

SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	Babeş-Bolyai University
1.2 Faculty	Faculty of Mathematics and Computer Science
1.3 Department	Department of Computer Science
1.4 Field of study	Computers and Information Technology
1.5 Study cycle	Bachelor
1.6 Study programme / Qualification	Information Engineering

2. Information regarding the discipline

2.1 Name of the discipline (en) (ro)	C Programming						
2.2 Course coordinator	Lect. Dr. Ing. Horea Adrian Grebla						
2.3 Seminar coordinator	Lect. Dr. Ing. Horea Adrian Grebla						
2.4. Year of study		2.5 Semester		2.6. Type of evaluation	C	2.7 Type of discipline	DD Facultative
2.8 Code of the discipline	MLR5076						

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	3	Of which: 3.2 course	1	3.3 seminar/laboratory	2 LP
3.4 Total hours in the curriculum	42	Of which: 3.5 course	14	3.6 seminar/laboratory	28
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					10
Additional documentation (in libraries, on electronic platforms, field documentation)					10
Preparation for seminars/labs, homework, papers, portfolios and essays					10
Tutorship					10
Evaluations					7
Other activities:					0
3.7 Total individual study hours	47				
3.8 Total hours per semester	75				
3.9 Number of ECTS credits	3				

4. Prerequisites (if necessary)

4.1. curriculum	•
4.2. competencies	•

5. Conditions (if necessary)

5.1. for the course	• Multimedia projector
5.2. for the seminar /lab activities	• Laboratory with caomputers having Linux and Windows operating systems

6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none"> • C3.1 Identifying classes of problems and solving methods that are specific to computing systems • C4.1 Identifying and describing technologies, programming environments and various concepts that are specific to programming engineering • C4.5 Developing, implementing and integrating software solutions
Transversal competencies	<ul style="list-style-type: none"> • CT1 Honorable, responsible, ethical behavior, in the spirit of the law, to ensure the professional reputation • CT3 Demonstrating initiative and pro-active behavior for updating professional, economical and organizational culture knowledge

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	<ul style="list-style-type: none"> • Acquiring the main entities and concepts with which it operates in the standard C language. • Mastering the basics of specific programming and using functions
7.2 Specific objective of the discipline	<ul style="list-style-type: none"> • Operators and expressions. • Control structures. • Data structures • IO operations with files. • Pointers. • Defining user functions. • Memory classes and C preprocessor.

8. Content

8.1 Course	Teaching methods	Remarks
Week 1 Introduction. <ul style="list-style-type: none"> • data types • operators • expressions • instructions 	Presentation: description, explanations, practical examples, demonstrations,	

<ul style="list-style-type: none"> • IO operations with standard files 	discussions on case studies.	
<p>Week 3 Control structures.</p> <ul style="list-style-type: none"> • if • switch • for • while • break, continue <p>Week 5. Data structures.</p> <ul style="list-style-type: none"> • arrays • structures • union • bit fields • access to command line arguments 	Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.	
<p>Week 7 Standard IO Library C; access to files.</p> <ul style="list-style-type: none"> • fopen, fclose • fgets, fprintf, fscanf • fread, fwrite, fseek 	Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.	
<p>Week 9 Pointers.</p> <ul style="list-style-type: none"> • defining and operating with pointers • malloc and free functions; dynamic variables. • pointer arithmetic • equivalence between arrays and pointers • use of dynamic arrays 	Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.	
<p>Week 11 Defining user functions</p> <ul style="list-style-type: none"> • prototype functions • transmission of parameters • returned values • recursion 	Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.	
<p>Week. 13-14. Memory classes and the preprocessor C</p> <ul style="list-style-type: none"> • automatic, external, static memory classes • C preprocessor: #include, #define, #if, typedef 	Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.	

Bibliography

1. BOIAN F, VANCEA A. IURIAN S Limbajul C, culegere de probleme. Lito UBB, 1992

2. COSTEA D. Inițiere în limbajul C. Ed. Teora, 1997.
3. KALICHARAN N. C By Example. Cambridge University Press, 1996
4. KERNINGHAN B.W., RITCHIE D.M The C Procraming Language -- Ansi C. Prentice Hall, 1988.
5. NEGRESCU L. Limbajele C și C++ pentru începători. Ecd. Albastră, Grupul Microinformatica, 2001
6. C Language Tutorial http://www.physics.drexel.edu/courses/Comp_Phys/General/C_basics/
7. C Tutorial <http://www.cprogramming.com/tutorial/c-tutorial.html>

8.2 Seminar / laboratory	Teaching methods	Remarks
<p>Each course has two laboratories affected. In the within these, illustrative demonstrations are made first and relevant examples on the course topic. Then the students, alone or in teams, solve problems received as homework in the laboratory or as homework.</p>	<p>Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.</p>	
<p>Week 1 Introduction.</p> <ul style="list-style-type: none"> • data types • operators • expressions • instructions • IO operations with standard files 	<p>Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.</p>	
<p>Week 3-4 Control structures.</p> <ul style="list-style-type: none"> • if • switch • for • while • break, continue 	<p>Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.</p>	
<p>Week 5-6. Data structures.</p> <ul style="list-style-type: none"> • paintings • structures • union • tighten • bit fields • access to command line arguments 	<p>Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.</p>	
<p>Week 7 Standard IO Library C; access to files.</p> <ul style="list-style-type: none"> • fopen, fclose • fgets, fprintf, fscanf • fread, fwrite, fseek 	<p>Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.</p>	

<p>Week 9 Pointers.</p> <ul style="list-style-type: none"> • defining and operating with pointers • malloc and free functions; dynamic variables. • pointer arithmetic • equivalence between pointers and pointers • use of dynamic pointers 	<p>Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.</p>	
<p>Week 11 Defining user functions</p> <ul style="list-style-type: none"> • prototype functions • transmission of parameters • returned values • recursion 	<p>Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.</p>	
<p>Week 13-14. Memory classes and the preprocessor C</p> <ul style="list-style-type: none"> • automatic, external, static memory classes • C preprocessor: #include, #define, #if, typedef 	<p>Presentation: description, explanations, practical examples, demonstrations, discussions on case studies.</p>	

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

<ul style="list-style-type: none"> • By mastering the theoretical-methodological concepts and approaching the practical aspects included in the discipline Operating Systems, students acquire a consistent knowledge, in accordance with the partial competencies required for possible occupations provided in Grid 1 - RNCIS • The course complies with the IEEE and ACM Curriculum Recommendations for Computer Science studies. • The course exists in the curriculum of Romanian universities and faculties • The content of the course is very well appreciated by the software companies that have as graduate employees of this course

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Mastery and correct understanding of the issues covered in the course	Last week exam	50%
	Correct problem solving		
10.5 Seminar/lab activities	Ability to solve practical		50%

	problems specific to the course, directly on the computer and in a limited time Activity in the laboratory		
10.6 Minimum performance standards			
<input type="checkbox"/>			

Date

19.05.2022

Signature of course coordinator

lect. Dr. Ing. Horea Adrian Grebla



Signature of seminar coordinator

lect. Dr. Ing. Horea Adrian Grebla



Date of approval

24.05.2022

Signature of the head of department

Prof. dr. Laura Dioşan

