

FIȘA DISCIPLINEI

1. Date despre program

| | |
|--|---|
| 1.1. Instituția de învățământ superior | Universitatea din București |
| 1.2. Facultatea | Facultatea de Matematică și Informatică |
| 1.3. Departamentul | Informatică |
| 1.4. Domeniul de studii | Informatică |
| 1.5. Ciclul de studii | Master |
| 1.6. Programul de studii / Calificarea | Securitate și logică aplicată |
| 1.7. Forma de învățământ | ZI |

2. Date despre disciplină

| | | | | | | | | |
|---|--------------------------------|----------------|----|------------------------|---|--------------------------|------------------------------|----|
| 2.1. Denumirea disciplinei | Verificarea programelor | | | | | | | |
| 2.2. Titularul activităților de curs | Conf. dr. Denisa Diaconescu | | | | | | | |
| 2.3. Titularul activităților de laborator | Conf. dr. Denisa Diaconescu | | | | | | | |
| 2.4. Anul de studiu | I | 2.5. Semestrul | II | 2.6. Tipul de evaluare | E | 2.7. Regimul disciplinei | Conținut ²⁾ | DF |
| | | | | | | | Obligativitate ³⁾ | DI |

3. Timpul total estimat (ore pe semestru al activităților didactice)

| | | | | | |
|---|-----|---------------------|----|----------------------------------|-----|
| 3.1. Număr de ore pe săptămână | 3 | din care: 3.2. curs | 2 | 3.3. seminar/ laborator/ proiect | 1 |
| 3.4. Total ore pe semestru | 42 | din care: 3.5. curs | 28 | 3.6. SF | 14 |
| Distribuția fondului de timp | | | | | Ore |
| 3.4.1. Studiul după manual, suport de curs, bibliografie și notițe – nr. ore SI | | | | | 50 |
| 3.4.2. Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren | | | | | 34 |
| 3.4.3. Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri | | | | | 50 |
| 3.4.4. Examinări | | | | | 4 |
| 3.4.5. Alte activități | | | | | |
| 3.7. Total ore studiu individual | 138 | | | | |
| 3.8. Total ore pe semestru | 180 | | | | |
| 3.9. Numărul de credite | 6 | | | | |

4. Precondiții (acolo unde este cazul)

| | |
|--------------------|----------------------------------|
| 4.1. de curriculum | Logică pentru informatică |
| 4.2. de competențe | Capacitate de analiză și sinteză |

5. Condiții (acolo unde este cazul)

| | |
|---|-------------------------------|
| 5.1. de desfășurare a cursului | Sală dotată cu videoproiector |
| 5.2. de desfășurare a seminarului/ laboratorului/ proiectului | Laborator de informatică |

6. Competențe specifice acumulate

| | |
|-------------------------|---|
| Competențe profesionale | CP1 Înțelegerea diferenței dintre testarea și verificarea programelor. CP2 Stabilirea deosebirilor dintre o proiectare bună și verificare formală. CP3 Clasificarea și aprofundarea diferitelor forme de verificare formală. |
| Competențe transversale | CT1 Aplicarea regulilor de muncă riguroasă și eficientă, manifestarea unor atitudini responsabile față de domeniul științific și didactic, pentru valorificarea optimă și creativă a propriului potențial în situații specifice, cu respectarea principiilor și a normelor de etică profesională. CT2 Utilizarea eficientă a surselor informaționale și a resurselor de comunicare și formare profesională asistată, atât în limba română, cât și într-o limbă de circulație internațională. |

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

| | |
|--|---|
| 7.1. Obiectivul general al disciplinei | În cadrul acestui curs studenții învață ce înseamnă o aplicație software corectă și cum pot dezvolta astfel de aplicații. |
| 7.2. Obiectivele specifice | La acest curs studenții învață o serie de tehnici avansate de validare și verificare a programelor, tehnici ce depășesc testarea ad-hoc. Accentul este pus pe tehnici care pot fi automatizate, sau parțial automatizate. Cursul prezintă mai multe metode uzuale de definire a semnificării unui program care permit definirea și demonstrarea corectitudinii programelor. |

