

SYLLABUS

PROGRAMMING BASICS 2024-2025

1. Program General Data

1.1. University	„1 Decembrie 1918” of Alba Iulia
1.2. Faculty	Faculty of Informatics and Engineering
1.3. Department	Informatics, Mathematics and Electronics Department
1.4. Area	Computer Science
1.5. Level	Undergraduate
1.6. Specialization	Computer Science ESCO-08: 2512/ Software developers Analyst 251201 Computer System Programmer 251204 Computer System Engineer 251203

2. Subject General Data

2.1. Subject	<i>PROGRAMMING BASICS</i>		2.2. Code	CSE 103			
2.3. Course holder/ Lecturer/ Instructor's Name	Domşa Ovidiu						
2.4. Teaching Assistant's Name	Incze Arpad						
2.5. Year	I	2.6. Semester	I	2.7. Evaluation form (E – final exam/C-examination /VP)	E	2.8. Status (C– Compulsory, Op – optional, F - Facultative)	C

3. Course Structure (Weekly number of hours)

3.1. Weekly number of hours	5	3.2. course	2	3.3. seminar, laboratory	3
3.4. Total number of hours according to the curricula	70	3.5. course	28	3.6. seminar, laboratory	42
Time distribution:					Hours
Individual study using the lecture notes					30
Documentation (library)					20
Homework, Essays, Portfolios					30
Tutoring					-
Evaluation (exams)					25
Other activities.....					-

3.7 Total number of hours for individual study	105
3.8 Total number of hours according to the curricula	70
3.9 Total number of hours per semester	175
3.10 Credits	7

4. Prerequisites

4.1. Curricula prerequisites	
4.2. according to the general competencies	Mathematics at the medium level.

5. Conditions

5.1. Conditions to support teaching	<i>Room equipped with video projector/board.</i>
5.2. Conditions for supporting seminar/laboratory activities	<i>Laboratory – computers. Software: Code Blocks, Internet acces.</i>

6. Specific competences to be acquired (chosen by the course leader from the programme general competences grid)

Professional competences	<i>CP 7 (1 ECTS), CP10 (1 ECTS), CP13 (1 ECTS), CP24 (1 ECTS), CP27 (1 ECTS), CP 29 (1 ECTS), CP33 (1 ECTS)</i>
Transversal competences	Not applicable

7. Course objectives

6.1 General course objectives	<ul style="list-style-type: none"> - <i>Develop algorithmic thinking and skills for developing elementary algorithms.</i> - <i>Learning basic tools for developing elementary algorithms.</i> - <i>Knowledge of types of methods and data structure regarding algorithms and their development methods.</i> - <i>Use of an advanced programming language for implementing the studied algorithms.</i> - <i>Programming in C language knowledge.</i>
6.2 Specific course objectives	

8. Course contents

Lectures	Didactic methods used	Observation
General principles for structured programming and algorithm development.	<i>Lecture, discussions, examples</i>	
Definitions: Algorithms. Characteristics. Structure, data and algorithmically steps.	<i>Lecture, discussions, examples</i>	
Organizing data and structure in structural programming. Linear, alternative and while structures.	<i>Lecture, discussions, examples</i>	
Algorithms: Elaboration, Correctness, Complexity and Testing.	<i>Lecture, discussions, examples</i>	
Elementary algorithms. Switch variable values, alternative structures, While and repeat structures, vectors, mathematical quantification each/exist, Cartesian product algorithm.	<i>Lecture, discussions, examples</i>	
Counting, Summary, Searching elementary algorithms.	<i>Lecture, discussions, examples</i>	
Evaluation	<i>Lecture, discussions, examples</i>	
Sub algorithms, defining parameters and variable transfer	<i>Lecture, discussions, examples</i>	
Elementary sorting methods (Bubble sort, Selection Sort, Numbering Sort, Insertion Sort)	<i>Lecture, discussions, examples</i>	

