

Maulana Abul Kalam Azad University of Technology, West Bengal*(Formerly West Bengal University of Technology)***Syllabus for B. Tech in Computer Science & Engineering**

(Applicable from the academic session 2018-2019)

Compiler Design**Code: PCC-CS501****Contact: 3L**

Name of the Course:	Compiler Design	
Course Code: PCC-CS501	Semester:V	
Duration:6 months	Maximum Marks:100	
Teaching Scheme	Examination Scheme	
Theory:3 hrs./week	Mid Semester exam: 15	
Tutorial: NIL	Assignment and Quiz: 10 marks	
	Attendance: 5 marks	
Practical: NIL	End Semester Exam:70 Marks	
Credit Points:	3	
Objective:		
1	To understand and list the different stages in the process of compilation.	
2	Identify different methods of lexical analysis	
3	Design top-down and bottom-up parsers	
4	Identify synthesized and inherited attributes	
5	Develop syntax directed translation schemes	
6	Develop algorithms to generate code for a target machine	

Unit	Content	Hrs/Unit	Marks/Unit
1	Introduction to Compiling [3L] Compilers, Analysis of the source program, The phases of the compiler, Cousins of the compiler.	3	
2	Lexical Analysis [6L] The role of the lexical analyzer, Tokens, Patterns, Lexemes, Input buffering, Specifications of a token, Recognition of a tokens, Finite automata, From a regular expression to an NFA, From a regular expression to NFA, From a regular expression to DFA, Design of a lexical analyzer generator (Lex).	6	
3	Syntax Analysis [9L] The role of a parser, Context free grammars, Writing a grammar, Top down Parsing, Non-recursive Predictive parsing (LL), Bottom up parsing, Handles, Viable prefixes, Operator precedence parsing, LR parsers (SLR, LALR), Parser generators (YACC). Error Recovery strategies for different parsing techniques.	9	
4	Syntax directed translation [5L] Syntax director definitions, Construction of syntax trees, Bottom-up evaluation of S attributed definitions, L attributed definitions, Bottom-up evaluation of inherited attributes.	5	
5	Type checking [4L] Type systems, Specification of a simple type checker, Equivalence of type expressions, Type conversions	4	
6	Run time environments [5L] Source language issues (Activation trees, Control stack, scope of declaration, Binding of names), Storage organization (Subdivision of run-time memory, Activation records), Storage allocation strategies, Parameter passing (call by value, call by reference, copy restore, call by name), Symbol tables, dynamic storage allocation techniques.	5	
7	Intermediate code generation [4L] Intermediate languages, Graphical representation, Three-address code, Implementation of three address statements (Quadruples, Triples, Indirect triples).	4	

8	Code optimization [5L] Introduction, Basic blocks & flow graphs, Transformation of basic blocks, Dag representation of basic blocks, The principle sources of optimization, Loops in flow graph, Peephole optimization.	5	
9	Code generations [4L] Issues in the design of code generator, a simple code generator, Register allocation & assignment.	4	

Text book and Reference books:

1. Aho, Sethi, Ullman - “Compiler Principles, Techniques and Tools” - Pearson Education.
2. Holub - “Compiler Design in C” - PHI.

Course Outcomes:

On completion of the course students will be able to

1. Understand given grammar specification develop the lexical analyser
2. Design a given parser specification design top-down and bottom-up parsers
3. Develop syntax directed translation schemes
4. Develop algorithms to generate code for a target machine