

**SYLLABUS / FIȘA DISCIPLINEI**
**1. Information on the study programme / Date despre programul de studii**

1.1. Institution / Instituția de învățământ superior	Universitatea de Vest din Timișoara
1.2. Faculty / Facultatea	Matematică și Informatică
1.3. Department / Departamentul	Computer Science (Informatică)
1.4. Study program field	Computer Science (Informatică)
1.5. Study cycle/ Ciclul de studii	Bachelor / licență
1.6. Study programme / Programul de studii / calificarea*	Computer Science / Informatică în limba engleză / Database administration / <i>Administrator baze de date - 252101; Computer network administration / Administrator de rețea de calculatoare - 252301; Analyst / Analist - 251201; Research assistant in computer science / Asistent de cercetare în informatică - 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 / Date despre disciplină**

2.1. Title of the course / Denumirea disciplinei	Programming I						
2.2. Teacher in charge of the course / Titularul activităților de curs	Cosmin Bonchis						
2.3. Teacher in charge of the seminar / Titularul activităților de seminar	Cosmin Bonchis						
2.4. Study year / Anul de studii	1	2.5. Semester / Semestrul	1	2.6. Examination type / Tipul de evaluare: E(xam)/C(olloquim)	E	2.7. Course type / Regimul disciplinei: M(andatory)/ E(lective)/ F(acultative)	DI

**3. Estimated study time (number of hours per semester) /Timpul total estimat (ore pe semestru al activităților didactice)**

3.1. Attendance hours per week / Număr de ore pe săptămână	4	out of which din care: 3.2 lecture/ curs	2	3.3. seminar/laborator	2
3.4. Attendance hours per semester / Total ore din planul de învățământ	56	out of which: 3.5 lecture / curs	28	3.6. seminar/laborator	28
<b>Distribution of the allocated amount of time / Distribuția fondului de timp*</b>					<b>hours/ore</b>
Individual study /Studiu după manual, suport de curs, bibliografie și notițe					10
Supplementary documentation at library or using electronic repositories / Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					20
Preparing for laboratories, homework, reports etc. /Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					60
Exams / Examinări					20

Tutoring / Tutorat		14
3.7. Total number of hours of individual study / Total ore studiu individual	124	
3.8. Total number of hours per semester / Total ore pe semestru	180	
3.9. Number of credits (ECTS) / Număr de credite	6	

#### 4. Prerequisites (if it is the case) / Precondiții (acolo unde e cazul)

4.1. curriculum / de curriculum	
4.2. skills / de competențe	

#### 5. Requirements (if it is the case) / Condiții (acolo unde e cazul)

5.1. for the lecture / de desfășurare a cursului	
5.2. for the seminar, laboratory / de desfășurare a seminarului/laboratorului	

#### 6. Acquired skills / Competențe specifice acumulate

Professional skills / Competențe profesionale	<ul style="list-style-type: none"> <li>• Learning the correct terminology and correct understanding of concepts and their utility</li> <li>• Understanding the programming specific mechanisms</li> <li>• Understanding the characteristics of language and the applications.</li> <li>• The ability to analyze specific situations and to interpret / explain the correct meaning of a sequence of code / program</li> <li>• Developing the capacity to design and carry out end-cycle development of complex applications, small / medium</li> </ul>
Transversal skills / Competențe transversale	<ul style="list-style-type: none"> <li>• Responsiveness / attention, interest shown by individual search questions and answers</li> </ul>

	<ul style="list-style-type: none"> <li>Helping colleagues in all occasions except examination activities</li> </ul>
--	---

