

FIȘA DISCIPLINEI

Verificarea si Validarea Sistemelor Software

Anul universitar 2026-2027

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babes-Bolyai
1.2. Facultatea	Facultatea de Matematica si Informatica
1.3. Departamentul	Departamentul de Informatica
1.4. Domeniul de studii	Informatica
1.5. Ciclul de studii	Licenta
1.6. Programul de studii / Calificarea	Informatica
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei	Verificarea si Validarea Sistemelor Software	Codul disciplinei	MLE5014		
2.2. Titularul activităților de curs	Conf. Dr. Vescan Andreea				
2.3. Titularul activităților de seminar	Conf. Dr. Vescan Andreea				
2.4. Anul de studiu	3	2.5. Semestrul	6	2.6. Tipul de evaluare	Examen
2.7. Regimul disciplinei	Obligativu	2.8. Tipul disciplinei	Disciplină fundamentală (DF)		

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/ proiect	2
3.4. Total ore din planul de învățământ	48	din care: 3.5. curs	24	3.6 seminar/laborator	24
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					33
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					33
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					34
Tutoriat (consiliere profesională)					7
Examinări					20
Alte activități					0
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				127	
3.8. Total ore pe semestru				175	
3.9. Numărul de credite				7	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Programare orientată pe obiecte, Metode avansate de programare, Sisteme de proiectare și implementare, Programare Web
4.2. de competențe	Abilități în medii de programare la nivel înalt orientat obiect

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Video proiector; acces la internet
5.2. de desfășurare a seminarului/ laboratorului	Laborator cu calculatoare; diverse instrumente pentru activități de testare/verificare

6.1. Competențele dobândite în urma absolvirii programului de studii (se preiau din planul de învățământ)¹

¹ Se vor prelua din Planul de învățământ al programului de studii acele competențe profesionale și/sau transversale la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa disciplinei. Pentru fiecare

Competențe profesionale	
Codul competenței	Competență
CP1	crează softuri
CP3	analizează specificații software
CP9	remediază erorile din software
Competențe transversale	
Codul competenței	Competență
CT1	Lucrează independent
CT3	Gândește analitic

6.2. Rezultatele învățării specifice programului de studii (se preiau din planul de învățământ)²

Rezultatele învățării vizate prin disciplină		
Codul competenței	Cunoștințe și înțelegere (Knowledge and understanding)	Abilități academice specifice (Specific academic skills)
CP1	1. Studentul/absolventul identifică, explică și argumentează concepte fundamentale de structuri de date, algoritmi și paradigme de programare, precum și a arhitecturii calculatoarelor.	1. Studentul/absolventul elaborează, dezvoltă și demonstrează soluții software complexe utilizând algoritmi eficienți și paradigme diverse de programare.
CP3	2. Studentul/absolventul numește, oferă exemple, concluzionează, specifică, recunoaște și argumentează critic metodele de proiectare și management al proiectelor informatice complexe, utilizând strategii moderne.	2. Studentul/absolventul inițiază, pregătește, realizează, propune metode de dezvoltare a proiectelor informatice complexe. Studentul/absolventul realizează rapoarte profesionale specifice.
CP9	3. Studentul/absolventul alege, descrie, analizează și explică paradigmele moderne de programare, inclusiv programarea funcțională, orientată pe obiect și paralelă, utilizând limbaje și framework-uri actuale.	3. Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software scalabile și utilizează eficient resursele hardware și software.
CT1	Studentul/absolventul are cunoștințele necesare pentru a înțelege și soluționa probleme complexe, pentru a planifica și organiza procese avansate în diverse domenii.	Absolventul este capabil să identifice probleme complexe și să examineze probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții.
CT3	Studentul/absolventul are cunoștințele necesare pentru a înțelege și soluționa probleme complexe, pentru a planifica și organiza procese avansate în diverse domenii.	Absolventul are abilitatea de a aplica reguli generale unor probleme specifice și de a produce soluții relevante.

7. Rezultatele învățării specifice disciplinei

Cunoștințe și înțelegere (Knowledge and understanding)
1. Absolventul are abilitatea de a înțelege și comunica eficient informațiile.
2. Absolventul are capacitatea de a observa și obține informații din diverse surse.

competență se va prelua întregul enunț, inclusiv codul competenței, cu formularea care apare în planul de învățământ, fără modificări. Dacă nu se preia nici o competență din oricare din cele două categorii, se șterge linia din tabel aferentă acelei categorii.

² Se menționează rezultatele învățării specifice programului de studiu la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa. Enunțurile, preluate fără modificări din Planul de învățământ în funcție de tipul disciplinei (DF/DS/DC) se trec în dreptul competenței asociate.

