

## SYLLABUS

### 1. Information on the study programme

1.1. Institution	West University of Timisoara
1.2. Faculty	Mathematics and Computer Science
1.3. Department	Computer Science
1.4. Study program field	Computer Science
1.5. Study cycle	Undergraduate
1.6. Study programme / Qualification*	Computer Science : <i>Database administration / Administrator baze de date - 252101; Computer network administration / Administrator de retea de calculatoare - 252301; Analyst / Analist - 251201; Research assistant in computer science / Asistent de cercetare în informatica - 214918; Teacher in secondary schools / Profesor în învățământul gimnazial - 233002; Programmer / Programator - 251202; Software systems designers / Proiectant sisteme informatice - 251101</i>

### 2. Information on the course

2.1. Course title	Practice Stage I						
2.2. Lecture instructor							
2.3. Seminar / laboratory instructor	Alexandru Ionașcu						
2.4. Study year	1	2.5. Semester	2	2.6. Examination type	C	2.7. Course type	DI

### 3. Estimated study time (number of hours per semester)

3.1. Attendance hours per week	1	out of which: 3.2. lecture	0	3.3. seminar/laboratory	1
3.4. Attendance hours per semester	14	out of which: 3.5. lecture	0	3.6. seminar/laboratory	14
<b>Distribution of the allocated amount of time*</b>					<b>hours</b>
Study of literature, course handbook and personal notes					10
Supplementary documentation at library or using electronic repositories					12
Preparing for laboratories, homework, reports, etc.					34
Exams					4
Tutoring					8
3.7. Total number of hours of individual study	68				
3.8. Total number of hours per semester	82				
3.9. Number of credits (ECTS)	2				

### 4. Prerequisites (if it is the case)

4.1. curriculum	Programming I, Algorithms and Data structures I
4.2. competences	Basic knowledge of logic and problem-solving abilities, Proficiency in English

### 5. Requirements (if it is the case)

5.1. for the lecture	-
5.2. for the seminar/laboratory	Face-to-face variant: Properly equipped laboratory room (computers with C / C ++ installed)

	Online variant: Computer with internet connection, video camera and microphone
--	--

## 6. Specific acquired competences

Professional skills	<ul style="list-style-type: none"> <li>• Good knowledge of procedural and object-oriented concepts</li> <li>• Introduction to generic programming</li> <li>• Ability to implement small-sized procedural and object-oriented projects in C/C++</li> <li>• Usage of an Integrated Development Environment</li> </ul>
Transversal skills	<ul style="list-style-type: none"> <li>• Ability to build complex systems based on elementary building blocks</li> <li>• Develop an analytical spirit and curiosity about how computer software works</li> <li>• Conceptual modelling, i.e., ability to represent real-life problems using abstract models</li> <li>• Ability to work in a team</li> </ul>

## 7. Course objectives

7.1. General objective	Assimilate and ability to operate with procedural and object-oriented paradigm in C/C++; develop an object-oriented mindset
7.2. Specific objectives	<p><i>Knowledge-wise objectives (KO):</i> (1) Good knowledge of procedural and object-oriented paradigms (2) Basic knowledge of generic programming concepts (3) Basic conceptual modelling and object-oriented design; (4) Code the concepts of procedural programming in C programming language; (5) Represent OOP and generic programming concepts in C++ programming language</p> <p><i>Ability-wise objectives (AbO):</i> (1) Ability to design simple problems using OOP concepts; (2) Ability to implement and test simple problems in C and C++ programming languages; (3) Ability to debug programmes.</p> <p><i>Attitude-wise objectives (AtO):</i> (1) Root cause analysis; (2) Generalization and conceptualization of real-life problems</p>

## 8. Content\*

8.1. Lecture	Teaching methods	Remarks, details
-	-	-
8.2. Seminar/laboratory	Teaching methods	Remarks, details
L1. (2h) Presentation of the work environment. Stages of compilation, linking and execution. Enunciation and distribution of the individual project in the C language (Project 1).	Problematization, dialogue, collaborative learning, discovery learning.	Students have access to the synthesis related to the laboratory topic and to the statements of the problems available through specific e-uvt.ro platforms. Communication through

