Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației

COURSE DESCRIPTION

1. Program identification information

1.1 Higher education institution	**National University of Science and Technology Politehnica Bucharest**
1.2 Faculty	**Electronics, Telecommunications and Information Technology**
1.3 Department	**Electronic Devices, Circuits and Architectures**
1.4 Domain of studies	Electronic Engineering, Telecommunications and Information Technology
1.5 Cycle of studies	Bachelor/Undergraduate
1.6 Programme of studies	Applied Electronics

2. Date despre disciplină

2.1 Course name (ro) (en)				Structuri de date și algoritmi Algorithms and Data Structures			
2.2 Course Lecturer				Conf. Dr. Radu HOBINCU			
2.3 Instructor for practical activities				Șl. Dr. Mihai ANTONESCU			
2.4 Year of studies	2	2.5 Semester	I	2.6. Evaluation type	E	2.7 Course regime	Ob
2.8 Course type		D	2.9 Course code	04.D.03.O.005		2.10 Tipul de notare	Nota

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

3.1 Number of hours per week	3.5	Out of which: 3.2 course	2.00	3.3 seminary/laboratory	1.5
3.4 Total hours in the curricula	49.00	Out of which: 3.5 course	28	3.6 seminary/laboratory	21
Distribution of time:					hours
Study according to the manual, course support, bibliography and hand notes Supplemental documentation (library, electronic access resources, in the field, etc) Preparation for practical activities, homework, essays, portfolios, etc.					48
Tutoring					3
Examinations					0
Other activities (if any):					0

3.7 Total hours of individual study	51.00
3.8 Total hours per semester	100
3.9 Number of ECTS credit points	4

4. Prerequisites (if applicable) (where applicable)

4.1 Curriculum	Computer Programming and Programming Languages 1, Computer Programming and Programming Languages 2

4.2 Results of learning	Logical, structured thinking. Knowledge of C and C++ languages syntax.

5. Necessary conditions for the optimal development of teaching activities (where applicable)

5.1 Course	Lecture hall with computer projector.
5.2 Seminary/ Laboratory/Project	Computer laboratory, running a Linux flavored operating system, with a C++ compiler (g++), code editor and development tools (a Git client, GNU Debugger and the Valgrind profiler), prefferably also a modern IDE (Jetbrains CLion). Lab capacity should be of minimum 15 seats and should have Internet access.

6. General objective *(Reffering to the teachers' intentions for students and to what the students will be thought during the course. It offers an idea on the position of course in the scientific domain, as well as the role it has for the study programme. The course topics, the justification of including the course in the currcula of the study programme, etc. will be described in a general manner)*

The general objective of the course is the study and assimilation of advanced analysis and implementation methods for complex software solutions. Specifically, the goal is to acquire knowledge regarding software implementation performance analysis in regards to computational complexity, as well as about algorithms (sorting, indexing, searching and parsing data) and data structures (sequences, sets, maps, binary search trees) which can be used for application optimization. The topic is critical in the context of developing high-performance software applications, in all specializations of the ETTI domain, whether related to telecommunications protocols, circuit simulations or embedded applications.

7. Competences *(Proven capacity to use knowledge, aptitudes and personal, social and/or methodological abilities in work or study situations and for personal and proffesional growth. They refflect the empolyers requirements.)*

Specific Competences	C3 Applying knowledge, concepts and basic methods to computer architecture, microcontrollers, and programming languages and techniques.
Transversal (General) Competences	CT1 Methodic analysis of the possible problems, identifying the already available solutions, when possible, to accomplish the professional obligations. CT3 Adaptation to new technologies and professional and personal development by long-life learning using printed documentation, specialized software, and electronic resources both in Romanian and in an international language.

8. Learning outcomes *(Synthetic descriptions for what a student will be capable of doing or showing at the completion of a course. The learning outcomes reflect the student's acomplishments and to a lesser extent the teachers' intentions. The learning outcomes inform the students of what is expected from them with respect to performance and to obtain the desired grades and ECTS points. They are defined in concise terms, using verbs similar to the examples below and indicate what will be required for evaluation. The learning outcomes will be formulated so that the correlation with the competences defined in section 7 is highlighted.)*

Knowledge	*The result of knowledge aquisition through learning. The knowledge represents the totality of facts, priciples, theories and practices for a given work or study field. They can be theoretical and/or factual.* Knowledge of C++ syntax and how to implement common data structures, as well as the pros and cons of using them in various scenarios.

Skills	*The capacity to apply the knowledge and use the know-how for completing tasks and solving problems. The skills are described as being cognitive (requiring the use of logical, intuitive and creative thinking) or practical (implying manual dexterity and the use of methods, materials, tools and intrumentation).* Skills related to analyzing a given problem in natural language and developing an own solution to solve the problem, as well as the ability to implement this solution in the C++ language, using common data structures and the STL library.
Responsability and autonomy	*The student's capacity to autonomously and responsably apply their knowledge and skills.* Ability to search for answers and online documentation related to the syntax and API of the C++ language, as well as to search for solutions to specific problems.

