

FIȘA DISCIPLINEI

Anul universitar 2021-2022
Anul de studiu 2/ Semestrul 1

1. Date despre program

1.1. Instituția de învățământ	Universitatea „1 Decembrie 1918”
1.2. Facultatea	de Științe Exacte și Inginerești
1.3. Departamentul	de Informatică, Matematică și Electronică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Masterat
1.6. Programul de studii/calificare	Programare avansata si baze de date/251202 /251205 / 251206

2. Date despre disciplină

2.1. Denumirea disciplinei	Verificarea și validarea sistemelor software		2.2. Cod disciplină	MI206.2			
2.3. Titularul activității de curs	Lect.univ.dr. Domsa Ovidiu						
2.4. Titularul activității de laborator	Lect.univ.dr. Domsa Ovidiu						
2.5. Anul de studiu	II	2.6. Semestrul	I	2.7. Tipul de evaluare (E/C/VP)	E	2.8. Regimul disciplinei (O – obligatorie, Op – opțională, F – facultativă)	

3. Timpul total estimat

3.1. Numar ore pe saptamana	3	din care: 3.2. curs	2	3.3. laborator	1
3.4. Total ore din planul de învățământ	42	din care: 3.5. curs	28	3.6. seminar/laborator	14
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					60
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					30
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					33
Tutoriat					-
Examinări					10
Alte activități					-

3.7 Total ore studiu individual	133
3.9 Total ore pe semestru	175
3.10 Numărul de credite	7

4. Precondiții

4.1. de curriculum	Programarea calculatoarelor
4.2. de competențe	C1 Utilizarea elementelor fundamentale referitoare la dispozitivele, circuitele, sistemele, instrumentația și tehnologia electronică C3.4 Elaborarea de programe într-un limbaj de programare general și/sau specific, pornind de la specificarea cerințelor și până la execuție, depanare și interpretarea rezultatelor în corelație cu procesorul utilizat

5. Condiții

5.1. de desfășurare a cursului	Videoproiector, Laptop, Copiator, Woofers și cursuri multimedia
5.2. de desfășurarea a seminarului/laboratorului	Sala dotată cu videoproiector/tabla, standuri de laborator specifice, rețea de calculatoare conectată la internet

6. Competențe specifice acumulate

Competențe profesionale	C4.1 Definirea conceptelor, principiilor și metodelor folosite în domeniile: programarea calculatoarelor, limbaje de nivel înalt și specifice, grafică, arhitecturi hardware reconfigurabile C3.3 Rezolvarea problemelor practice concrete care includ elemente de structuri de date și algoritmi, programare și utilizare de testare C4.4 Utilizarea criteriilor de performanță adecvate pentru evaluarea, inclusiv prin simulare, a hardware-ului și software-ului unor sisteme dedicate sau a unor activități de servicii în care se folosesc module de testare C3.5 Realizarea de proiecte de testare care implică diferite componente software (programare)
Competențe transversale	

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	La terminarea cursului, studenții trebuie să aibe: a) cunoștințe sistematice despre metodologiile de realizare a produselor soft b) familiarizarea cu conceptele și preocupările moderne în scrierea softului de aplicație c) cunoștințe despre tipurile de modele și instrumentele de modelare folosite în dezvoltarea aplicațiilor soft de testare d) motivația necesară privind importanța documentării produselor soft
7.2 Obiectivele specifice	Să știe să utilizeze mediile de programare în scopul realizării de produse software; Să știe să folosească tehnici de modelarea conceptuală a datelor; Să știe să facă studii de caz complete de testare software; Să știe să interpreteze rezultatele obținute.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
Curs 1 Privire de ansamblu 1.1 Defecte și eșecuri. 1.2 Combinații de intrare și condițiile 1.3 Costuri 1.4 Personalul implicat	Proiectare, dezvoltare proiect.	4 ore
Curs 2- Istoria testării software	Proiectare, dezvoltare proiect.	4 ore
Curs 3 Metode de testare 3.1 Încercări statice vs dinamice 3.2 Abordarea box 3.2.1 Testare White-box 3.2.2 Testare Black-box 3.2.2.1 Testarea Visual 3.2.3 Testarea Grey-box	Proiectare, dezvoltare proiect.	4 ore
Curs 4 Nivele de testare 4.1 Unitatea de testare 4.2 Testarea de integrare 4.3 Testarea sistemului 4.4 Acceptarea de testare;	Proiectare, dezvoltare proiect.	4 ore
Curs 5 Tipuri de testare 5.1 Testarea de instalare 5.2 Teste de compatibilitate 5.3 Teste de fum și testare de bun-simț 5.4 Testarea de regresie 5.5 Acceptarea de testare 5.6 Testarea Alpha 5.7 Testare Beta 5.8 Testare funcțională vs non-funcțională 5.9 Controlul distructiv 5.10 Testele de performanță software-ului	Proiectare, dezvoltare proiect.	4 ore

