schools.com – C++ Tutorial - https://www.w3schools.com/cpp/

9.2 Seminar / laborator / proiect		Metode de predare	Observaţii
1. Introducere – reguli de protecția muncii, prezentarea platformei de lucru. Familiarizarea cu mediul de dezvoltare.	2		
2. Rezolvarea unor algoritmi simpli folosind scheme logice si descriere in limbaj pseudocod. Tabel de urmarire variabile.	2		
3. Familiarizarea cu limbajul. Citire de date. Procesare matematica. Diferente intre tipurile numerice. Afisare rezultate.	2		
4. Probleme cu factor decizional. Utilizarea structurii alternative. Probleme cu operatii logice.	2		

5. Probleme cu cicluri cu numar fix de pasi. Gruparea datelor in siruri. Operatiuni aritmetice (suma/produs/medie) pe sir. Exemple de probleme de indexare.	2		
6. Cicluri cu test initial.Cicluri cu test final. Rezolvarea unor probleme tipice. Tablouri bidimensionale.	2		
7. Prelucrarea datelor de tip caracter. Modificari pe un text dat. Probleme tipice pe string-uri. Conversii text – numeric si invers.	2		
8. Exercitii de detinire a unor subrutine proprii. Descompunerea uneiprobleme complexe in module simple. Testul vizibilitatii variabilelor.	2		
9. Implementarea unor functii matematice. Testarea rezultatelor. Apel de parametri prin valoare si prin referinta.	2		
10. Rezolvarea unor functii matematice definite recursiv. CMMDC/CMMMC. Sirul lui Fibonacii rezolvat recursiv. Probleme tipice de recursivitate. Algoritmi recursivi.	2		
11. Fisiere tip text. Probleme tipice. Modificarea unor probleme pentru citirea datelor din fisier si scrierea rezultatelor in fisier.	2		
12. Probleme de grafica. Exercitii legate de culori. Exercitii de desenare a formelor geometrice complexe. Animatii simple.	2		
13. Crearea unei clase. Instantierea unui obiect. Exemple. Probleme tipice.	2		
14. Dezvoltarea unei clase. Exemple declase derivate. Aplicatii practice ale mostenirii si polimorfismului.	2		

Bibliografie

1. Stanley Lippman - C++ Primer (5th Edition) - Addison-Wesley Professional, 2012
2. Bjarne Stroustrup - Programming: Principles and Practice Using C++ (2nd Edition) - Addison-Wesley Professional; 2nd edition 2014
3. Scott Meyers - Effective C++: 55 Specific Ways to Improve Your Programs and Designs 3rd Edition - Addison-Wesley Professional; 3rd edition, 2005
4. Andrei Alexandrescu - Modern C++ Design: Generic Programming and Design Patterns Applied 1st Edition, 2001
5. Thomas H. Cormen - Introducere in Algoritmi – Byblos, 2004
6. Gabriel Braharu - Limbajul C++ Pentru Incepatori (Volumul I + Volumul II), 2020
7. w3schools.com – C++ Tutorial - https://www.w3schools.com/cpp/
8. Probleme de informatica - enunturi si rezolvari - http://info.mcip.ro/

10. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatorilor reprezentativi din domeniul aferent programului

Conținutul este actualizat conform dezvoltărilor din domeniu și cu necesitățile angajatorilor din domeniu.

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
11.4 Curs	Examenul constă dintr-un test grilă (0.5 oră) subiect de sinteza (0.5 ora), probleme (1 ora);	Examen scris	50%
11.5 Seminar/Laborator	Soluții proiectare software	Evaluare laborator și teme	50%

/Proiect	Calitate documentare	de casă	

11.6 Standard minim de performanţă:
Rezolvarea unor probleme specifice mecatronicii pe baza utilizării unor sisteme adecvate de calcul.

Data completării:	**Titulari**	**Titlu Prenume NUME**	**Semnătura**
16.09.2026	Curs	S.L.dr.ing. Donca Radu	
	Aplicații	S.L.dr.ing. Donca Radu	

Data avizării în Consiliul Departamentului de Mecatronica si dinamica masinilor
28.05.2026
_____

Director Departament
prof. dr. ing. Mircea BARA

Data aprobării în Consiliul Facultății de Autovehicule Rutiere, Mecatronică si Mecanică
16.09.2026
_____

Decan
prof. dr. ing. Nicolae FILIP