

SYLABUS

Approved,
DEAN,
Prof.dr. Eng. Gilbert-Rainer GILLICH

1. Program data

1.1. Higher Education Institution	Babes Bolyai University Cluj-Napoca
1.2. Faculty	Faculty of Engineering
1.3. Department	Department of Engineering Sciences
1.4. Field of study	Electrical Engineering
1.5. Cycle of studies	License
1.6. Study program	Electromechanics

2. Discipline data

2.1. Discipline name	COMPUTER PROGRAMMING AND PROGRAMMING LANGUAGES I						
2.2. Holder of course activities	Ş.I.dr.ing. Cornelia-Victoria Anghel-Drugărin						
2.3.1. Holder of seminar activities	-						
2.3.2. Holder of laboratory activities	Ş.I.dr.ing. Cornelia-Victoria Anghel-Drugărin						
2.3.3. Holder of project activities	-						
2.4. Year of study	1	2.5. Semester	1	2.6. Type of assessment	E	2.7. Discipline regimen	Obl.

3. Estimated total time (hours per semester of teaching activities)

3.1. Number of hours per week	5	of which: 3.2. course	2	3.3. laboratory	2
3.4. Total hours in the curriculum	56	of which: 3.5. course	28	3.6. laboratory	28
Time fund distribution					69Hours
Study by textbook, course support, bibliography and notes					22
Additional documentation in the library, on specialized electronic platforms and in the field					20
Preparation of laboratories/ projects, homeworks, program lines, applications					25
Tutoring					-
Examination					2
Other activities:					YES
3.7. Total hours of self-study	69				
3.8. Total hours per semester	125				
3.9. Number of credits	5				

4. Preconditions (where applicable)

4.1. of curriculum	<ul style="list-style-type: none"> Informatics; TIC
4.2. of competencies	<ul style="list-style-type: none"> Digital competence

5. Conditions (where applicable)

5.1. of the course	<ul style="list-style-type: none"> PC/ laptop, video projector, smart board
5.2.1. of seminar development	<ul style="list-style-type: none"> -
5.2.2. of laboratory development	<ul style="list-style-type: none"> Computer with Internet network access, Install free kit used programming language: Codeblocks C++
5.2.3. of project development	<ul style="list-style-type: none"> -

6. Specific competences acquired

Professional skills	<ul style="list-style-type: none"> Proper application of knowledge of the fundamental disciplines mathematics, computer science, physics in the field of electrical engineering (CP 1). Proper use of fundamental concepts in computer science and information technology (CP 2). Proper use of fundamental concepts in the field of electrical engineering (CP 3).
Transversal competences	<ul style="list-style-type: none"> Establishing the objectives to be achieved, the conditions of implementation, the stages of work and the deadlines for completion. (CT 1). Efficient use of information sources, communication resources and assisted training (internet, specialized software applications, dedicated databases, on-line courses, etc.) in Romanian language and/or in an international language. (CT 4).

7. The objectives of the discipline (based on the grid of specific competences accumulated)

7.1. The general objective of the discipline	Acquiring the competence of computer programmer of future engineers.
7.2. Specific objectives: Objectives of knowledge (O.k.)	Knowledge of the data necessary to transpose engineering problems in the form of logic schemes, algorithms, pseudocode language, for finding out and displaying solutions. Elaboration of problem solving algorithms, the necessary steps to be taken to find the solution, by programming in C++, of the technical and engineering problems. Familiarization with the use of the facilities of a programming environment and of some programs for calculating and solving specific problems in the field of engineering, implementing them in source codes of DevC++ programs.
Objectives of attitude (O.at)	Developing a positive attitude towards individual work and teamwork. Developing the attitude of assuming and empowering the role of self-evaluation and collegial evaluation.
Objectives of empowerment (O.em)	Development of practical C++ programming applications, in correlation with the requirements of employers on the labor market.

