3/4 B.Tech. SIXTH SEMESTER

ELECTIVE – I

EM6T4B COMPILER DESIGN Credits: 3

Lecture: 3 periods/week **Internal assessment : 30 marks**
Tutorial: 1 period /week **Semester end examination: 70 marks**

Course Objective:

- To enrich the knowledge in various phases of compiler ant its use, code optimization techniques, machine code generation, and use of symbol tables and data flow analysis.

Learning Outcomes:

At the end of this course the Students will be able to

- Learn various phases of the compiling process.
- To understand lexical analysis, grammars and parsing, type-checking, intermediate Representations, static analysis.
- To understand the common optimizations techniques, instruction selection, register allocation, code generation, and runtime organization

UNIT – I
Overview of Compilation: Phases of Compilation – Lexical Analysis, Regular Grammar and regular expression for common programming language features, pass and Phases of translation, interpretation, bootstrapping, data structures in compilation – LEX lexical analyzer generator.

UNIT – II
Top down Parsing: Context free grammars, Top down parsing – Backtracking, LL(1), recursive descent parsing, Predictive parsing, Pre-processing steps required for predictive parsing.

UNIT – III
Bottom up parsing: Shift Reduce parsing, LR and LALR parsing, Error recovery in parsing , handling ambiguous grammar, YACC – automatic parser generator.

UNIT – IV
Semantic analysis: Intermediate forms of source Programs – abstract syntax tree, polish notation and three address codes. Attributed grammars, Syntax directed translation, Conversion of popular Programming languages, language constructs into Intermediate code forms, Type checker.

UNIT – V
Symbol Tables: Symbol table format, organization for block structures languages, hashing, tree Structures representation of scope information. Block structures and non block structure storage allocation: static, Runtime stack and heap storage allocation, storage allocation for arrays, strings and records.

UNIT – VI
Code optimization: Consideration for Optimization, Scope of Optimization, local optimization, loop optimization, frequency reduction, folding, DAG representation.

UNIT – VII
Data flow analysis: Flow graph, data flow equation, global optimization, redundant sub expression elimination, Induction variable elements, Live variable analysis, Copy propagation.

UNIT – VIII
Object code generation: Object code forms, machine dependent code optimization, register allocation and assignment generic code generation algorithms, DAG for register allocation.

Learning resources

Text books:

1. Principles of compiler design -A.V. Aho. J.D.Ullman; Pearson Education.
2. Modern Compiler Implementation in C- Andrew N. Appel, Cambridge University Press.

Reference books:

1. Modern Compiler Design- Dick Grune, Henry E. Bal, Cariel T. H. Jacobs, Wiley dreamtech.
2. Engineering a Compiler-Cooper & Linda, Elsevier.
3. Compiler Construction, Louden, Thomson.