



### CIT 203 – Programming Languages I Teaching Plan

<b>Course Name</b>		Programming Languages I	
<b>Course Code</b>		CIT 203	
<b>Course Type</b>		Compulsory course	
<b>Course Level</b>		Undergraduate	
<b>ECTS Credits</b>		6 ECTS	
<b>Course hours per week (Institutional)</b>		3	
<b>Practice hours per week</b>		-	
<b>Laboratory hours per week</b>		2	
<b>Academic Semester</b>		2013-2014 Fall	
<b>Course coordinator(s)</b>		Dr. Hüseyin Lort	
<b>Teaching system</b>			
<b>Medium language</b>		English	
<b>Prerequisite</b>		CIT 102 – Information Technology in Education II	
<b>Other recommended matters</b>		No	
<b>Training required</b>		No	
<b>Aim of the course</b>		<p>The major goals of this course are the followings:</p> <ul style="list-style-type: none"> <li>• Understanding the use and the purpose of computer programming.</li> <li>• Teaching the principles and practices of computer programming.</li> <li>• Solving business and educational problems using computer programming.</li> <li>• Designing and developing computer programs with particular reference to the structural programming paradigm.</li> </ul>	
<b>Learning outcomes</b>		<p>At the end of this course students should :</p> <ol style="list-style-type: none"> <li>1. Understand the use and the necessity for computer programs and programming languages.</li> <li>2. Understand how to solve business and educational problems using computer programming constructs.</li> <li>3. Declare and define what a variable is in computer programming.</li> <li>4. Declare and define what a programming sequence is in computer programming.</li> <li>5. Understand multi-assignment statements, increment and decrement operations.</li> <li>6. Understand the necessity and the use of basic computer programming constructs such as programming sequence, selection statement, iteration and functions.</li> <li>7. Effectively use and combine different programming constructs to solve complicated programming tasks.</li> <li>8. Use programming functions to divide solutions into repeatable and reusable programming patterns.</li> <li>9. Program computer programs in a structure programming paradigm using visual C++.</li> <li>10. Learning to teach basic computer programming.</li> </ol>	
<b>Course Content</b>			
<b>Course content per week</b>	<b>Week</b>	<b>Topics</b>	
		<b>Theory</b>	<b>Practice</b>
	1	Introduction to algorithms and flowcharts	Constructing algorithms and their corresponding flowcharts

	2	Programming libraries, standard input/output in programming, data types	Defining Variables in Visual C++. What is a variable and why do we use it in programming? Validation of variable names. Data types of variables.
	3	Programming sequence. Arithmetic and logical operations in computer programming. Input-output operations, constants, defining and using constants.	Exercises on understanding programming sequence. What is the difference between a variable and a constant? Arithmetic and logical operations in Visual C++
	4	Selection and Decision making in programming. Conditional statements (if/else). Selection among many of choices.	Selection and decision making exercises in visual C++. Operation order, operands and operators in programming. Using if/else as selection statement to solve various business problems.
	5	Nested selection and decision making in computer programming (if, else if, else and switch statement)	Selection and decision making exercises in visual C++ with nested if statements and the switch statement.
	6	Iteration, recursion, recursive statements and loops (while,for,do...while)	Using iteration to solve business problems in Visual C++. Using loops to solve problems. Infinite loops and recursion.
	7	Review	QUIZ
	8	Programming exercises	
	9	Midterm Exam	
	10	nested loops, multiple usage of loops (while, for, do...while)	Problems related to recursion and loops, creating matrices through loops.
	11	Combining decision making, loops and programming sequence.	Problem solving with decision making and loops
	12	Functions in computer programming. Creating, calling and using user-defined functions.	Exercise on functions. Creating repeatable patterns using Visual C++ functions.
	13	Combining functions with other programming constructs (such as loops or conditional statements)	Creating abstract solutions in computer programming. Argument passing. Pass by value, pass by reference. Return type of functions.
	14	Review	QUIZ
	15	Programming exercises	

<b>Course book :</b>	<b>Course book:</b> C How to program, Dr. Harvey Deitel, Prentice Hall, 5th Edition (January 2005).	
<b>Evaluation</b>		
Quiz:	30%	
Midterm exam:	30%	
Final exam:	40%	
<b>Semester Activities</b>	<b>Number</b>	<b>Contribution percentage to course mark %</b>
Midterm Exam	<b>1</b>	<b>30</b>
Final Exam	<b>1</b>	<b>40</b>
Quiz	<b>2</b>	<b>30</b>
<b>TOTAL</b>		<b>100</b>
<b>Calculating workload (Teaching, learning and evaluation)</b>		
<b>THIS SECTION IS DONE SEPERATELY, PLEASE SEE THE ATTACHMENT</b>		

### Programme and learning outcomes

Learning Outcomes (LO)	Programme Outcomes (PO)																
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PO 13	PO 14	PO 15	PO 16	PO 17
LO1	3									5							
LO2				3	3					5							
LO3					3					5							
LO4					3					5							
LO5					3					5							
LO6					5					5							
LO7	3			5	5	5				5							
LO8	3			5	5	5				5							
LO9				5	5					5							
LO10	3	5	5		3	3	3	5		4				4	5	4	

#### Contribution Level:

1 very low  
4 high

2 low  
5 very high

3 medium

## **CITT Department Programme Outcomes**

- 1.** Having adequate level of knowledge and skills in current/new computing and educational technologies.
- 2.** Having sufficient communication and teaching skills in teaching profession.
- 3.** Being able to teach updated computing technologies efficiently in English.
- 4.** Being able to identify information technology problems through using various analysis and synthesis.
- 5.** Being pragmatic to develop and apply persistent information technology solutions to educational and business problems.
- 6.** Being able to use critical and computational thinking skills to produce alternative solutions at every level of project development life-cycle.
- 7.** Being capable to work in disciplinary and interdisciplinary teamwork.
- 8.** Being sensitive, reactive and responsive to professional, social and ethical issues. Having social and ethical awareness in teaching and in providing solutions to problems.
- 9.** Having adequate level of knowledge and skills in current/new computer hardware, operating systems and computer networks.
- 10.** Adequate level of knowledge and skills in current/new programming languages, programming paradigms (procedural and object-oriented) and programming environments (visual, console-based programming).
- 11.** Being able to analyse, plan and manage educational software design and project development.
- 12.** Having the capability of evaluating and criticising educational software design and development.
- 13.** Adequate level of knowledge in using and integrating current/new e-learning and distance education systems such as learning management systems (LMS).
- 14.** Having sufficient skills and knowledge in using instructional technology and material design.
- 15.** Having skills to apply and use special teaching approaches, theories, teaching strategies, methods and techniques (such as to those people with disabilities).
- 16.** Using appropriate measurement and evaluation techniques to assess students' learning and development in addition to supporting them with good level of feedback.