9. Teaching techniques *(Student centric techniques will be considered. The means for students to participate in defining their own study path, the identification of eventual fallbacks and the remedial measures that will be adopted in those cases will be described.)*

Theoretical concepts are presented using a tablet and stylus and projected on the white screen. At the end of the lecture, the notes are exported as PDF and published on the Moodle course page. After the theoretical presentation of the necessary notions, they are demonstrated with short code samples, written, compiled and executed in real time, the whole process being displayed on the projection screen. Students are constantly asked for feedback and to contribute to the implemented solutions. During the laboratory, students are given access to a computer and a set of exercises which they must solve with help from the lab assistant and the given documentation. They are graded depending on how many problems they solve until the end of the allocated time slot.

10. Contents

COURSE		
Chapter	**Content**	**No. hours**
1	Pointers and Object Oriented Programming – review	2
2	Definitions: Abstract and concrete data structures and algorithms, performance metrics	2
3	The sequence – array implementation	2
4	Iterators; Ordering – the comparator; Multi-criteria sorting; Binary search	4
5	Sequence implementations using linked lists and hybrid list-array structures	2
6	Double-ended queues, queues and stacks – vector, list and hybrid implementations	2
7	Trees; Sets; Implementing sets with Binary Search Trees	4
8	Hash functions; Implementing sets with hash functions	4
9	Implementing maps with hash functions and Binary Search Trees	2
10	Solving np-hard problems - backtracking	2
11	Lambda expressions; Using lambda expressions in STL functions	2
	Total:	28

Bibliography:

1. Radu HOBINCU, Călin BÎRĂ, Data Structures and Algorithms, electronic support files, https://archive.curs.upb.ro/2020/course/view.php?id=11682
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, ”Introduction to Algorithms, Fourth Edition”, MIT Press, ISBN: 9780262046305, 2022
3. https://en.cppreference.com/w/cpp - C++ API Documentation

LABORATORY

Crt. no.	Content	No. hours
1	Exercises with pointers (dynamic memory allocation, pointer arithmetic, persistent function parameter changes)	3
2	Excercises with classes and objects - OOP recap	3
3	Exercises with sequences implemented with vectors and singly and doubly linked lists	3
4	Exercises with queues and stacks implemented with vectors and singly and doubly linked lists.	3
5	Exercises with sets implemented with hash functions (Hash Set) and binary trees (Tree Set).	3
6	Exercises with maps implemented with trees and Hash functions (Tree Map and Hash Map).	3
7	Laboratory practical assignment	3
	Total:	

Bibliography:
Laboratory materials - https://wiki.dcae.pub.ro/index.php/Structuri_de_Date_%C8%99i_Algoritmi

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade
11.4 Course	Assimilation of the theoretical notions throughout the semester.	Homework assignments of medium to high complexity, which are automatically evaluated and graded by the Virtual Programming Lab plugin on the Moodle platform.	20
	Assimilation of the theoretical notions as a whole.	Final exam, composed of 50 questions, taken on the Moodle platform.	30

11.5 Seminary/laboratory/project	The ability to use theoretical concepts to solve programming assignments.	Programming assignments during each laboratory.	20
	The capacity to abstract a real problem and devise a software solution in the C programming language.	Laboratory assignment, automatically evaluated and graded by the Virtual Programming Lab plugin on the Moodle platform, consisting in a simple requirement, similar to the ones practiced in the lab, with the specific requirement for it to be implemented correctly and completely for the points to be awarded.	30
11.6 Passing conditions			
• Obtaining 50% of the total points. • Obtaining 50% of the final laboratory assignment points.			

12. Corroborate the content of the course with the expectations of representatives of employers and representative professional associations in the field of the program, as well as with the current state of knowledge in the scientific field approached and practices in higher education institutions in the European Higher Education Area (EHEA)

There 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 can gain adequate competences, as required by the current demanded professional skills, having a modern, competitive and high-level scientific and technical education. This allows for 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 skills gained and international opportunities offered after graduation.

Date	Course lecturer	Instructor(s) for practical activities
21.02.2025	Conf. Dr. Radu HOBINCU	Șl. Dr. Mihai ANTONESCU

Date of department approval — Head of department

Universitatea Națională de Știință și Tehnologie Politehnica București

Facultatea de Electronică, Telecomunicații și

Tehnologia Informației

Prof. Dr. Claudius DAN

Date of approval in the Faculty Council

Dean

Prof. Dr. Radu-Mihnea UDREA