8. Conținuturi

| 8.1. Curs | Metode de predare | Observații |
|---|--|---|
| 1. Limbaje de programare. Sintaxă; semantică. | Explicația Descrierea și exemplificarea Demonstrația | Resurse folosite - Videoproiector - Calculator - Tablă |
| 2. Semantica statică – tipuri. Tipuri pentru securitate. | | |
| 3. Semantica dinamică – sisteme de tranziție. | | |
| 4. Semantica operațională structurală. | | |
| 5. Semantica axiomatică – logici Hoare/Floyd. | | |
| 6. Verificarea programelor cu stări – logica separării. | | |
| 7. Verificarea programelor concurente (bazate pe semantica axiomatică) – non-interferență, condiții Owicki-Gries, logica separării concurentă | | |
| 8. Verificarea programelor concurente (bazate pe semantica dinamică) – logici temporale, LTL, model checking | | |
| 9. Problema exploziei stărilor – abstractizare, CEGAR | | |
| 10. Execuție simbolică – constrângeri de drumuri, satisfiabilitatea modulo teorii (SMT), probleme de decizie, Nelson-Oppen | | |
| 11. Verificarea lightweight: execuție „concolic”, generarea de teste | | |
| 12. Analiza dinamică pentru programe concurente | | |
| Bibliografie: <i>Logic in Computer Science: Modeling and Reasoning about Systems, 2nd edition</i> , Michael Huth, Mark Ryan, Cambridge University Press, 2004. <i>Verification of Sequential and Concurrent Programs, 3rd edition</i> , Krzysztof R. Apt, Frank S. de Boer, Ernst-Rüdiger Olderog, Springer. <i>Systems and Software Verification: Model-Checking Techniques and Tools</i> , Berard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., Schnoebelen, P., Springer, 2001. <i>Practical Foundations for Programming Languages, 2nd edition</i> , Robert Harper, Cambridge University Press, 2016. <i>Model Checking</i> , Edmund M. Clarke, O. Grumberg, Doron A. Peled, MIT Press, 2000. | | |
| 8.2. Seminar | Metode de predare-învățare | Observații |
| | | |
| Bibliografie: | | |
| 8.3. Laborator | Metode de transmitere a informației | Observații |
| Aplicații practice ale temelor prezentate în cadrul cursului | Studiul de caz Exercițiul | Resurse folosite - Videoproiector - Calculator - Tablă |
| Bibliografie: Aceeși ca la curs | | |
| 8.4. Proiect | Metode de predare-învățare | Observații |
| | | |
| Bibliografie: | | |

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

Noțiunile introduse în acest curs vor contribui la abilitățile studenților de a dezvolta produse software de calitate, pregătindu-i mai bine pentru piața muncii. Mai mult, cursul oferă și acces la dezvoltări actuale în domeniul verificării, pregătind astfel studenții pentru programe doctorale sau pentru a deveni membrii ai departamentelor de cercetare în firme.

10. Evaluare

| Tip activitate | 10.1. Criterii de evaluare | 10.2. Metode de evaluare | 10.3. Pondere din nota finală |
|---|--------------------------------|----------------------------------|-------------------------------|
| 10.4. Curs | Evaluare finală | Lucrare scrisă | 60% |
| 10.5.1. Seminar | | | |
| 10.5.2. Laborator | Evaluarea temelor de laborator | Prezentarea temelor de laborator | 40% |
| 10.5.3. Proiect | | | |
| 10.6. Standard minim de performanță. Obținerea mediei 5. | | | |

Data completării

Semnătura titularului de curs
Conf. dr. Denisa Diaconescu

Semnătura de seminar/laborator
Conf. dr. Denisa Diaconescu

Data avizării în
departament
.....

Director de departament
Conf.dr. Alin Ștefănescu

Notă:

- 1) Regimul disciplinei (conținut) - *pentru nivelul de licență se alege una din variantele: DF* (disciplină fundamentală) / **DD** (disciplină din domeniu) / **DS** (disciplină de specialitate) / **DC** (disciplină complementară).
- 2) Regimul disciplinei (obligativitate) - *se alege una din variantele: DI* (disciplină obligatorie) / **DO** (disciplină opțională) / **DFac** (disciplină facultativă).
- 3) SI – studiu individual; TC – teme de control; AA – activități asistate; SF – seminar față în față; L – activități de laborator; P – proiect, lucrări practice.