5.11 Teste de utilizabilitate 5.12 Testare accesibilitate 5.13 Testarea securității 5.14 Internaționalizare și localizare 5.15 Testare de dezvoltare 5.16 Testare A / B		
Curs 6 Procesul de testare 6.1 Model de dezvoltare cascadă 6.2 Model de dezvoltare Agile sau Extreme 6.3 Testare Top-down and bottom-up 6.4 Exemplu de testare ciclică	Proiectare, dezvoltare proiect.	4 ore
Curs 7 Testare automată 7.1 Instrumente de testare 7.2 Măsurarea în testarea software-ului	Proiectare, dezvoltare proiect.	4 ore
8.2 Laborator		
Laborator 1- 3 Metode de testare -aplicații	Aplicatii, proiect	2 ore
Laborator 2- Nivele de testare-aplicații	Aplicatii, proiect	2 ore
Laborator 3- Tipuri de testare-aplicații	Aplicatii, proiect	2 ore
Laborator 4- Tipuri de testare-aplicații	Aplicatii, proiect	2 ore
Laborator 5- Procesul de testare-aplicații	Aplicatii, proiect	2 ore
Laborator 6- Testare automată-aplicații	Aplicatii, proiect	2 ore
Laborator 7- Testare automată-aplicații	Aplicatii, proiect	2 ore
8.2 Bibliografie		
1. I. Sommerville. Software Engineering (6th, 7th, 8th, 9th editions). Addison Wesley (2001, 2004, 2006, 2010). 2. T. Lethbridge, R. Laganier. Object-Oriented Software Engineering: Practical Software Development using UML and Java (2nd edition). McGraw-Hill, 2005. http://www.lloseng.com . 3. E. Currie. The essence of Z. Prentice Hall, 1999. 4. E.N. Todoran. Inginerie software: studii în prototipizare și specificare formală. Mediamira, ClujNapoca, 2006. 3. Software Engineering Body of Knowledge, IEEE, 2004. [http://www.swebok.org].		

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

- *Elaborata pe baza consultării fișelor disciplinei similare din Centrele universitare Cluj- Napoca (UBB, UT), Timisoara (UP), Bucuresti (UP), Craiova*
- *Pe baza recomandarilor comisiei de evaluare a disciplinei*
- *Feedback din partea studentilor*

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<i>Evaluare finala</i>	<i>Portofoliu de lucrari practice</i>	100%
10.6 Standard minim de performanță:			
Conceperea unui pland de documentare si de realizarea a unuie aplicatii hardware/ software. Similar oricarui sistem de management al proiectelor, se urmareste obtinerea unui plan de munca, cu etape, chei de verificare, cu plan de indeplinire a cerintelor hardware/ software formulate.			

Recuperarea laboratoarelor se face prin proiecte suplimentare, pe parcursul semestrului.

Data completării
23.09.2021

Semnătura titularului de curs

Semnătura titularului de seminar

Data avizării în departament

Semnătura directorului de departament