|  |  |  |
| --- | --- | --- |
| 1 | Course Code | **CSE 303/CSP303** |
| 2 | Course Title | **Compiler Design** |
| 3 | Credits | **5** |
| 4 | Contact Hours | **3-1-2** |
| 5 | Course Objective | 1. To provide students with an overview of the issues that arise in Compiler construction as well as to throw light upon the significant theoretical developments and tools that are deep rooted into computer science. 2. To introduce the major phases of Compiler construction and also its theoretical aspects including regular expressions, context-free grammars, Finite Automata etc. |
| 6 | Course Outcomes | After the successful completion of this course, students will be able to :   1. Employ formal attributed grammars for specifying the syntax and semantics of programming languages. 2. Apply regular patterns and grammars. (ABET program outcomes a and j) 3. Comprehend the working knowledge of the major phases of compilation, particularly lexical analysis, parsing, semantic analysis, and code generation. 4. Implement parsing and translation techniques for automation of computing tasks. 5. Design and write a complex programming project. (ABET program outcomes c and k) |
| **7** | **Prerequisite** |  |
| **8** | **Course Contents** | |
| 8.01 | Unit A | **Introduction** |
| 8.02 | Unit A Topic 1 | Introduction to Compiler, Phases and passes, Bootstrapping, Cross-Compiler |
| 8.03 | Unit A Topic 2 | Finite state machines and regular expressions and their applications to lexical analysis |
| 8.04 | Unit A Topic 3 | lexical-analyzer generator, LEX-compiler Lexical Phase errors |
| 8.05 | Unit B | **Parsing Techniques** |
| 8.06 | Unit B Topic 1 | The syntactic specification of programming languages: Context free grammars, derivation and parse trees. |
| 8.07 | Unit B Topic 2 | Basic Parsing Techniques: Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers.  Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables |
| 8.08 | Unit B Topic 3 | Constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars. YACC. Syntactic phase errors and semantic errors. |
| 8.09 | Unit C | **Syntax Directed Translations And Intermediate Code Generation** |
| 8.10 | Unit C Topic 1 | Syntax directed definition, Construction of syntax trees, syntax directed translation scheme |
| 8.11 | Unit C Topic 2 | Variants of Syntax Trees, Three Address Codes |
| 8.12 | Unit C Topic 3 | Translation of Expression, Type Checking and control flow. |
| 8.13 | Unit D | **Symbol table** |
| 8.14 | Unit D Topic 1 | Data structure for symbols tables, representing scope information. |
| 8.15 | Unit D Topic 2 | Run-Time Administration: Implementation of simple stack allocation scheme, storage allocation in block structured language. |
| 8.16 | Unit D Topic 3 | Run Time Storage Management |
| 8.17 | Unit E | **Code Generation And Optimization** |
| 8.18 | Unit E Topic 1 | Sources of Optimization of basic blocks and flow graphs |
| 8.19 | Unit E Topic 2 | Basic Blocks, Flow graphs, DAG |
| 8.20 | Unit E Topic 3 | Global Data Flow Analysis |
| 10 | Reading Content | |
| 9.1 | Text book\* | 1. Aho, Sethi, Ulman, compilers Principles, Techniques, and Tools, Pearson Education, 2003 |
| 9.2 | other references | 1. Lauden, Principles of Compiler Construction. 2. D. M. *Dhamdhere Compiler* Construction--Principles and Practice, Macmillan India, 3. Internet as a resource for reference |