### 7. Objectives of the course / Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1. General objective / Obiectivul general al disciplinei	<p>The course is intended to familiarize the students with a programming language (the C language). We intend to acquire the basic elements of the C language and learning the minimum experience in their use. Students should improve their knowledge and skills needed to use an integrated application development environment.</p> <p>C programming language continues to be, about 30 years from appearance - more than half the time since there are computers and programming languages! - one of the most used languages.</p> <p>Concise syntax and semantics, along with total freedom in choosing the solution approach inspired fear those who do not understand and pleasure and safety of those who know him.</p>
7.2. Specific objectives / Obiectivele specifice	<p>The C language course presents elements of a functional perspective, focusing on watched / utility methods implementation. Starting from a functional perspective of a computer generally is presented motivation and how to implement various types of data used in the C language, the possible actions on them and how to structure a program. Over this, knowledge with highly intuitive language elements are introduced, calling for illustrate some significant examples.</p>

### 8. Content / Conținuturi\*

8.1. Lecture / Curs	Teaching strategies / Metode de predare	Remarks, details / Observații
1. Introduction (2 hours) <ol style="list-style-type: none"> <li>Functional schema for a generic computer</li> <li>Steps for development of an applications</li> <li>The C programming language: history</li> </ol>	Lectures, illustration, demonstration	2h
2. Data (4 hours) <ol style="list-style-type: none"> <li>Classification</li> <li>Methods of data representation</li> <li>Data Types</li> </ol>	Lectures, illustration, demonstration	4h

<p>4. Specifying constants</p> <p>5. Variables: declaration, storage class, field, vision, life.</p>		
<p>3. Operators and expressions (2 hours)</p> <ul style="list-style-type: none"> <li>• Classes of operators</li> <li>• Previous assessment order operators and expressions</li> <li>• Implicit type conversion</li> </ul>	Lectures, illustration, demonstration	2h
<p>4. Control of execution (2 hours)</p> <ul style="list-style-type: none"> <li>• Instruction Test</li> <li>• Instruction selection of multiple choice</li> <li>• Instructions for cycling</li> <li>• Instruction execution cycle control</li> <li>• Instruction of jump unconditional</li> <li>• Instruction to return the call function</li> </ul>	Lectures, illustration, demonstration	2h
<p>5. Functions (2 hours)</p> <ul style="list-style-type: none"> <li>• Declaration, call, definition</li> <li>• Communication between functions</li> <li>• Call context</li> <li>• Recursive functions</li> </ul>	Lectures, illustration, demonstration	2h
<p>6. Pointers and Arrays..(6 hours)</p> <ul style="list-style-type: none"> <li>• Pointers and Addresses</li> <li>• Pointers and Function Arguments</li> <li>• Pointers and Arrays</li> <li>• Address Arithmetic</li> <li>• Character Pointers and Functions</li> <li>• Pointer Arrays; Pointers to Pointers</li> <li>• Multi-dimensional Arrays</li> <li>• Initialization of Pointer Arrays</li> <li>• Pointers vs. Multi-dimensional Arrays</li> <li>• Command-line Arguments</li> <li>• Pointers to Functions</li> <li>• Complicated Declarations</li> </ul>	Lectures, illustration, demonstration	6h
<p>7. Structures (4 hours)</p> <ul style="list-style-type: none"> <li>• Basics of Structures</li> <li>• Structures and Functions</li> <li>• Arrays of Structures</li> <li>• Pointers to Structures</li> <li>• Self-referential Structures</li> <li>• Table Lookup</li> <li>• Typedef.</li> </ul>	Lectures, illustration, demonstration	4h

