



### CIT 307– Web design and web based programming Teaching Plan

<b>Course Name</b>		Web design and web based programming	
<b>Course Code</b>		CIT 307	
<b>Course Type</b>		Compulsory course	
<b>Course Level</b>		Undergraduate	
<b>AKTS Credit</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>Instruction system</b>			
<b>Medium language</b>		English	
<b>Prerequisite</b>		CIT 102 – Information Technology in Education II	
<b>Suggestions related to course</b>		N/A	
<b>Training required</b>		N/A	
<b>Aim of the course</b>		<p>The major goals of this course are the followings:</p> <ol style="list-style-type: none"> <li>1. Provide a practical introduction to some of the basic technologies that are used to develop web pages and systems.</li> <li>2. Teach the principles of web design and programming, with particular reference to HTML and CSS.</li> <li>3. Give experience of designing and developing enterprise-level web pages/sites using various technologies.</li> <li>4. Explain the need for various web technologies and development environments.</li> <li>5. Examine the advances in Web and Internet technologies.</li> <li>6. Describe the ideas behind the standard mark-up languages of web pages (HTML, XHTML).</li> <li>7. Using Cascading StyleSheets (CSS) efficiently for advanced web design.</li> </ol>	
<b>Learning outcomes</b>		<p>At the end of this course students should,</p> <ol style="list-style-type: none"> <li>1. Understand the basic technologies underlying web pages and systems.</li> <li>2. Understand and apply web-based concepts.</li> <li>3. Design and implement web-based systems.</li> <li>4. Solve problems using web-based systems.</li> <li>5. Create and design basic to advanced looking web pages using stylesheets (CSS) and mark-up languages (HTML).</li> <li>6. Learn how to create external and embedded style sheets that enhance the appearance of web pages.</li> <li>7. Understand how to use fundamental functionalities of HTML forms on web applications.</li> <li>8. Describe the difference between client-side and server-side script programming.</li> </ol>	
<b>Course Content</b>			
	<b>Week</b>	<b>Topics</b>	
		<b>Theory</b>	<b>Practice</b>

<b>Course content per week</b>	1	Introduction to web technologies, What is Internet? Client/Server side programming	Introduction to HTML and XHTML. Examples of Client and server side technologies
	2	Browsers, Basic technologies underlying web, HTML/XHTML tags and attributes	XHTML : Font, Paragraph, Bold, Italic, Underline
	3	Good web design principles and web standards	XHTML : Anchor Tag, Internal linking, external linking, mail link
	4	Bad web design, bad design examples	XHTML : html lists, ordered, unordered and definition list
	5	Introduction to HTML 5, differences between HTML 4.0.1 and HTML 5	XHTML : html tables, sub tags and attributes
	6	QUIZ (on Computer)	Quiz Solution and review
	7	Midterm exam	
	8	Web templates, recent changes in web design (XHTML templates)	XHTML : frames Web Project, example of past web projects
	9	Standard tags in XHTML and future standards in web (Flash VS HTML 5)	XHTML : meta, div, span, embed and object
	10	Advanced web design : Introduction to CSS and JavaScript	CSS : Introduction to CSS, Selector and properties
	11	Advanced web Design : XHTML Forms and JavaScript validation	CSS : Advanced CSS, CSS classes, multiple CSS classes and pseudo classes
	12	Web design project presentation	
	13	Quiz	Quiz solution
	14	Review	-
	15	Final exam	
<b>Course book and references :</b>	<p><b>Course book:</b> Internet &amp; World Wide Web: How to Program, Deitel &amp; Deitel, Prentice Hall, 4th Edition (September 2007).</p> <p><b>References:</b> Beginning Web Programming with HTML, XHTML and CSS (Wrox Beginning Guides), (August 2005).</p>		
<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																
LO2				3	3	3				3							
LO3				3	3	3				3							
LO4				4	4	4				5							
LO5	4			4	4	4				5							
LO6	4			4	4	4				5							
LO7	3			5	5	5				5							
LO8	3			3	3	3				3		5	2				

\*Contribution Level:

1 very low      2 low      3 medium      4 high      5 very high

#### **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.