|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 13CS3204 | - | COMPILER DESIGN | | | | | | | | |
|  |  | |  | | | | |
| Hours / Week | : | 4 | |  | Sessional Marks | : | 40 |
| Credits | : | 4 | |  | End Examination Marks | : | 60 |

|  |
| --- |
| **UNIT – I** |
| **Introduction to Compiling**: Compilers, Analysis of the Source program. Phases of a compiler, Cousins of the Compiler. Grouping of phases, Compiler construction tools. Lexical Analysis: Role of the analyzer. Input buffering, Specification of tokens, Recognition of tokens, A language for Specifying Lexical analyzer. |
|  |
| **UNIT – II** |
| **Syntax Analysis**: Role of the parser, Context-free grammars, Writing a grammar, Top-down parsing, Bottom-up parsing, Operator-precedence parsing, LR parsers. Using ambiguous grammars, Parser generators. |
|  |
| **UNIT – III** |
| **Syntax Directed Translation**: Syntax-directed definitions, Construction of syntax trees, Bottom-up evaluation of S-attributed definitions. L-attributed definitions. Top-down translations. Bottom-up evaluation of inherited attributes. **Type Checking**: Type systems, Specification of simple type checker. Equivalence of type expressions, Type conversions, Overloading of functions and operators, Polymorphic functions. |
|  |
| **UNIT – IV** |
| **Run-Time Environments**: Source Language issues, Storage organization, Storage-allocation strategies. Access to non-local names. Symbol tables, Language facilities for dynamic storage allocation. Dynamic storage allocation techniques. **Intermediate Code generation**: Intermediate languages. Declarations, Assignment statements. |
|  |
| **UNIT - V** |
| **Code Generation**: Issues in the Design of a code generator, The target machine, Run-time storage management, Basic blocks and flow graphs, Next-use information. A simple code generator, Register allocation and assignment. **Code Optimization**: Introduction. The principle source of optimization. |
|  |
|  |
| TEXT BOOKS |
| 1. Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, Compilers-Principles, Techniques and Tools, Pearson Education,2004 |
|  |
| REFERENCE BOOKS |
| 1. Alfred V. Aho, Jeffrey D. Ullman, Principles of Compiler Design, Narosa publications. 2. J.P.Benne, Introduction to Compiling Techniques, 2nd Edition, Tata McGraw-Hill. |