<ul style="list-style-type: none"> <li>• Unions</li> <li>• Bit-fields</li> </ul>		
<p>8. Using External Libraries (2 hours)</p> <ul style="list-style-type: none"> <li>• Symbols and Linkage</li> <li>• Static vs. Dynamic Linkage</li> <li>• Linking External Libraries</li> <li>• Symbol Resolution Issues</li> </ul>	Lectures, illustration, demonstration	2h
<p>9. Input and Output (2 hours)</p> <ul style="list-style-type: none"> <li>• Standard Input and Output.</li> <li>• Formatted Output – printf</li> <li>• Variable-length Argument Lists</li> <li>• Formatted Input – Scanf</li> <li>• File Access</li> <li>• Error Handling - Stderr and Exit</li> <li>• Line Input and Output</li> <li>• Miscellaneous Functions</li> <li>• String Operations</li> <li>• Symbols and Linkage</li> <li>• Static vs. Dynamic Linkage</li> <li>• Linking External Libraries</li> <li>• Symbol Resolution Issues</li> </ul>	Lectures, illustration, demonstration	2h
<p>10 Standard Library (2 hours)</p> <ul style="list-style-type: none"> <li>• &lt;stdio.h&gt;</li> <li>• &lt;ctype.h&gt;</li> <li>• &lt;stdlib.h&gt;</li> <li>• &lt;assert.h&gt;</li> <li>• &lt;stdarg.h&gt;</li> <li>• &lt;time.h&gt;</li> </ul>	Lectures, illustration, demonstration	2h
<p><b>Recommended bibliography / Bibliografie</b>  Kernigham B. and D. Ritchie - The C Programming Language, 2nd ed., Prentice-Hall, 1988  Ivor Horton – Beginning C: From Novice to Professional  Steve Oualline - Practical C Programming, Third Edition</p> <p>Online lectures:  C Programming. Brian Brown, Central Institute of Technology, NZ. Constantin quizzes  C Programming Steven Summit, Experimental College, University of Washington, USA.  Introduction to C Programming, University of Leicester, UK.  C Programming. Steve Holmes, University of Strathclyde, UK.  C Language Tutorial. Drexel University, USA. A short introduction</p>		

<b>8.2. Seminar, lab / Seminar, laborator</b>	<b>Teaching/learning strategies / Metode de predare/ învățare</b>	<b>Remarks, details / Observații</b>
1. Steps for development of an applications	Problem solving, questioning, dialogue	2h
2. Methods of data representation	Problem solving, questioning, dialogue	4h
3. Operators and expressions	Problem solving, questioning, dialogue	2h
4. Control of execution	Problem solving, questioning, dialogue	2h
5. Functions	Problem solving, questioning, dialogue	2h
6. Pointers and Arrays	Problem solving, questioning, dialogue	6h
7. Structures and Functions	Problem solving, questioning, dialogue	4h
8. Symbols and Linkage	Problem solving, questioning, dialogue	2h
9. Input and Output	Problem solving, questioning, dialogue	4h
<b>Recommended bibliography / Bibliografie</b> Kernigham B. and D. Ritchie - The C Programming Language, 2nd ed., Prentice-Hall, 1988 Ivor Horton – Beginning C: From Novice to Professional Steve Oualline - Practical C Programming, Third Edition  Online lectures: C Programming. Brian Brown, Central Institute of Technology, NZ. Constantin quizzes C Programming Steven Summit, Experimental College, University of Washington, USA. Introduction to C Programming, University of Leicester, UK. C Programming. Steve Holmes, University of Strathclyde, UK. C Language Tutorial. Drexel University, USA. A short introduction		

**9. Correlations between the content of the course and the requirements of the IT field / Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului**

**10. Evaluation / Evaluare\***

Activity / Tip de activitate	10.1. Evaluation criteria / Criterii de evaluare**	10.2. Evaluation methods / Metode de evaluare***	10.3. Weight in the averaged mark / Pondere din nota finală
10.4. Lecture / Curs	<i>Theoretical Examination (multiple choice)</i>	Quiz with multiple choices	25%

	<i>Practical Examination (the ability to write programs)</i>	Individual written / oral examination	25%
10.5. Seminar/ lab	Teoretical continuous testing throughout the semester	Quiz with multiple choices	25%
	Activities such as homework/projects	Individual programming examination	25%
10.6. Minimal knowledge for passing / Standard minim de performanță			
<p><i>Knowledge of basic principles of programming.</i>  <i>Knowing the structure of a computer program. Getting an average over 5 for all tests and homework during the semester. Minimum 5 per examination task (multiple choice, practical test).</i></p>			

Date/ Data completării

10.10.2016

Signature (lecture) /  
Semnătura titularului de curs



Signature (seminar)  
Semnătura titularului de seminar



Signature (director of the department)  
Semnătura directorului de departament  
Conf.dr. Victoria Iordan