8. Content

8.1. Course	Teaching methods	Observations
1. Introductory notions regarding the structure and basic components of a personal computer (terminology, definitions, concepts). The basic structure of a computer, in general. Hardware and software architectures, basic components of the computer. Input-output equipment and peripherals. Operating system basics and computer startup program configuration (BIOS). Internal representation of data. Bases of numbering. Data and informations. Directories and Files.	Initial assessment of students' previous knowledge! Presentation of the content of the discipline sheet and bibliography . Lecture, lecture, explanations, interactive discussions and dialogue with students	2 hours
2. Stages of making a program. General notions and definition of algorithms for solving technical problems. Properties of algorithms. Mathematical operations used in algorithms. Basic rules and principles in operations with algorithms. Types of algorithms. Algorithmic programming languages. Flowcharts. Pseudocode language.	Lecture, presentation/explanations, discussions and dialogue with students. Practical examples. Compiling a summary journal.	2 hours
3. Programming languages: overview, specific notions, set of actions, data set, action sequence. Presentation of the C programming language and comparison with the C++ programming language. Introduction to the C++ programming language: basic elements of the language, C++ vocabulary, identifiers, keywords, separators, comments, operations and operators. Examples of C vs.C++ program source code lines	Lecture, explanations, discussion and dialogue with students. We have proposed/foreseen various ways to come into contact with information (inductive teaching methods: problem-based learning). Edifying examples of program code lines	2 hours
4. Standard data types in C/C++ programming languages. Constants and variables C++. Equivalences between data types in C/C++ programming languages. Conversions of data types. Explicit conversions. Local variables. Global variables. Examples of conversions of data types. Examples by C++ programs.		2 hours
5. Linear programming. Data flows in the C++ programming language. The flow of input data (cin). C++ program source code examples Output data flow (cout). C++ program source code examples System functions cin>> and cout<< C++ programming language specific instructions: empty instruction, expression instruction, simple, compound instruction. Code C++ program examples	Exposition, problematization, heuristic conversation, detailed explanation, students' challenge to dialogue, discussions. Brainstorming Method	2 hours
6. C++ language control instructions. Instruction syntax Decision-making instruction (conditional) if, simple version. Compound (conditional) decision-making instruction: if - else , variants 1 and 2 <i>Conditional decision instructions switch- cases.</i> Syntax and effects of the switch instruction. Examples of C++ program source code, with decision structures.	We have provided situations in which the student can reflect on how heuristic discussions have learned, fixing the acquired knowledge	2 hours

7. Repetitive instructions in the C++ programming language. Repetitive instruction for . The syntax of the for instruction. Execution of the for instruction, repetition with a known number of steps. Examples of source code of C++ programs, using repetitive control statements.	Lecture, dialogue, discussions, fixing the acquired notions Summary journals. Examples of program source code instruction lines in the DevC++ programming language	2 hours
8. Repetitive control instructions in the C++ programming language. Repetitive instruction while syntax of the instruction while . Loop instruction do while . Syntax of the do while statement. Examples of source code of C++ programs, using repetitive control instructions.		2 hours
9. Break instructions: break, continue, goto, return, label. Examples of C++ interrupting applicative programs.	Exposure problematization, heuristic conversation, detailed explanation, challenge of students to dialogue, discussions, fixation of acquired notions.	2 hours
10. Functions in C++ programming. Prototype of functions. Call of functions. Return results. Lists and operations with lists: sorting, recursion, etc.	Exposition, problematization, heuristic conversation, detailed explanation, challenge of students to dialogue, discussions	2 hours
11. Structured programming. Strings. Functions for manipulating strings. Single-dimensional, multidimensional paintings . C++ programs.	We have proposed/foreseen tasks related to the future profession, relevant examples from professional practice	2 hours
12. Pointers and references in C++. Definition and operation with pointers. Pointer arithmetic. Sending parameters to functions. Pointers and functions. Generic functions. C++ programs.		2 hours
13. Pointer arrays. Use dynamic arrays. Dynamic memory allocation. Pointers and paintings. C++ programs	Exposure, problematization, heuristic conversation. Program debugging	2 hours
14. Text files. Binary files. File processing functions. Send arguments to the execution of programs. Examples of programs.	Lecture, explanation, recap, summary journals. We have foreseen situations in which the student can reflect on how he has learned and acquired programming knowledge.	2 hours
Bibliography 1. Anghel Drugărin Cornelia Victoria (2018) - <i>Programarea calculatoarelor si limbaje de programare</i> Editura EFTIMIE MURGU, Reșița, 122 pagini, ISBN 978-606-631-074-1 2. Anghel Drugărin Cornelia-Victoria, Terfăloagă Irina Maria , Programarea calculatoarelor si limbaje de programare C/C++. Teorie și aplicații, Volumul1, CD, Editura Eftimie Murgu Resita, 2015; 3. Bancila M , Modern C++ Programming, Ed. Cluj-Napoca, 2020 4. Alexandrescu , Programare moderna in C++. Programare generica si modele de proiectare aplicate, Editura Teora, 2002 5. E. Horowitz, S. Sahni, D. Mehta , Fundamentals of Data Structures in C++, Computer Science Press, Oxford, 1995. 6. Dr. Kris Jamsa, Lars Klander , „ <i>Totul despre C și C++ - Manualul fundamental de programare în C și C++</i> ”, EdTeora, 1999. 7. Negrescu I. <i>Limbajele C și C++ pentru începători</i> . Ed. albastră, grupul microinformatica, 2001 8. https://world-it.ro/software/limbaje-de-programare/c/introducere-c 9. https://www.youtube.com/watch?v=JjY9U9lu37I tutorial C++ 10. https://www.youtube.com/watch?v=ZKH9r0b9f9Q C++ 11. https://world-it.ro/cursuri-it-gratuite/curs-c-exemple-c++		
8.2.2. Laboratory	Teaching methods	Observations
1. Presentation of windows operating system and workstations in the Computer Lab. Presentation of the laboratory file. PSI training rules. Data and information. Directories and Files.	We have provided an activity (evaluation / solving of exercises, etc.)	2 hours

