COMPILER DESIGN

Course Code: 15CS1103

L	T	P	C
3	0	0	3

Pre-requisites:

Formal Languages and Automata Theory

Course Outcomes :

At the end of the Course, the Student will be able to:

CO 1 Summarize working of a compiler.

CO 2 Discuss role of lexical and syntax analysers in a compiler.

CO 3 Differentiate top down and bottom up parsers.

CO 4 Explain semantic analysis.

CO 5 Explain intermediate code generation, code optimization and code generation.

UNIT-I (10 Lectures)

INTRODUCTION TO COMPILING:

Overview of Compilers, Analysis of the Source Program, the Phasesof a Complier, Pre-Processors, Assemblers, Two Pass Assembly,Loaders and Link-Editors, Bootstrapping, The Grouping of Phases, Compiler Construction Tools.

UNIT-II (10 Lectures)

LEXICAL ANALYSIS:

The Role of the Lexical Analyzer, Strings and Languages, Operationson Languages, Regular Expressions, Regular Definitions, NotationalShorthands, Recognition of Tokens, A Language for specifyingLexical Analyzers(LEX).

SYNTAX ANALYSIS:

The Role of the Parser, Context-free Grammars, Writing a Grammar.

UNIT-III (10 Lectures)

TOP-DOWN PARSING:

Recursive Descent Parsing, Predictive Parsers, Non-RecursivePredictive Parsing, First and Follow, Construction of Predictive ParsingTables, LL(1) Grammars, Error Recovery in Predictive Parsing.

BOTTOM-UP PARSING:

Handles, Handle Pruning, Stack Implementation, Operator-PrecedenceParsing, LR Parsers-SLR, Canonical LR, LALR.Using AmbiguousGrammars, Parser Generator (YACC).

SYNTAX-DIRECTED TRANSLATION:

Syntax-Directed Definition, Construction of Syntax Trees, S-Attributed Definitions, L-Attributed Definitions.

UNIT-IV (10 Lectures)

SEMANTIC ANALYSIS:

Type Systems, Specification of a Type Checker, Equivalence of typeexpressions,Type Conversions, Overloading of functions andoperators, Polymorphic functions, Algorithm for Unification.

RUN-TIME ENVIRONMENT:

Source Language Issues, Storage Organization, Storage Allocation Strategies, Blocks, Access Links, Procedure Parameters, Displays, Parameter Passing, Symbol Tables.

UNIT-V (10 Lectures)

INTERMEDIATE CODE GENERATION:

Intermediate Languages-Graphical Representations, Three AddressCode, Implementations, Boolean Expressions.

CODE OPTIMIZATION:

Introduction, Principle sources of Optimization, Optimization of BasicBlocks.

CODE GENERATION:

Issues, the Target Machine, Run-Time Storage Management, BasicBlocks and Flow graphs, Loops in Flow graphs, Data-Flow Analysis.

TEXT BOOK:

Alfred V Aho, Ravi Sethi, Jeffrey D.Ullman, "Compilers-Principles Techniques and Tools", 2nd Edition, PearsonEducation, 2008.

REFERENCES:

1. Raghavan, "Principles of Compiler Design", 2nd Edition,TMH, 2011.
2. Kenneth C.Louden, "Compiler Construction-Principles and Practice", 2nd Edition, Cengage, 2010.
3. Cooper and Linda, "Engineering a Compiler", 4th Edition,Elsevier, 2008.

WEB REFERENCES:

i. http://nptel.ac.in/courses/106108052/1

ii. http://nptel.ac.in/courses/106108052/2
http://nptel.ac.in/courses/106108052/3