

**III Year – I Semester**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>4</b>	<b>0</b>	<b>0</b>	<b>3</b>

## **COMPILER DESIGN**

### **OBJECTIVES:**

- Understand the basic concept of compiler design, and its different phases which will be helpful to construct new tools like LEX, YACC, etc.

### **UNIT – I**

Introduction Language Processing, Structure of a compiler the evaluation of Programming language, The Science of building a Compiler application of Compiler Technology. Programming Language Basics.

Lexical Analysis-: The role of lexical analysis buffering, specification of tokens. Recognitions of tokens the lexical analyzer generator lexical

### **UNIT –II**

Syntax Analysis -: The Role of a parser, Context free Grammars Writing A grammar, top down parsing bottom up parsing Introduction to Lr Parser.

### **UNIT –III**

More Powerful LR parser (LR1, LALR) Using Armigers Grammars Equal Recovery in Lr parser Syntax Directed Transactions Definition, Evolution order of SDTS Application of SDTS. Syntax Directed Translation Schemes.

### **UNIT – IV**

Intermediated Code: Generation Variants of Syntax trees 3 Address code, Types and Deceleration, Translation of Expressions, Type Checking. Canted Flow Back patching?

### **UNIT – V**

Runtime Environments, Stack allocation of space, access to Non Local date on the stack Heap Management code generation – Issues in design of code generation the target Language Address in the target code Basic blocks and Flow graphs. A Simple Code generation.

### **UNIT –VI**

Machine Independent Optimization. The principle sources of Optimization peep hole Optimization, Introduction to Date flow Analysis.

**OUTCOMES:**

- Acquire knowledge in different phases and passes of Compiler, and specifying different types of tokens by lexical analyzer, and also able to use the Compiler tools like LEX, YACC, etc.
- Parser and its types i.e. Top-down and Bottom-up parsers.
- Construction of LL, SLR, CLR and LALR parse table.
- Syntax directed translation, synthesized and inherited attributes.
- Techniques for code optimization.

**TEXT BOOKS:**

1. Compilers, Principles Techniques and Tools. Alfred V Aho, Monical S. Lam, Ravi Sethi Jeffery D. Ullman, 2<sup>nd</sup> edition, pearson, 2007
2. Compiler Design K. Muneeswaran, OXFORD
3. Principles of compiler design, 2<sup>nd</sup> edition, Nandhini Prasad, Elsevier.

**REFERENCE BOOKS:**

1. Compiler Construction, Principles and practice, Kenneth C Loudon, CENGAGE
2. Implementations of Compiler, A New approach to Compilers including the algebraic methods, Yunlinsu, SPRINGER