2. Bases of numbering, transformations. Algorithms. Representation of algorithms. Algorithmic program languages. Logical schemes, pseudocode. Examples		2 hours
3. Programs applied to data types, operators, and expressions in the C++ language.		2 hours
4. Programs applied to types of constants and variables in the C++ language.		2 hours
5. Linear programs, with write / read functions in C++ language.		2 hours
6. Programs applied to decision instructions: if-then-else, switch-case		2 hours
7. Applied programs to repetitive instructions: for		2 hours
8. Programs applied to looping instructions: do while, while do, in C++ language. Evaluation along the way.		2 hours
9. Applicative programs with C++ interruptions.		2 hours
10. Applicative programs with functions. Recursiveness of C++		2 hours
11. Programs with vectors and arrays in C++.		2 hours
12. Applicative programs with pointers in C++. Pointers to functions		2 hours
13. Problems and structured programming. Union, enumeration.		2 hours
14. Text files. Final discussions. End of laboratory work, teaching homework.		2 hours

Bibliography

1. **Anghel Drugărin Cornelia Victoria** (2018) - *Programarea calculatoarelor si limbaje de programare* Editura EFTIMIE MURGU, Reșița, 122 pagini, ISBN 978-606-631-074-1
2. **Anghel Drugărin Cornelia Victoria**, Terfaloaga Irina Maria (2016), *Programarea calculatoarelor si limbaje de programare C/C++*. Vol2, Reșița: Editura Eftimie Murgu, ISBN 978-606-631-054-3.
3. **Anghel Drugărin Cornelia Victoria**, Terfaloaga Irina Maria (2015)– “*Programarea calculatoarelor si limbaje de programare C/C++*”, Vol1, CD Editura EFTIMIE MURGU, 170 pagini, Reșița;
4. Logofătu Doina – *Bazele programarii in C. Aplicatii*. Ed. Polirom, Iasi, **2006**
5. **Alexandru**, Andrei *Programarea moderna in C++*. *Programare generica si modele de proiectare aplicate*. Ed. Teora, Bucuresti 2002.
6. https://info64.ro/Intro_functii-operatori_referinte/
7. <https://www.youtube.com/watch?v=JjY9U9lu37I> tutorial C++
8. <https://www.cprogramming.com/>
9. <https://www.youtube.com/watch?v=ZKH9r0b9f9Q>

9. Corroborating the contents of the discipline with the expectations of the representatives of the epistemic community, professional associations and representative employers in the field related to the program

The content of the discipline was established, after consultation with the main employers in the field of engineering, the discussions with the committee for drawing up and reviewing the curriculum, in force, as well as with the colleagues who teach related subjects.

Together with the students, we participate in external visits and / or teaching trips, at specialized companies in the county and region, in events such as: computer science circle, projects, round tables, workshops, Summer School, national and even international student scientific symposiums, which are related to the field of training.

We have proposed/foreseen tasks related to the future profession, relevant examples from professional practice.

10. Evaluate

Activity type	10.1. Assessment criteria	10.2. Valuation methods	10.3. Weighting of the final grade
10.4. Course	Participation in debates	Număr de intervenții	10 %
	Level of knowledge gained	Grid test + C++ program	60 %
10.5.1. Seminar	Involvement in activities		
	Level of competences acquired		
10.5.2. Laboratory	Involvement in activities	Number of interventions Accomplished themes	10 %
	Level of practical competences acquired	Practical test, programming applications, portfolio	20 %
10.5.3. Proiect	Promptitudinea parcurgerii etapelor din proiect		
	Calitatea proiectului		
10.6. Minimum performance standard			
<p>We have provided evaluations (formative) along the way (one in the course and two evaluation papers at the laboratory) in order to be able to discuss based on them a self-directed learning plan</p> <ul style="list-style-type: none"> • Promotion of laboratory applied activities with a minimum grade of 5 (five); • Brief treatment of the subjects, the exam being considered to be passed, if the final grade calculated is a minimum of 5(five). 			

Date of completion

05.05.2022

Signature of the course holder

**Ș.I.dr.ing. Cornelia-Victoria
ANGHEL-DRUGĂRIN**

Signature of the ~~seminar~~ / laboratory holder

**Ș.I.dr.ing. Cornelia-Victoria
ANGHEL-DRUGĂRIN**

Date of approval in the department

.....

Signature of the DSI department director

Ș.I.dr. fiz. Hațiegan Cornel