

# SYLLABUS

## 1. Information about the program

1.1 Higher education institution	UNIVERSITY POLITEHNICA OF TIMISOARA
1.2 Faculty <sup>1</sup> / Department <sup>2</sup>	ELECTRONICS, TELECOMUNICATON AND INFORMATION TECHNOLOGIES / Automation and Applied Informatics Department
1.3 Field of study (name/code <sup>3</sup> )	ELECTRONIC ENGINEERING, TELECOMUNICATION AND INFORMATION TECHNOLOGIES
1.4 Study cycle	License
1.5 Study program (name/code/qualification)	TST-ENG/20/20/10/100/10/TST-ENG

## 2. Information about the discipline

2.1 Name of discipline/ formative category <sup>4</sup>	Programming Languages 1/ DF						
2.2 Coordinator (holder) of course activities	Assoc. prof. Adriana ALBU, PhD.						
2.3 Coordinator (holder) of applied activities <sup>5</sup>	Monica NEDELCEA						
2.4 Year of study <sup>6</sup>	1	2.5 Semester	1	2.6 Type of evaluation	D	2.7 Regime of discipline <sup>7</sup>	DII

## 3. Total estimated time – hours / semester: direct teaching activities (fully assisted or partly assisted) and individual training activities (unassisted)<sup>8</sup>

3.1 Number of fully assisted hours / week	4 of which:	3.2 course	2	3.3 seminar / laboratory / project	0/2/0
3.1* Total number of fully assisted hours / semester	56 of which:	3.2* course	28	3.3* seminar / laboratory / project	0/28/0
3.4 Number of hours partially assisted / week	0 of which:	3.5 training	0	3.6 hours for diploma project elaboration	0
3.4* Total number of hours partially assisted / semester	0 of which:	3.5* training	0	3.6* hours for diploma project elaboration	0
3.7 Number of hours of unassisted activities / week	3.14 of which:	additional documentary hours in the library, on the specialized electronic platforms and on the field			1
		hours of individual study after manual, course support, bibliography and notes			1
		training seminars / laboratories, homework and papers, portfolios and essays			1.14
3.7* Number of hours of unassisted activities / semester	44 of which:	additional documentary hours in the library, on the specialized electronic platforms and on the field			14
		hours of individual study after manual, course support, bibliography and notes			14
		training seminars / laboratories, homework and papers, portfolios and essays			16
3.8 Total hours / week <sup>9</sup>	7.14				
3.8* Total hours /semester	100				
3.9 Number of credits	4				

## 4. Prerequisites (where applicable)

<sup>1</sup> The name of the faculty which manages the educational curriculum to which the discipline belongs

<sup>2</sup> The name of the department entrusted with the discipline, and to which the course coordinator/holder belongs.

<sup>3</sup> The code provided in HG - on the approval of the Nomenclature of fields and specializations / study programs, annually updated.

<sup>4</sup> Discipline falls under the educational curriculum in one of the following formative disciplines: Basic Discipline (DF), Domain Discipline (DD), Specialist Discipline (DS) or Complementary Discipline (DC).

<sup>5</sup> Application activities refer to: seminar (S) / laboratory (L) / project (P) / practice/training (Pr).

<sup>6</sup> Year of studies in which the discipline is provided in the curriculum.

<sup>7</sup> Discipline may have one of the following regimes: imposed discipline (DI) or compulsory discipline (DOb)-for the other fundamental fields of studies offered by UPT, optional discipline (DO) or optional discipline (Df).

<sup>8</sup> The number of hours in the headings 3.1 \*, 3.2 \*, ..., 3.8 \* is obtained by multiplying by 14 (weeks) the number of hours in headings 3.1, 3.2, ..., 3.8. The information in sections 3.1, 3.4 and 3.7 is the verification keys used by ARACIS as: (3.1) + (3.4) ≥ 28 hours / wk. and (3.8) ≤ 40 hours / wk.

<sup>9</sup> The total number of hours / week is obtained by summing up the number of hours in points 3.1, 3.4 and 3.7.

4.1 Curriculum	<ul style="list-style-type: none"> <li>not applicable</li> </ul>
4.2 Competencies	<ul style="list-style-type: none"> <li>not applicable</li> </ul>

## 5. Conditions (where applicable)

5.1 of the course	<ul style="list-style-type: none"> <li>Video projector and whiteboard</li> </ul>
5.2 to conduct practical activities	<ul style="list-style-type: none"> <li>18 computers (with C programming environment), video projector and whiteboard</li> </ul>

## 6. Specific competencies acquired through this discipline

Specific competencies	<ul style="list-style-type: none"> <li>To properly use information technology</li> <li>To communicate in technical language</li> <li>To find the right algorithms for solving given problems</li> </ul>
Professional competencies ascribed to the specific competencies	<ul style="list-style-type: none"> <li>Application of knowledge, concepts and basic methods related to computer system architecture, microprocessors, microcontrollers, programming languages and techniques</li> </ul>
Transversal competencies ascribed to the specific competencies	<ul style="list-style-type: none"> <li>Methodical analysis of field-related problems aimed at identifying acknowledged solutions, thus ensuring the accomplishment of professional tasks.</li> <li>Adaptation to new technologies, professional and personal development through continuous training, using printed documentation sources, specialized software and electronic resources in Romanian and at least one foreign language.</li> </ul>

## 7. Objectives of the discipline (based on the grid of specific competencies acquired - pct.6)

7.1 The general objective of the discipline	<ul style="list-style-type: none"> <li>Acquiring the fundamental concepts of programming, with examples in the C programming language</li> </ul>
7.2 Specific objectives	<ul style="list-style-type: none"> <li>Acquiring an overview of the computers' domain and of the programming field</li> <li>Designing and implementing C programs of small and medium complexity</li> <li>Obtaining skills regarding testing and debugging programs</li> <li>Developing a correct programming style</li> </ul>

