MCA416: Compiler Design

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/Week	Class Test -12Marks
Tutorials: 1 hr/Week	
	Teachers Assessment - 6Marks
	Attendance – 12 Marks
Credits: 4	End Semester Exam – 70 marks

Prerequisite: - Theory of Formal Language (Automata), Basic Concepts of C Language, Discrete Mathematics.

Course Objectives:

1. To learn the process of translating a modern high-level language to executable code.
2. To learn about Automata Theory of basic Concepts.
3. To develop concepts of parsing.
4. To Analyze concept of Chomsky, Syntax tree..

Head
Department of Computer Applications
Faculty of Computer Applications
Invertis University, Bareilly (UP)

Registrar
Invertis University
Bareilly

Dean Academ
Faculty of Computer A
Invertis University, B

5. To learn run time enviournment concepts, run time concepts.
6. To understand concepts of code optimization techniques.

Detailed Syllabus

UNIT I

Introduction to Compiler: Compilers, Analysis of the source program, The phases of the compiler, Major data structures in a Compiler, Issues in a Compiler Structure, Bootstrapping and Porting.

UNIT II

Formal Language and Regular Expressions: Languages, Definition Languages regular expressions, Finite Automata – DFA, NFA, Conversion of regular expression to NFA, NFA to DFA. Applications of Finite Automata to Compiler Construction- lexical analysis, Construction of lexical analyses using LEX tool, Phases of Compilation and A simple One-Pass Compiler.

UNIT III

Context Free grammars and parsing: Context free grammars, derivation, parse trees, ambiguity, Application CFG in compilation-Preprocessing steps in Parsing, LL(1) parsing, Bottom up parsing handle pruning LR Grammar Parsing, LALR parsing, parsing ambiguous grammars, YACC programming specification.

UNIT IV

Semantics: Syntax directed translation, S-attributed and L-attributed grammars, Intermediate code – abstract syntax tree, translation of simple statements and control flow statements, Context Sensitive features – Chomsky hierarchy of languages and recognizers, Type checking, type conversions, equivalence of type expressions, overloading of functions and operations.

UNIT V

Run-time Environments: Memory organization during program execution, Fully static run-time environment, Stack-based run-time environments, Dynamic memory, Parameter passing mechanism, Run-time environment for Tiny language.

UNIT VI

Code Generation: Intermediate code and data structures for code generation, Basic code generation techniques, Code generation of Control statements and Logical expressions, Code generation of Procedure and Function calls, Code generation for a tiny language, A survey of code optimization techniques..

Text and Reference Books:

1. "Compiler Principles, Techniques and Tools", Aho, Sethi, Ullman, Pearson Education,2007.
2. "Introduction to Automata Theory, Languages and Computation" ,Hopcroft, RajeevMotwani and Ullman, Addition Wesley, 2006.
3. "Compiler Construction – Principle and Practice", Kenneth C. Louden, Thomson 2007.
4. "Introduction to Theory of computation", Sipser, Thomson, 2009.

Course Outcomes:

After completing the course, students will be able to:

1. Understand concepts of Phase of Compiler.
2. Understand the concept of Automata Theory.
3. Understand the Concept of Parsing and YACC compiler.
4. Understand the concept Run time enviournment and tiny Language Concepts.
5. Understand the concepts of code generation.
6. Understand the connectivity of syntax tree and Chomsky hierarchical tree.

Head
Department of Computer Applications
Faculty of Computer Applications
Invertis University, Bareilly (UP)

Registrar
Invertis University
Bareilly

Dean Academics
Faculty of Computer Applic
Invertis University, Bareilly