Military Institute of Science & Technology

Mirpur Cantonment, Dhaka

Department of Computer Science & Engineering

COURSE OUTLINE

Subject
Title: Compiler
Code: CSE-303
Credit Hour: 3.00
Contact Hour: 3.00
Level-3, Term-I

Instructor
Name: Lec Moumita Sarker
E-mail: moumita.sarker123456@gmail.com
Name: Lec Nabila Shahnaz Khan
Email: nabilakhan.mist@gmail.com

1.0 Course Objectives:

1.1 To introduce the theory and tools that can be employed in order to perform syntax-directed translation of a high-level programming language into an executable code.

1.2 To understand the role of compilers in programming languages and various stages in compilation process.

1.3 To provide knowledge on designing scanners and parsers using tools.

1.4 To provide deeper insights into the more advanced semantics aspects of programming languages, such as recursion, dynamic memory allocation, types.

2.0 Course Outcomes:

2.1 Describe the role of compilers in programming languages and the stages in compilation process.

2.2 Separate the lexical, syntactic and semantic analysis into meaningful phases for a compiler to undertake language translation and specify and analyze the lexical, syntactic and semantic structures of advanced language features.

2.3 Design scanners and parsers using tools, and build abstract syntax trees in connection with this.

2.4 Describe the issues involved in machine code generation.

3.0 Text Books:

3.1 Compilers Principles, Techniques and Tools, Second Edition-- Alfred V. Aho, Jeffrey D. Ullman

4.0 Reference Books:

4.1 Engineering a Compiler, Second Edition-- Linda Torczon and Keith Cooper

5.0 Distribution of Marks:

Description	Percentage	Marks
Class Attendance	05%	15
Midterm Examination	15%	45
Class Test	20%	60
Final Examination (Section A & Section B)	60%	180
Total	**100%**	**300**

6.0 Mapping of Course Outcomes (CO) and Program Outcomes:

Course Outcomes(CO) of the Course	Program Outcome (PO)											
	1	2	3	4	5	6	7	8	9	10	11	12
Describe the role and purpose of compilers in programming languages.	√											
Discuss the stages in compilation process.	√			√		√						
Separate the lexical, syntactic and semantic analysis into meaningful phases for a compiler to undertake language translation and specify and analyze the lexical, syntactic and semantic structures of advanced language features.		√				√						
Design scanners and parsers using tools, and build abstract syntax trees in connection with this.			√									
Describe the issues involved in machine code generation.		√										

7.0 Distribution (Planning) of the Course Contents:

Week	Lecture	Topics	Chapter	Remarks
1	Lec 1	Introduction to Compilers, Language Processors	1.1	Aho and Ullman
	Lec 2	The Structure of a Compiler	1.2.1-1.2.4	
	Lec 3		1.2.5-1.2.9	
2	Lec 4	The Role of the Lexical Analyzer, Input Buffering	3.1.1-3.1.3	
	Lec 5		3.1.4, 3.2.1-3.2.2	
	Lec 6	Recognition of Tokens, Transition Diagram	3.4, 3.4.1	
3	Lec 7	Recognition of Reserved Words and Identifiers	3.4.2, 3.4.3	
	Lec 8	Architecture of a Transition-Diagram-Based Lexical Analyzer	3.4.4	
	Lec 9	The Lexical-Analyzer Generator Lex	3.5.1-3.5.3	
Class Test -1				
4	Lec 10	Top-Down Parsing	2.4.1	
	Lec 11	Predictive Parsing	2.4.2, 2.4.3	
	Lec 12			
5	Lec 13	Designing a Predictive Parser, Left Recursion	2.4.4, 2.4.5	
	Lec 14	The Role of the Parser, Representative Grammars	4.1.1, 4.1.2	
	Lec 15	Syntax Error Handling, Writing a Grammer	4.1.3, 4.3	
6	Lec 16	Elimination of Left Recursion	4.3.3	
	Lec 17	Left Factoring	4.3.4	
	Lec 18	Top-Down Parsing, First and Follow	4.4, 4.4.2	
Class Test – 2				
7	Lec 19	LL(1) Grammars, Construction of Predictive Parsing Table	4.4.3	
	Lec 20	Nonrecursive Predictive Parsing	4.4.4	
	Lec 21	Parsers Generators	4.9.1, 4.9.3	
8	Lec 22	Syntax-Directed Definitions, Inherited and Synthesized Attribute	5.1, 5.1.1	
	Lec 23	Evaluating an SDD at the Nodes of a Parse Tree	5.1.2	
	Lec 24	Dependency Graph	5.2.1	
9	Lec 25	Ordering the Evaluation of Attributes, S-Attributed Definitions	5.2.2, 5.2.3	
	Lec 26	L-Attributed Definitions, Semantic Rules with Controlled Side Effect	5.2.4, 5.2.5	
	Lec 27	Applications of Syntax Directed Translation	5.3.1, 5.3.2	
Midterm Examination				
10	Lec 28	Variants of Syntax Tree, Directed Acyclic Graphs for Expressions	6.1, 6.1.1	
	Lec 29	The Value Number Method for Constructing DAG's	6.1.2	
	Lec 30	Three-Address Code, Addresses and Instructions	6.2, 6.2.1	
11	Lec 31	Quadruples, Triples	6.2.2, 6.2.3	
	Lec 32	Static Single-Assignment Form, Types and Declarations	6.2.4, 6.3.1-6.3.2	
	Lec 33		6.3.3-6.3.5	

12	Lec 34	Storage Organization, Static VS Dynamic Storage Allocation	7.1, 7.1.1	
	Lec 35	Stack Allocation of Space, Activation Trees	7.2, 7.2.1	
	Lec 36	Activation Records	7.2.2	
Class Test – 3				
13	Lec 37	Issues in the Design of a Code Generator	8.1,8.1.1- 8.1.2	
	Lec 38		8.1.3	
	Lec 39	The Target Language	8.2, 8.2.1- 8.2.2	
14	Lec 40	Addresses in the Target Code, Static Allocation	8.3, 8.3.1	
	Lec 41	Optimization of Basic Blocks	8.5.1-8.5.3, 8.55	
	Lec 42	Peephole Optimization	8.7.1- 8.7.3	

Date: 04 February, 2019

Lec Moumita Sarker
Lec Nabila Shahnaz Khan