



**BAHRIA UNIVERSITY  
DEPARTMENT OF SOFTWARE ENGINEERING  
COURSE SYLLABUS**

Course Title	Course Code	Semester	Credit Hours/Week		Total Credits
			Theory	Practice	
SOFTWARE CONSTRUCTION	SEN-204	FALL 2014	2	1	3
<b>CourseType</b>	Core				
<b>Degree Program</b>	BS Software Engineering				
<b>Prerequisites Course(s) (compulsory)</b>	Having successfully completed the basic courses in Automata and Discrete Mathematics				
<b>Course Instructor(s)</b>					
<b>Name SURNAME</b>	<b>Mail</b>	<b>Web</b>			
Assist. Prof. Dr. Kashif ZIA	<a href="mailto:kashif5095@gmail.com">kashif5095@gmail.com</a>				
<b>Course Assistant(s)/Tutor(s)</b>					
<b>Name SURNAME</b>	<b>Mail</b>	<b>Web</b>			
Ms Rafia Khan	<a href="mailto:rafia_khan2886@yahoo.com">rafia_khan2886@yahoo.com</a>				
<b>Course Web Site</b>					
<b>Course Objectives</b>					
This course deals with the theory and practice of compiler design. Topics emphasized are scanning and parsing, semantic analysis and code generation.					
<b>Course Content</b>					
Compiler Construction, building blocks, Scanner, Parser, Semantic Analyzer and Code Generator. The knowledge base includes Regular Expressions, Finite Automata, Context-free grammars, Parse trees and syntax diagrams, BNF and EBNF, Top-down and Bottom up parsing and related data structures. Implementation of TINY language in OOP paradigm.					
<b>Learning Outcomes of the Course</b>					
<b>Upon successful completion of this course, the enrolled students will be gaining the following knowledge, skills and competences:</b>					
1	Distinguish formal and practical properties of different approaches to parsing				
2	Understand and implement various techniques to parse source code				
3	Implement a basic compiler				
4	Transform an Syntax tree to intermediate representation (IR)				
5	Generate target code from IR				

<b>Course Outline/Schedule (Weekly) Planned Learning Activities</b>			
<b>Week</b>	<b>Topics</b>	<b>Preliminary Preparation</b>	<b>Teaching Methods(Theory, Practice, assignment etc.)</b>
1	Compiler Introduction, History and related components, Building Blocks of compilers		Theory
2	Data Structures in Compilers, Related issues, TINY Language		Theory, Quiz 1
3	Code Scanning Process Regular Expressions for TINY Language		Theory
4	Finite Automata and DFAs		Theory, Assignment 1 (coding scanner)
5	Implementation of TINY Scanner, Practical implications in OOP		Theory
6	Review and windup		Theory, Quiz 2
7	Context-Free-Grammars and Parsing The Parsing Process Context-Free Grammars		Theory
8	Parse Trees and Abstract Syntax Trees Ambiguity Extended Notations: EBNF and Syntax Diagrams		Theory, Assignment 2 (design parser)
9	MID-TERM EXAMS		1.5 Hrs Exam
10	Syntax-Analysis Top-Down Parsing by Recursive-Descent		Theory, Quiz 3
11	LL(1) Parsing First and Follow Sets		Theory
12	A Recursive-Descent Parser for the TINY Language Error Recovery in Top-Down Parsers		Theory, Assignment 3 (coding a parser)
13	Bottom-up parsing: a quick review		Theory
14	Semantic-Analysis Attributes and Attribute Grammars Algorithms for Attribute Computation		Theory, Quiz 4
15	The Symbol Table Data Types and Type Checking		Theory
16	Code Generation		Theory
17	Code Generation		Theory, Assignment 4 (code generation)
18	Final Exam		2 Hr Exam
<b>Resources</b>			
<b>Required Course Material(s)/Reading(s)/Text Book(s)</b>			
1	Kenneth C.Louden Compiler Construction: Principles and Practice		
<b>Recommended Course Material(s)/Reading(s)/Other</b>			
2	Compilers Principles, Techniques and Tools by Alfred V. Aho		
<b>Assessment Methods</b>			
<b>Semester Activities/ Studies</b>	<b>No</b>	<b>WEIGHT in %</b>	
Mid-Term	1	20	
Attendance	0	0	
Quiz	4	10	
Assignment(s)	4	10	
Project	0	0	
Field Studies(Technical Visits)	0	0	
Presentation/Seminar	0	0	
Practice(Laboratory, Virtual Court,Studio Studies etc.	4	10	
Others(Placement/Intership etc.)	0	0	
Final Examination	1	50	
<b>TOTAL</b>	<b>12</b>	<b>100</b>	
Contribution of Semester Activities/Studies to the Final Grade		50	
Contribution of final Examination/final Project/Dissertation to the final Grade		50	
<b>TOTAL</b>		<b>100</b>	