

CORE COURSE XXVIII – COMPILER DESIGN

Unit I : Introduction

Compilers – Analysis of the source program – Phases of a compiler – Cousins of the Compiler – Grouping of Phases – Compiler construction tools – Lexical Analysis – Role of Lexical Analyzer – Input Buffering – Specification of Tokens

Unit II: Basic Data Structures

Role of the parser, Writing Grammars – Context – Free Grammars – Top Down parsing – Recursive Descent parsing – Predictive parsing – bottom –up parsing – shift Reduce Parsing – Operator Precedent Parsing – LR Parsers – SLR Parser – Canonical LR Parser – LALR Parser

Unit III: Advanced Data Structures

Intermediate Languages – Declarations – Assignment Statements – Boolean Expressions – Case Statements – Back patching – procedure calls

Unit IV: Sorting & Searching Techniques

Issues in the design of code generator – The target machine – Runtime Storage management – Basic Blocks and Flow Graphs – Next use Information – A simple Code generator – DAG representation of Basic Blocks – Peephole optimization

Unit V: Files

Introduction – Principal Sources of Optimization – Optimization of basic Blocks – Introduction to Global Data Flow Analysis – Runtime Environments – Source Language issues – Storage Organization – Storage Allocation strategies – Access to non-local names – Parameter Passing.

Text Book(s)

1. Alfred Aho, Ravi Sethi, Jeffrey D.Ullman, “Compilers – Principles, Techniques and Tools”, Pearson Education Asia, 2003

References

1. Henk Alblas and Albert Nymeyer, “Practice and Principles of Compiler Building with C”, PHI, 2001
2. Kenneth C. Loudon, ‘Compiler Construction’: Principales and Practices», Thompson Learning, 2003.