



Course Weekly Outline

Course Name: Compiler II

Course Instructor											
E-mail											
Title											
Course Coordinator											
Course Objective	<p>A. Definition of how to build and design of programming languages by looking at the work of the translator techniques and how to build it</p> <p>B. Training students to design and build programming languages through the implementation of some stages of the translator in the practical side</p> <p>C. Accommodate the student how the data is stored within the memory process through simulation methods of storage</p> <p>D. Increase the possibility of student programming by giving him examples of different issues within the limits set</p>										
Course Description	<p>1 - To distinguish between the types of algorithms of Compiler</p> <p>2 - Determine the best algorithm for designing compiler</p> <p>3 - The language used components to convert any algorithm to the interpreter program</p> <p>4- Determine the evolution in the field of design compilers and programming languages</p> <p>5- Distinction between the types of translators by knowing the the input and output of the compiler</p> <p>6- Take collective project to design and build compiler for some simple programming languages proposed</p>										
Textbook	Compilers Principles, Techniques, and Tools , Aho Law, Addison Wesley										
References	Basics of Compiler Design, T. Mogensen, Copenhagen Uni.										
Course Assessments	<table border="1"> <thead> <tr> <th>Term Tests</th> <th>Laboratory</th> <th>Quizzes</th> <th>Project</th> <th>Final Exam</th> </tr> </thead> <tbody> <tr> <td>30%</td> <td>15%</td> <td>5%</td> <td>-</td> <td>50%</td> </tr> </tbody> </table>	Term Tests	Laboratory	Quizzes	Project	Final Exam	30%	15%	5%	-	50%
	Term Tests	Laboratory	Quizzes	Project	Final Exam						
30%	15%	5%	-	50%							
General Notes											



Course Weekly Outline

Week	Date	Topics Covered	Lab. Experiment Assignments	Notes
1	First week	Introduction to Back-End	First & follow	/
2	Second week	Intermediate Code Generation	First & follow	/
3	Third week	Intermediate Code Generation	First & follow	/
4	Fourth week	Code Optimization Concepts	First & follow	/
5	Fifth week	Local Optimization	Predicative parser	/
6	Sixth week	Data – Flow Analysis	Predicative parser	/
7	Seventh week	Global Optimization	Predicative parser	/
8	Eighth week	Code Generation	Predicative parser	/
9	Ninth week	Code Generation	Predicative parser	/
10	Tenth week	Optimization during Code Generation	Bottom-up	/
11	Eleventh week	Assembler & Loader – Linker Editor	Bottom-up	/
12	Twelfth week	Decompiler concepts	Shift reduce parser	/
13	Thirteenth week	Decompiler concepts	Shift reduce parser	/
14	Fourteenth week	Compiler of Object Oriented Language	Shift reduce parser	/
15	Fifteenth week	Debugging concepts	Shift reduce parser	/
Final Year Exam				

Instructor Signature:

Dean Signature: