**FIŞA DISCIPLINEI**

**1. Date despre program**

|  |  |
| --- | --- |
| 1.1 Instituţia de învăţământ superior | Universitatea POLITEHNICA din Bucureşti |
| 1.2 Facultatea | Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei |
| 1.3 Departamentul | Departamentul de Electronică Aplicată şi Ingineria Informaţiei |
| 1.4 Domeniul de studii | Calculatoare şi Tehnologia Informaţiei |
| 1.5 Ciclul de studii | Licenţă |
| 1.6 Programul de studii/Calificarea | Ingineria Informaţiei |

**2. Date despre disciplină**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2.1 Denumirea disciplinei | | | | Structuri de Date și Algoritmi (SDA) | | | |
| 2.2 Titularul activităţilor de curs | | | | Conf. Dr. Ing. Bogdan Ionescu | | | |
| 2.3 Titularul activităţilor de laborator | | | | Conf. Dr. Ing. Bogdan Ionescu  Dr. Ing. Ionuț Mironică | | | |
| 2.4 Anul de studiu | I | 2.5 Semestrul | 2 | 2.6 Tipul de evaluare | Verificare | 2.7 Regimul disciplinei | Obligatorie |

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3.1 Număr de ore pe săptămână din care | 3 | | 3.2 curs | 2 | 3.3 laborator | | 1 |
| 3.4 Total ore din planul de învăţământ din care | 42 | | 3.5 curs | 28 | 3.6 laborator | | 14 |
| Distribuţia fondului de timp | | | | | | | ore |
| Studiul după manual, suport de curs, bibliografie și notițe | | | | | | | 26 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate si pe teren | | | | | | | 20 |
| Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri | | | | | | | 10 |
| Tutoriat | | | | | | | 0 |
| Examinări | | | | | | | 6 |
| Alte activităţi | | | | | | | 0 |
| 3.7 Total ore studiu individual | | 62 | | | |  |  |
| 3.9 Total ore pe semestru | | 104 | | | |  |  |
| 3.10 Numărul de credite | | 4 | | | |  |  |

**4. Precondiţii (acolo unde este cazul)**

|  |  |
| --- | --- |
| 4.1 de curriculum | Programarea Calculatoarelor. |
| 4.2 de competenţe | Cunoștințe generale de limbajului C și de programarea calculatoarelor. |

**5. Condiţii (acolo unde este cazul)**

|  |  |
| --- | --- |
| 5.1 de desfăşurare a cursului | Nu este cazul. |
| 5.2 de desfăşurare a seminarului/laboratorului | Prezența obligatorie la lucrările de laborator (conform regulamentului studiilor universitare de licență în UPB). |

**6. Competenţe specifice acumulate**

|  |  |
| --- | --- |
| Competenţe profesionale | C1 Operarea cu fundamente ştiinţifice, inginereşti şi ale informaticii;  C2 Proiectarea componentelor hardware, software şi de comunicaţii;  C3 Soluţionarea problemelor folosind instrumentele ştiinţei şi ingineriei calculatoarelor;  C4 Utilizarea tehnologiilor şi mediilor de programare. |
| Competenţe transversale | - |

**7. Obiectivele disciplinei (reieşind din grila de competenţe specifice acumulate)**

|  |  |
| --- | --- |
| 7.1 Obiectivul general al disciplinei | *Curs:* însușirea mecanismelor de stocare, structurare și prelucrare a datelor cu alcătuire complexă. Studiul principiilor de bază utilizate în alcătuirea algoritmilor ca etapă esențială în dezvoltarea eficientă a aplicațiilor software. Criterii de proiectare eficientă a programelor. Studii de caz și metode de evaluare a performanțelor algoritmilor.  *Laborator:* însușirea practică, prin implementare de programe software, a noțiunilor predate la curs. Rezolvarea unor probleme practice concrete care includ elemente de structuri de date și algoritmi. |
| 4.2 Obiective specifice | Crearea abilităţilor de a aplica cunoştinţele generale privind stucturile de date și structurarea algoritmică a programelor. Posibilitatea de a evalua eficiența implementării unei aplicații software folosind criteriilor de performanţă analizate la curs. Dezvoltarea calităților necesare unui programator: abstractizare, analiză, și sinteză: combinare (abstracţie algebrică), echivalenţă (abstracţie topologică / sinteză), izolare (descompunere / analiză), evidenţiere (extragere din context), evidenţiere a unei stări (abstracţie temporală), idealizare (abstracţie de ordine). |