Abilități academice specifice (Specific academic skills)

1. Absolventul are abilitatea de a crea teste automate de diferite nivele de granularitate pentru asigurarea calității sistemelor dezvoltate.
2. Absolventul este familiar cu instrumente folosite pentru testarea, depanarea, validarea aplicații software.
3. Absolventul cunoaște metodele de testare și verificare a sistemelor software.
4. Absolventul este familiarizat cu instrumentele de management de proiect, sistemele de control al versiunilor, precum și conceptele, metodele, instrumentele de continuous integration/continuous delivery (CI/CD).

8. Conținuturi

8.1 Curs	Metode de predare - învățare	Observații ³
Verificare și validare. Inspectarea programelor	Expunere interactivă Explicație Conversație Demonstratie didactica	
Testarea programelor (1): conceptul de testare a programelor; testare unitară: criteriile de testare – testare cutie neagră,	Expunere interactivă Explicație Conversație Demonstratie didactica	
Testare programelor (2): conceptul de testare program; testare unitară: criteriile de testare – testare cutie albă (continuare)	Expunere interactivă Explicație Conversație Demonstratie didactica	
Testarea programelor (3): Niveluri de testare (unitate, integrare, sistem, regresie, acceptare)	Expunere interactivă Explicație Conversație Demonstratie didactica	
Testarea aplicațiilor web	Expunere interactivă Explicație Conversație Demonstratie didactica	
Testare agilă. Testarea scripturilor versus testarea exploratorie	Expunere interactivă Explicație Conversație Demonstratie didactica	
Symbolic execution	Expunere interactivă Explicație Conversație Demonstratie didactica	
Model checking	Expunere interactivă Explicație Conversație Demonstratie didactica	
Teoria corectitudinii programului. Evoluția conceptului de corectitudine a programului. --- Metoda lui Floyd pentru a demonstra corectitudinea. --- Metoda de axiomatizare a lui Hoare	Expunere interactivă Explicație Conversație Demonstratie didactica	

³ De exemplu aspecte organizatorice, recomandări pentru studenți, aspecte specifice legate de curs/seminar cum ar fi invitarea unor practicieni în domeniu etc.

--- Dijkstra: cea mai slabă condiție prealabilă. Rafinament în trepte din specificații		
Calitatea programului	Expunere interactivă Explicație Conversație Demonstratie didactica	
Activități legate de verificare/testare: abilități de testare tehnică, abilități de testare soft, feedback. Această activitate se realizează în colaborare a profesorului cu studentii.	Expunere interactivă Explicație Conversație Demonstratie didactica	
Pregătirea examenului final.	Expunere interactivă Explicație Conversație Demonstratie didactica	

Bibliografie

Bibliography

Books

- [Fre10] FRENTIU, M., Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010
- [Pres10] R. S. Pressman, Software engineering: a practinioner's approach, seventh edition, Higher Education, 2010
- [Crs09] L. Crispin, J. Grecory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009
- [You08] M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008
- [Nai08] K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
- [Kat08] J. P. Katoen, Principles of Model Checking, MIT Press, May 2008
- [Pat05] R. Patton, Software Testing, Sams Publishing, 2005
- [Mye04] Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004
- [Brn02] I. Brnstein, Practical software testing, Springer, 2002
- [Mor90] Morgan, C., Programing from Specifications, Prentice Hall, NewYork, 1990.
- [Dro89] DROMEY G., Program Derivation. The Development of Programs From Specifications, Addison Wesley Publishing Company, 1989.

Articles

- [Kin75] J. Darringer, J. King, Applications of symbolic execution to program testing, 1975
- [Dij75] DIJKSTRA, E., Guarded commands, nondeterminacy and formal derivation of programs, CACM, 18(1975), 8, pg.453-457.
- [Hoa69] HOARE, C.A.R., An axiomatic basis for computer programming, CACM, 12(1969), pg.576-580, 583.

Tutorials

În timpul prelegerilor/seminarelor/laboratoarelor vor fi oferite tutoriale pentru fiecare temă.