Sorting and searching algorithms complexity. Elementary algorithm methods. Intercalation.	<i>Lecture, discussions, examples</i>	
Recursively algorithms. Recursively function.	<i>Lecture, discussions, examples</i>	
C language. Elementary concepts. Vocabulary. Data definition. Input/output data in C.	<i>Lecture, discussions, examples</i>	
Programming structure in C. Instructions IF, WHILE, DO, CASE	<i>Lecture, discussions, examples</i>	
Elementary algorithms, applications.	<i>Lecture, discussions, examples</i>	
References		
<ol style="list-style-type: none"> 1. Ovidiu Domsa, Imperative / Procedural programming, Course notes, 2013. 2. Cormen T.H., Leiserson E.C., Rivest R.R., Introduction in algorithms, MIT Press, 2001. 3. Dahl O.J., Dijkstra E.W., Hoare C.A.R., Structured Programing, Academic Press, 1972. 4. Donald E. Knuth, The Art of Computer Programming, Volumes 1–3, Addison-Wesley Professional Volume 1: Fundamental Algorithms (3rd edition), 1997. Addison-Wesley Professional, Volume 2: Seminumerical Algorithms (3rd Edition), 1997. Addison-Wesley Professional, Volume 3: Sorting and Searching (2nd Edition), 1998. Addison-Wesley Professional. 		
Seminars-laboratories	Didactic methods used	
General principles for structured programming and algorithm development. Examples.	<i>laboratory works</i>	
Describe algorithms using logical schema flow, pseudo code language and programming languages.	<i>laboratory works</i>	
C language. IDE Code Blocks. C programs structure. C elementary programs examples.	<i>laboratory works</i>	
Elementary C language data and structure representation. Input/ Output data in C.	<i>laboratory works</i>	
Structural programming in C.	<i>laboratory works</i>	
Elementary Vectors program examples.	<i>laboratory works</i>	
Elementary String program examples.	<i>laboratory works</i>	
Elementary algorithm implementation. Switch, Search, Counting, numbering, Sum	<i>laboratory works</i>	
Sorting algorithms. (Bubble sort, Selection Sort, Numbering Sort, Insertion Sort)	<i>laboratory works</i>	
Application. Sorting and searching algorithms complexity. Elementary algorithm methods. Intercalation.	<i>laboratory works</i>	
Recursively algorithms. Recursively function.Examples.	<i>laboratory works</i>	
Application.	<i>laboratory works</i>	
Evaluation. <i>Portfolio of laboratory practical works</i>	<i>laboratory works</i>	
References		
<ol style="list-style-type: none"> 1. Ovidiu Domsa, Imperative / Procedural programming, Course notes, 2013. 2. Cormen T.H., Leiserson E.C., Rivest R.R., Introduction in algorithms, MIT Press, 2001. 3. Dahl O.J., Dijkstra E.W., Hoare C.A.R., Structured Programing, Academic Press, 1972. 4. Donald E. Knuth, The Art of Computer Programming, Volumes 1–3, Addison-Wesley Professional Volume 1: Fundamental Algorithms (3rd edition), 1997. Addison-Wesley Professional, Volume 2: Seminumerical Algorithms (3rd Edition), 1997. Addison-Wesley Professional, Volume 3: Sorting and Searching (2nd Edition), 1998. Addison-Wesley Professional. 		

9. Corroborating Course content expectations to the epistemic community representatives, professional associations and employers representative for the curricula

- *Not applicable.*

10. Assessment

Activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage from the final mark
10.4 Course	<i>Final evaluation</i>	<i>Written exam</i>	50%
	-	-	-
10.5 Seminar/laboratory	<i>Continuous assessment</i>	<i>Portfolio of laboratory practical works</i>	50%
	-	-	-
10.6 Minimum performance standard:			

Completion date

Instructor's signature

Teaching assistant's signature

.....

.....

Date of approval within the department

Head of department's signature

.....

.....