SYLLABUS

IMPERATIVE / PROCEDURAL PROGRAMMING
2022-2023

1. Program General Data

1.1. University	**„1 Decembrie 1918”**
1.2. Faculty	**Facultatea de Informatică și Inginerie**
1.3. Department	**Informatică și Inginerie**
1.4. Area	**Computer Science**
1.5. Level	undergraduate
1.6. Specialization	**Computer Science**

2. Subject General Data

2.1. **Subject**	*Programming_basics*			2.2. **Code**		CSE103	
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)	**O**

3. Course Structure (Weekly number of hours)

3.1. Weekly number of hours	**4**	3.2. course	***2***	3.3. seminar, laboratory	***2***
3.4. Total number of hours according to the curricula	**56**	3.5. course	***28***	3.6. seminar, laboratory	***28***
Time distribution:					Hours
Individual study using the lecture notes					***50***
Documentation (library)					***50***
Homework, Essays, Portfolios					***50***
Tutoring					**-**
Evaluation (exams)					***28***
Other activities…….					**-**

3.7 Total number of hours for individual study	**108**
3.8 Total number of hours according to the curricula	**70**
3.9 Total number of hours per semester	**178**
3.10 Credits	**5**

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	*Room equipped with video projector/board.*
5.2. Conditions for supporting seminar/laboratory activities	*Laboratory – computers. Software: BorlandC, Internet acces.*

6. Competenţe specifice acumulate (cele alese de titular din grila de competente)

Professional competences	- Development of skills required to solve problems using the fundamentals structure in algorithms. - Identify the addressed problems with the studied techniques and algorithms. -The student will be able to translate in algorithmic language (pseudo code, programming language) the solution of the elementary date structures and problems. - Thoroughly study of elementary data structures and algorithms concepts and the elementary methods used for handling them (simple data, tables, strings,).
Transversal competences	**Cognitive skills:** acquisition of basic and specific knowledge about the concept of elementary algorithm; the ability to identify the applicability of the studied algorithms in real problems; understanding the need of using elementary methods to create algorithms when addressing problems from an algorithmic perspective; acquiring basic knowledge on the concept of algorithms complexity. **Affective skills**: develop the capacity of analysis and understanding of an algorithmically problems and effectively address it from an algorithmic perspective. Team spirit: encouraging students to work in design, analysis and programming teams. Awareness of the importance of the knowledge and thoroughly study of elementary algorithms.

7. Course objectives

6.1 General course objectives	- *Develop algorithmic thinking and skills for developing* elementary *algorithms.* - *Learning basic tools for developing* elementary *algorithms.* - *Knowledge of types of methods and data structure regarding algorithms and their development methods.* - *Use of an advanced programming language for implementing the studied algorithms.* - *Programming in C language knowledge.*
6.2 Specific course objectives	

8. Course contents

Lectures	**Didactic methods used**	**Observation**
General principles for structured programming and algorithm development.	***Lecture, discussions, examples***	
Definitions: Algorithms. Characteristics. Structure, data and algorithmically steps.	***Lecture, discussions, examples***	
Organizing data and structure in structural programming. Linear, alternative and while structures.	***Lecture, discussions, examples***	
Algorithms: Elaboration, Correctness, Complexity and Testing.	***Lecture, discussions, examples***	

Elementary algorithms. Switch variable values, alternative structures, While and repeat structures, vectors, mathematical quantification each/exist, Cartesian product algorithm.	***Lecture, discussions, examples***	
Counting, Summary, Searching elementary algortitms.	***Lecture, discussions, examples***	
Evaluation	***Lecture, discussions, examples***	**On-line, Teams**
Sub algorithms, defining parameters and variable transfer	***Lecture, discussions, examples***	**On-line, Teams**
Elementary sorting methods (Bubble sort, Selection Sort, Numbering Sort, Insertion Sort)	***Lecture, discussions, examples***	**On-line, Teams**
Sorting and searching algorithms complexity. Elementary algorithm methods. Intercalation.	***Lecture, discussions, examples***	**On-line, Teams**
Recursively algorithms. Recursively function.	***Lecture, discussions, examples***	**On-line, Teams**
C language. Elementary concepts. Vocabulary. Data definition. Input/output data in C.	***Lecture, discussions, examples***	
Programming structure in C. Instructions IF, WHILE, DO, CASE	***Lecture, discussions, examples***	
Elementary algorithms, applications.	***Lecture, discussions, examples***	

References

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.	***laboratory works***	
Describe algorithms using logical schema flow, pseudo code language and programming languages.	***laboratory works***	
C language. IDE Code Blocks. C programs structure. C elementary programs examples.	***laboratory works***	
Elementary C language data and structure representation. Input/ Output data in C.	***laboratory works***	
Structural programming in C.	***laboratory works***	**On-line, Teams**
Elementary Vectors program examples.	***laboratory works***	**On-line, Teams**
Elementary String program examples.	***laboratory works***	**On-line, Teams**
Elementary algorithm implementation. Switch, Search, Counting, numbering, Sum	***laboratory works***	**On-line, Teams**
Sorting algorithms. (Bubble sort, Selection Sort,	***laboratory works***	**On-line,**

Numbering Sort, Insertion Sort)		**Teams**
Application. Sorting and searching algorithms complexity. Elementary algorithm methods. Intercalation.	***laboratory works***	
Recursively algorithms. Recursively function.Examples.	***laboratory works***	
Application.	***laboratory works***	
Evaluation. *Portfolio of laboratory practical works*	***laboratory works***	

References

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	*Final evaluation*	*Written exam*	50%
	-	-	-
10.5 Seminar/laboratory	*Continuous assessment*	*Portfolio of laboratory practical works*	50%
	-		-
10.6 Minimum performance standard:			

Completion date Instructor’s signature Teaching assistant’s signature

23.09.2022 ……………..………… ………………………….

Date of approval within the department Head of departament’s signature

…………….. ……………………………….