**8. Conţinuturi**

|  |  |  |  |
| --- | --- | --- | --- |
| 8.1 Curs | | Metode de predare | Observaţii |
| Pointeri şi structuri de date: lucrul cu variabile de tip pointer; alocarea dinamică a memoriei (utilizarea reprezentărilor dinamice vs. statice); lucrul cu structuri şi uniuni; structuri recursive; exemple, aplicaţii practice. | | Predarea se bazează pe folosirea videoproiectorului acoperind funcţia de comunicare şi demonstrativă. Metodele de comunicare orală utilizate sunt metoda expozitivă şi metoda problematizării utilizate frontal. Materialele de curs sunt: notele şi prezentările de curs, bibliografia prezentată la curs, platformele de laborator. Toate materialele sunt disponibile în format electronic, pe pagina web a cursului. | 4 ore |
| Liste de date: definirea şi lucrul cu liste de date; lista circulară; definirea şi lucrul cu stive; definirea şi lucrul cu cozi; exemple, aplicaţii practice. | | 6 ore |
| Arbori de date: generalităţi - abordarea unor tipuri de aplicaţii prin intermediul arborilor; definirea şi lucrul cu arbori binari; definirea şi lucrul cu arbori n-ari; arbori binari speciali (arbori plini, compleţi); exemple, aplicaţii practice. | | 6 ore |
| Algoritmi de sortare: metode de sortare prin interschimbare; metode de sortare prin inserţie şi selecţie; evaluarea eficienţei algoritmilor; exemple, aplicaţii practice. | | 6 ore |
| Studiul unor algoritmi clasici: algoritmi de tip Divide et Impera; algoritmi Greedy; exemple, aplicaţii practice. | | 6 ore |
| Bibliografie  - B. Ionescu, Note de curs Programarea Calculatoarelor, UPB, ETTI-EAII, LAPI – Laboratorul de Analiza și Prelucrarea Imaginilor, <http://imag.pub.ro/~bionescu/index_files/Page328.htm>;  - C++, <http://www.cplusplus.com> (ultima accesare 2015);  - Dumitru Iulian Năstac, Structuri de date şi algoritmi – Aplicaţii, Editura Printech, Bucureşti, 2008, ISBN 978-973-718-989-9. | | | |
| 8.2 Laborator | Metode de predare | | Observaţii |
| Pointeri şi Structuri de date. | Laboratorul se bazează pe programarea individuală în limbajul C la posturi de lucru dotate cu mediul de dezvoltare Dev-C++. Se solicită studenților studierea lucrării din îndrumar înainte de prezentarea în sala de laborator. | | 2 ore |
| Liste si stive de date. | 4 ore |
| Arbori de date. | 2 ore |
| Algoritmi de sortare. | 2 ore |
| Divide et impera | 2 ore |
| Colocviu final. | 2 ore |
| Bibliografie  - bibliografia de la curs. | | | |

**9. Coroborarea conţinuturilor disciplinei cu asteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatori reprezentativi din domeniul aferent programului**

|  |
| --- |
| Programa de curs oferă absolvenților capacitatea de a algoritmiza o problemă de calcul de complexitate medie, cunoaşterea unor algoritmi clasici de soluţionare a problemelor de calcul, capacitatea de a transforma o problemă de calcul exprimată în limbaj natural într-un limbaj de programare şi soluţionarea acesteia. Progresului tehnologic actual al dispozitivelor electronice și de telecomunicații este condiționat de abilitatea de dezvoltare și experimentare folosind limbaje de programare. Astfel că disciplina de programarea calculatoarelor este fundamentală în formarea viitoarelor generații de ingineri și cercetători din domeniu.  Programa asigură astfel absolvenţilor competenţe adecvate cu necesităţile calificărilor actuale și o pregătire ştiinţifică şi tehnică modernă, de calitate şi competitivă, care să le permită angajarea rapidă după absolvire. Aceasta este perfect încadrată în politica Universităţii Politehnica din Bucureşti, atât din punctul de vedere al conţinutului şi structurii, cât şi din punctul de vedere al aptitudinilor şi deschiderii internaţionale oferite studenţilor. Posibilii angajatori vizează atât mediul academic (profil didactic şi de cercetare) cât şi mediul industrial de cercetare-dezvoltare precum organizaţii/firme de orice dimensiune, de la cele mici create de studenţi/masteranzi (de exemplu start-up şi spin-off), până la cele multinaţionale. |

**10. Evaluare**

|  |  |  |  |
| --- | --- | --- | --- |
| Tip activitate | 10.1 Criterii de evaluare | 10.2 Metode de evaluare | 10.3 Pondere în nota finală |
| 10.4 Curs | - cunoaşterea noţiunilor teoretice fundamentale de structuri de date și algoritmi în limbajul C;  - algoritmizarea unor probleme de calcul folosind limbajul C; | Două teste scrise de verificare, de ponderi egale, în timpul semestrului, susţinute la date fixate la începutul cursului; subiectele acoperă întreaga materie, realizând o sinteză între parcurgerea teoretică comparativă a materiei şi explicitarea prin exerciţii şi probleme a modelelor de aplicaţie. | 60% |
| 10.5 Laborator | - participarea la sedințele de laborator; | - parcurgerea lucrărilor de laborator; | 10% |
| - rezolvarea unor probleme de calcul ce implică algoritmizare și folosirea structurilor de date complexe în limbajul C; | - evaluare practică, la calculator, pe parcurs și la finalul laboratorului. Colocviu final; | 30% |
| 10.6 Standard minim de performanţă | | | |
| * participarea la lucrările de laborator; * validarea laboratorului prin obținerea a minim 50% din punctajul aferent acestuia; * cumularea a minim 50% din punctajul aferent disciplinei (laborator și verificare). | | | |

Data completării Semnătura titularului de curs Semnătura titularului de aplicaţii

15.09.2015 Conf. Dr. Ing. B. Ionescu Conf. Dr. Ing. B. Ionescu

Dr. Ing. I. Mironică

Data avizării în catedră Semnătura directorului de departament

15.09.2015 Prof. Dr. Ing. S. Paşca