**COURSE DESCRIPTION**

**1. Program identification information**

|  |  |
| --- | --- |
| 1.1 Higher education institution | Politehnica University of Bucharest |
| 1.2 Faculty | Faculty of Electronics, Telecommunications and Information Technology |
| 1.3 Department | Dept. of Applied Electronics and Information Engineering |
| 1.4 Domain of studies | Computers and Information Technology |
| 1.5 Cycle of studies | Licence (engineering) |
| 1.6 Program of studies/Qualification | Information Engineering |

**2. Course identification information**

|  |  |
| --- | --- |
| 2.1 Name of the course | Data Structures and Algorithms (SDA) |
| 2.2 Lecturer | Assoc. Prof. Dr. Eng. Bogdan Ionescu |
| 2.3 Instructor for practical activities | Assoc. Prof. Dr. Eng. Bogdan Ionescu Dr. Eng. Ionuț Mironică |
| 2.4 Year of studies | I | 2.5 Semester | 2 | 2.6 Evaluation type | Verification | 2.7 Course choice type | Compulsory |

**3. Total estimated time** (hours per semester for academic activities)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3.1 Number of hours per week, out of which | 3 | 3.2 course | 2 | 3.3 practical activities | 1 |
| 3.4 Total hours in the curricula, out of which | 42 | 3.5 course | 28 | 3.6 practical activities | 14 |
| Distribution of time | hours |
| Study according to the manual, course support, bibliography and hand notes | 26 |
| Supplemental documentation (library, electronic access resources, in the field, etc) | 20 |
| Preparation for practical activities, homeworks, essays, portfolios, etc. | 10 |
| Tutoring | 0 |
| Examinations | 6 |
| Other activities | 0 |
| 3.7 Total hours of individual study | 62 |  |  |
| 3.9 Total hours per semester | 104 |  |  |
| 3. 10 Number of ECTS credit points | 4 |  |  |

**4. Prerequisites (if applicable)**

|  |  |
| --- | --- |
| 4.1 curricular | Computer Programming. |
| 4.2 competence-based | General knowledge of computer programming in C. |

**5. Requisites (if applicable)**

|  |  |
| --- | --- |
| 5.1 for running the course | Not applicable. |
| 5.2 for running of the applications | Compulsory presence at laboratory classes, according to current PUB regulations. |

**6. Specific competences**

|  |  |
| --- | --- |
| Professional competences | C1 Operating with scientific, engineering and computer science fundamentals;C2 Projection of hardware, software and communication components;C3 Solving engineering problems using computer science;C4 Use of programming environments and technologies. |
| Transversalcompetences | - |

**7. Course objectives (as implied by the grid of specific competences)**

|  |  |
| --- | --- |
| 7.1 General objective of the course | *Course:* understanding of data storage mechanisms and complex data structuring and processing. Study of algorithm fundamentals as a stepping stone towards efficient software development. Effective projection of algorithms. Case studies and methods for algorithm performance assessment.*Applications:* implementing the course notions in practical software applications. Solving diverse computational problems using data structures and algorithms. |
| 4.2 Specific objectives | Developing the abilities of employing general knowledge of data structures and algorithms for creating efficient programs. The possibility of assessing the effectiveness of a software application using standard algorithm performance assessment tools. Developing computer programming abilities such as: abstraction, analysis and synthesis, equivalence, isolation, emphasizing, idealization of the algorithmization problems.  |

**8. Content**

|  |  |  |
| --- | --- | --- |
| 8.1 Lectures | Teaching techniques | Remarks |
| Pointers and data structures: working with pointers, dynamic memory allocation, working with structures and unions, recurrence, examples and practical applications. | Teaching is carried out using video facilities. During classes, a permanent interaction between students and professor is maintained. Students are stimulated to develop solutions and discuss various programming exercises thus stimulating their creativity. Course materials consist of class notes, class bibliography and the platforms for the practical applications. All the materials are available to students on the course website.  | 4 hours |
| Lists: working with data lists, circular lists, working with stacks, working with queues, examples and practical applications. | 6 hours |
| Trees: fundamentals, working with binary trees, working with n-ary trees, specific binary trees (completeness, fullness), examples and practical applications. | 6 hours |
| Sorting: data sorting with interchanging and selection, assessing algorithm performance, examples and practical applications. | 6 hours |
| Classic algorithms: Divide et Impera, Greedy, examples and practical applications. | 6 hours |
| Bibliography- B. Ionescu, Class notes on Computer Programming, UPB, ETTI-EAII, LAPI – The Image Processing and Analysis Laboratory, <http://imag.pub.ro/~bionescu/index_files/Page328.htm>;- C++, <http://www.cplusplus.com> (last accessed 2015);- Dumitru Iulian Năstac, "Structuri de date şi algoritmi – Aplicaţii", Printech Publishing House, Bucharest, 2008, ISBN 978-973-718-989-9. |
| 8.2 Practical applications | Teaching techniques | Remarks |
| Pointers and Data structures. | The practical applications are carried out individually by each student. Each student has access to a fully equiped PC machine. Programming is carried out using the Dev-C++ environment. Students have to study the materials prior to each of the practical sessions. | 2 hours |
| Lists and stacks. | 4 hours |
| Data trees. | 2 hours |
| Sorting algorithms. | 2 hours |
| Divide et impera. | 2 hours |
| Final exam.  | 2 hours |
| Bibliography- course bibliography. |

**9. Bridging the course content with the expectations of the epistemic community representatives, professional associations and employers representatives for the domain of the program**

|  |
| --- |
| The course provides the students with the abilities of creating algorithms for medium complexity computational problems, knowledge of some classic algorithm solutions to computational problems, the ability of going from a general problem statement to a concrete software implementation. The current technological progress of electronic and telecommunication devices is conditioned by the ability of the engineers to develop and experiment new technologies with the help of computer programming. Therefore, computer programming plays a critical part in the training of the future generation of engineers that will foster new technologies in the field. |

**10. Evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| Type of activity | 10.1 Evaluation criteria | 10.2 Evaluation methods | 10.3 Weight in the final mark |
| 10.4 Lectures | - fundamental knowledge of data structures and algorithm implementation with C; - algorithmization of computational problems using C language; | Two written tests, of equal weight, during the semester, at pre-defined dates; the subjects cover the entire course material, being a synthesis between the comparative theoretical knowledge and the explicitation of the theory via problems and exercises. | 60% |
| 10.5 Practical applications | - attendance to the application sessions; | - finalizing each practical session tasks; | 10% |
| - solving various computational problems which involve algorithmization and use of complex data structures in C; | - practical evaluation during the semester and at the end of the laboratory applications; | 30% |
| 10.6 Minimal performance standard |
| * attendance to the practical application sessions;
* validating the applications by achieving at least 50% of the allotted points;
* accumulating in total at least 50 points from 100 (applications and final examination).
 |

Date Lecturer Instructor for practical activities

15.09.2015 Assoc.Prof. Dr. Eng. B. Ionescu Assoc.Prof. Dr. Eng. B. Ionescu

 Dr. Eng. I. Mironică

Date of department approval Director of Department,

15.09.2015 Prof. Dr. Eng. S. Paşca