

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	Electronic Technology and Reliability
1.4 Domain of studies	Electronic Engineering, Telecommunications and Information Technologies
1.5 Cycle of studies	Licence (engineering)
1.6 Program of studies/Qualification	Telecommunications Technologies and Systems

2. Course identification information

2.1 Name of the course				Data Structures and Algorithms (SDA)			
2.2 Lecturer				Assoc. prof. dr. eng. Dumitru-Iulian NASTAC			
2.3 Instructor for practical activities				Dr. eng. Virgil ILIAN			
2.4 Year of studies	1	2.5 Semester	2	2.6 Evaluation type	Verification	2.7 Course choice type	Mandatory

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					20
Supplemental documentation (library, electronic access resources, in the field, etc)					18
Preparation for practical activities, homeworks, essays, portfolios, etc.					14
Tutoring					
Examinations					10
Other activities					
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 UPB regulations.

6. Specific competences

Professional competences	C3. Applying the knowledge, concepts and methods concerning the computing systems architecture, microprocessors, microcontrollers, programming languages and computing techniques.
Transversal competences	CT1. Methodical analysis of the problems encountered in practical activity, by identifying the items for which there are dedicated solutions, ensuring the accomplishment of the professional tasks. CT3. Adapting to new technologies, professional and personal development through a continuous training by using documentation of printed sources, specialized software and electronic resources in at least one international language (English).

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

7.1 General objective of the course	Learn the principles of data acquisition, storage, and complex data structure processing. Study of the basic principles in software design as decisive phases on reliable applications. Efficient program designing criteria. Case studies and performance evaluation methods for the discussed algorithms.
7.2 Specific objectives	Acquiring programming skills (specific to the presented course) by implementing software applications and the learned data structures. Design of practical applications that involve data structures and various algorithms.

8. Content

8.1 Lectures	Teaching techniques	Remarks
1. Introduction 1.1. Data types: structures, unions, pointers, array 1.2. Programming paradigms 1.3. Structured programming 1.4. Recursion	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	4 hours
2. Basic notions concerning data structures 2.1. Static and Dynamic Representations 2.2. Specific Functions		2 hours
3. Linked lists and associated algorithms 3.1. Singly-linked lists. Doubly -linked lists. 3.2. Linearly-linked list. Circularly-linked list.		6 hours

3.3. Queue (FIFO) and Stacks (LIFO)	website. (www.euroqual.pub.ro/downloads)	
4. Tree structure and associated algorithms 4.1. Definitions. 4.2. Implementing of trees. 4.3. Binary tree. Iterating over binary trees. 4.4. Search. 4.5. Selection.		6 hours
5. Graph theory 5.1. Definitions. 5.2. Adjacency matrix. 5.3. Adjacency list. 5.4. Graph path searching.		4 hours
6. Other general algorithms 6.1. Sort Algorithms 6.2. Search Algorithms 6.3. Algorithms Efficiency		6 hours

Bibliography

- course notes and laboratory files (provided by course lecturer):

<http://www.euroqual.pub.ro/download/>

and on moodle (<http://electronica.curs.pub.ro/2016/course/view.php?id=8>)

- Bruce Eckel, *Thinking in C++*, 2nd edition, Prentice Hall, 2000, ISBN: 0139798099

- Kris Jamsa and Lars Klander, *Jamsa's C/C++ Programmer's Bible*, Publisher: Cengage Learning, US, 2010, ISBN: 1884133258

- Dumitru Iulian Năstac, *Structuri de date și algoritmi – Aplicații*, Editura Printech, București, 2008. ISBN 978-973-718-989-9.

- Dumitru Iulian Năstac, *Programarea calculatoarelor in limbajul C – Elemente fundamentale*, Editura Printech, București, 2006, ISBN 973-718-464-5

8.2 Practical applications	Teaching techniques	Remarks
Data structures	The practical applications are carried out individually by each student. Each student has access to a fully equipped 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
Linked lists		2 hours
Queues and Stacks Applications		2 hours
Trees Applications		2 hours
Search algorithm applications		2 hours
Evaluation of a project		2 hours
Laboratory test		2 hours

Bibliography

- laboratory files (provided by course lecturer): <http://www.euroqual.pub.ro/download/>

and on moodle (<http://electronica.curs.pub.ro/2016/course/view.php?id=8>)

- Bruce Eckel, *Thinking in C++*, 2nd edition, Prentice Hall, 2000, ISBN: 0139798099

- Kris Jamsa and Lars Klander, *Jamsa's C/C++ Programmer's Bible*, Publisher: Cengage Learning, US, 2010, ISBN: 1884133258

- Dumitru Iulian Năstac, “Structuri de date și algoritmi – Aplicații”, Editura Printech, București, 2008. ISBN 978-973-718-989-9.

- Dumitru Iulian Năstac, *Programarea calculatoarelor in limbajul C – Elemente fundamentale*, Editura Printech, București, 2006, ISBN 973-718-464-5

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

It is a big need for qualified engineers, specialized in computer programming and software applications, with a strong background in electronics, information technology and systems, to track the quick changes in software and hardware development.

The proposed curriculum fits exactly these modern advancement requirements, derived from the electronic engineering services required by the European economy. Considering the current technological progress in electronic devices, the envisaged fields are in fact open, starting with microelectronics, optoelectronics, telecommunications, industrial control (product evaluation), robotics (brain-machine interfaces), etc.

The students get then the adequate competences, as required by the current demanded professional skills, having a modern, competitive and high-level scientific and technical education. This allow them a fast integration in the labor market, as desired at University Politehnica of Bucharest, based not only on the content and the structure of the subject, but also on the gained skills and international opportunities offered after graduation.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Weight in the final mark
10.4 Lectures	- knowing the basic knowledge; - knowing to apply the theoretical knowledge to solve specific problems; - quantitative evaluation of the theoretical methods and techniques.	Verification at the end of the semester. Topics cover the entire course bibliography as well as the practical aspects of programming in C.	50 %
10.5 Practical applications	- practical knowing of the basic data structures and their processing algorithms to solve specific problems - knowing how to write, control and debug programs	Final exam at laboratory (oral and on computer), with particular emphasis on the practical component. Assessing a homework project, in order to estimate the practical skills.	50 %