		Google Classroom and Google Meet platforms.
L2. (2h) Working with Files. Presentation and analysis of the deliverables of the individual project.	Problematization, dialogue, collaborative learning, discovery learning. Evaluation.	The teacher provides additional details, answers students' questions and checks / evaluates how the students solved the problems.
L3. (2h) Data structures. Presentation and analysis of the deliverables of the individual project.	Problematization, dialogue, collaborative learning, discovery learning. Evaluation.	Idem
L4. (2h) Presentation and analysis of the individual project.	Project 1 final evaluation.	Idem
L5. (2h) Presentation and analysis of the individual project. Enunciation and distribution of the team project in the C ++ language (Project 2).	Project 1 final evaluation. Problematization, dialogue, collaborative learning, discovery learning.	Idem
L6. (2h) Presentation and analysis of the deliverables of the team project.	Evaluation.	Idem
L7. (2h) Presentation and analysis of the deliverables of the team project.	Evaluation.	Idem
<b>Recommended bibliography</b> <ol style="list-style-type: none"> <li>1. <b>B. Kernighan and D. Ritchie - The C Programming Language (2nd edition), Prentice-Hall, 1988.</b></li> <li>2. B. Kernighan și D. Ritchie - Limbajul C, Editura Teora, 2000.</li> <li>3. <b>Bjarne Stroustrup - The C++ Programming Language (3rd edition), Addison Wesley, 1997.</b></li> <li>4. Bjarne Stroustrup - Limbajul de programare C++, Editura Teora, 2003.</li> <li>5. L. Negrescu – Limbajele C și C++ pentru începători, vol. 1, Ed. Albastră, Cluj-Napoca, 2001.</li> </ol>		

### 9. Correlations between the content of the course and the requirements of the professional field and relevant employers

<p>The content is in line with the structure of similar courses at other universities and covers the fundamental aspects necessary to become familiar with the issue of procedural and object-oriented programming. The ability to identify, design, implement and analyze problems that can be solved using object-oriented principles is essential for any IT activity. The skills offered by this discipline are necessary for an IT specialist to identify solutions to solve specific problems, regardless of the specific field of activity.</p>
--

### 10. Evaluation\*

Activity	10.1. Assessment criteria**	10.2. Assessment methods***	10.3. Weight in the final mark
10.4. Lecture	-	-	-
10.5. Seminar/Laboratory	Ability to solve a problem in the C programming language. (AbO2, AbO3, AtO1, AtO2)	Project 1 evaluation (C, individual project), out of which: <b>10%</b> -	50%

		intermediate evaluation during weeks 3-4 (L2); <b>10%</b> - intermediate evaluation during weeks 5-6 (L3); and <b>30%</b> - project final evaluation during weeks 7-10 (L4-5).	
	Ability to solve a problem in the C++ programming language. (AbO1, AbO2, AbO3, AtO1, AtO2)	Project 2 evaluation (C++, team project), out of which: <b>10%</b> - intermediate evaluation during weeks 11-12 (L6); <b>10%</b> - intermediate evaluation during weeks 13-14 (L7); and <b>30%</b> - project final evaluation in the exam session.	50%
<p>10.6. Minimal needed performance for passing</p> <ul style="list-style-type: none"> <li>• Knowledge of the notions introduced by procedural and object-oriented programming.</li> <li>• Developing a simple program based on the object-oriented paradigm and implementing algorithms based on the peculiarities of the C language.</li> </ul>			
<p>The final grade is computed as a weighted average of the grades given for the components specified in 10.5. The exam is considered passed if the average is at least 5. At each of the exam sessions the grade is computed according to the same rule. The student need to re-take only the failed component, unless the student wishes to re-take the component already passed.</p> <p>All students all welcome to tutoring meetings as scheduled by the department.</p>			

Date  
3.02.2021

Signature (lecture instructor)

Signature (seminar instructor)

Signature (director of the department)  
Lect. Dr. Flavia Micota