## 8. Content <sup>10</sup>

8.1 Course	Number of hours	Teaching methods <sup>11</sup>
1. Introduction. Programming basics	1	Presentation of theoretical aspects, examples, discussions, solved problems, questions. All the educational resources are available on "Campus Virtual UPT" platform.
2. Logical diagrams	2	
3. Data and data types (General concepts, Data types description)	2	
4. A first C program (Stages of a C program implementation, The structure of a C program)	1	
5. Library functions (Input/output functions, Mathematical functions, Conversion functions, Other functions)	3	
6. Instructions (Simple instructions, Alternative instructions, Repetitive instructions)	3	

<sup>10</sup> It details all the didactic activities foreseen in the curriculum (lectures and seminar themes, the list of laboratory works, the content of the stages of project preparation, the theme of each practice stage). The titles of the laboratory work carried out on the stands shall be accompanied by the notation "(\*)".

<sup>11</sup> Presentation of the teaching methods will include the use of new technologies (e-mail, personalized web page, electronic resources etc.).

7. Arrays (Single-dimensional arrays, Two-dimensional arrays)	3	
8. Strings (Characters, Strings)	2	
9. Pointers (Pointer variables, Operations with pointers, Pointers and other elements)	1	
10. User defined functions	3	
11. User defined types (structure, enumeration, union)	3	
12. Files (File handling, Header files)	4	
Bibliography <sup>12</sup>		
1. Adriana ALBU: "Computer Programming – The C Language", Conspress, Bucuresti 2013, ISBN 978-973-100-270-5		
2. Brian KERNIGHAN, Dennis RITCHIE: "The C Programming Language", 2nd Edition, Prentice-Hall, 1988, ISBN 0-13-110370-9		
3. "Programming Tutorials – C Tutorial", <a href="http://www.cprogramming.com/tutorial/c-tutorial.html">http://www.cprogramming.com/tutorial/c-tutorial.html</a> , accessed: October 2022		
<b>8.2 Applied activities <sup>13</sup></b>	<b>Number of hours</b>	<b>Teaching methods</b>
Logical diagrams	2	Theoretical presentations, discussions, explanations, problem solving, case studies
Data and data types. A first C program	2	
Library functions	2	
Instructions	4	
Arrays	4	
Strings	2	
Pointers	2	
User defined functions	4	
User defined types	2	
Files	4	
Bibliography <sup>14</sup>		
1. Adriana ALBU: "Computer Programming – The C Language", Conspress, Bucuresti, 2013, ISBN 978-973-100-270-5		

**9. Corroboration of the content of the discipline with the expectations of the main representatives of the epistemic community, professional associations and employers in the field afferent to the program**

- The basics of programming are important for all disciplines (belonging to the curriculum of this study program) that have connections to software development (e.g., Applied Computer Programming, Programming Languages 2, Programming Languages 3).
- The main employers in the field of this study program ask for both general programming knowledge and knowledge of the C programming language.

**10. Evaluation**

Type of activity	10.1 Evaluation criteria <sup>15</sup>	10.2 Evaluation methods	10.3 Share of the final grade
<b>10.4 Course</b>	Two multiple choices written tests (30 theoretical and practical questions; each question has four or five possible answers of which one only is correct)	Written examination	2/3
<b>10.5 Applied activities</b>	<b>S:</b>		
	<b>L:</b> Two practical tests (the following aspects are appreciated: a proper implementation, an adequate way of presenting solutions, correct answers to the	Practical examination (on a computer)	1/3

<sup>12</sup> At least one title must belong to the discipline team and at least one title should refer to a reference work for discipline, national and international circulation, existing in the UPT library.

<sup>13</sup> Types of application activities are those specified in footnote 5. If the discipline contains several types of applicative activities then they are sequentially in the lines of the table below. The type of activity will be in a distinct line as: "Seminar:", "Laboratory:", "Project:" and / or "Practice/training".

<sup>14</sup> At least one title must belong to the discipline team.

<sup>15</sup> Syllabus must contain the procedure for assessing the discipline, specifying the criteria, methods and forms of assessment, as well as specifying the weightings assigned to them in the final grade. The evaluation criteria shall be formulated separately for each activity foreseen in the curriculum (course, seminar, laboratory, project). They will also refer to the forms of verification (homework, papers, etc.)

	questions)		
	<b>P<sup>16</sup>:</b>		
	<b>Pr:</b>		
<b>10.6</b> Minimum performance standard (minimum amount of knowledge necessary to pass the discipline and the way in which this knowledge is verified <sup>17</sup> )			
<ul style="list-style-type: none"> <li>• Understanding the fundamental concepts of the C programming language</li> <li>• The student should be able to work with the basic elements of the C programming language (data types, variables, function, instructions)</li> <li>• To pass the multiple choices written tests – 50% of the questions must have correct answers</li> <li>• The practical tests are passed if the programs are functional and solve the minimum requirements</li> </ul>			

**Date of completion**

21.07.2023

**Course coordinator  
(signature)**

**Coordinator of applied activities  
(signature)**

**Head of Department  
(signature)**

**Date of approval in the Faculty Council <sup>18</sup>**

14.09.2023

**Dean  
(signature)**

---

<sup>16</sup> In the case where the project is not a distinct discipline, this section also specifies how the outcome of the project evaluation makes the admission of the student conditional on the final assessment within the discipline.

<sup>17</sup> It will not explain how the promotion mark is awarded.

<sup>18</sup> The endorsement is preceded by the discussion of the board's view of the study program on the discipline record.