8.2 Seminar / laborator	Metode de predare - învățare	Observații
Seminar 1/Laborator 1 Inspectare Instrument de inspectare	Prezentare, Conversație, Problematizări, Descoperire, Alte metode – studiu individual, exerciții	

Seminar 2/Laborator 2 Cazuri de testare folosind Black-box Testing (BBT) Instrument de integrare continuă (Jenkins)	Prezentare, Conversație, Problematizări, Descoperire, Alte metode – studiu individual, exerciții	
Seminar 3/Laborator 3 Cazuri de testare folosind White-box Testing (WBT) Instrument de integrare continuă (Jenkins)	Prezentare, Conversație, Problematizări, Descoperire, Alte metode – studiu individual, exerciții	
Seminar 4/Laborator 4 Niveluri de testare - Testare de integrare Instrument de integrare continuă (Jenkins)	Prezentare, Conversație, Problematizări, Descoperire, Alte metode – studiu individual, exerciții	
Seminar 5/Laborator 5 Testare web Instrument de testare web (de exemplu, Selenium Web Driver) Instrument de integrare continuă (Jenkins)	Prezentare, Conversație, Problematizări, Descoperire, Alte metode – studiu individual, exerciții	
Seminar 6/Laborator 6 Corectitudine. Analiza statica ESCJava2, JML	Prezentare, Conversație, Problematizări, Descoperire, Alte metode – studiu individual, exerciții	
Bibliografie		
Vedeți referințele din Prelegeri. Remarcă. Pentru fiecare seminar, studenții trebuie să fie pregătiți. Diverse articole/capitole din cărți trebuie citite înainte de fiecare seminar.		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare ⁴	9.2 Metode de evaluare ⁵	9.3 Pondere din nota finală
9.1 Curs	La sfârșitul semestrului un scris examenul va da Nota E.	Examen Scris	50%
9.2 Seminar/laborator	Activitatea de la seminarul, constând din participare la rezolvarea exercitiilor iar discuțiile vor fi notate cu nota S.	Seminar = Nota pentru Activitatea seminarului	25%
	Activitatea de la laboratoare, constând din participare la rezolvarea exercitiilor și discuții, vor fi notate cu nota L.	Laborator activitate	25%
9.3 Puncte bonus	Studenții vor avea posibilitatea de a obține puncte bonus la nota finală pentru activități suplimentare care sunt legate de verificarea și validarea sistemelor software: cercetare/raport de conducere și diverse activități în timpul prelegerilor. De asemenea, ar putea fi selectat un proiect de cercetare și dezvoltare.	Puncte bonus	Puncte bonus la nota finală (după obținerea minimului final de nota 5).

⁴ Criteriile de evaluare trebuie să reflecte direct rezultatele învățării vizate la nivel de program de studii, respectiv la nivel de disciplină. Mai concret, se evaluează achizițiile de învățare menționate în rezultatele anticipate ale învățării.

⁵ Se recomandă stabilirea atât a metodelor de evaluare finală, cât și a strategiei de evaluare pe parcurs.

9.6 Standard minim de promovare

Observație.

- Seminarul/Temele de laborator/Temele practice de laborator nu pot fi refăcute în sesiunea de reluare.
- Examenele scrise pot fi susținute în timpul sesiunii de reluare.
- Studenți din anii anteriori până în anul universitar curent
 - o Toate regulile de mai sus se aplică studenților din anii anteriori.
 - o Temele de seminar/laborator și activitatea practică de laborator trebuie refăcute în timpul activității didactice (în cele 12 săptămâni înainte de sesiunea normală).
- Activitate de laborator: fiecare student va veni cu propria semigrupă.
- Activitate de laborator: 3 din 6 laboratoare trebuie predate.
- Predare cu întârziere a sarcinilor va fi penalizată. Sunt permise maxim 4 săptămâni pentru a preda o tema. După termenul limită, tema va fi notată cu 0.
- Nota finală calculată cu formula dată trebuie să fie minim 5 pentru a promova examenul.
Nota finală=50%Examen scris+25%Seminar+25%Laborator
- Participați la 75% din activitățile de seminar în timpul semestrului ȘI participați la 90% din activitățile de laborator în timpul semestrului.

10. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)⁶

	<input type="radio"/> Eticheta generală pentru Dezvoltare durabilă							
								
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
								Nu se aplică nici o etichetă
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Data completării:

22/05/2026

Semnătura titularului de curs

Conf. Dr. Vescan Andreea

Semnătura titularului de seminar

Conf. Dr. Vescan Andreea

Data avizării în departament:

Semnătura directorului de departament

Conf.dr. Adrian STERCA

⁶ Selectați o singură etichetă, cea care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivește cel mai bine disciplinei. Dacă disciplina tratează tema dezvoltării durabile la modul general (de ex. prin prezentarea/introducerea cadrului general al dezvoltării durabile etc.) atunci se poate alocă eticheta generală de Dezvoltare Durabilă. Dacă niciuna dintre etichete nu descrie disciplina, selectați ultima opțiune: „Nu se aplică nici o etichetă”.