|  |  |
| --- | --- |
| ***Department:*** *Computer Science and Engineering* | ***Course Type****:* **Programme Core** |
| ***Course Title****: Compiler Design* | ***Course Code****:* 14CS73 |
| *L-T-P: 4-0-0* | ***Credits****: 3* |
| ***Total Contact Hours****: 39 hrs* | ***Duration of SEE****: 3 hrs* |
| ***SEE Marks****:50* | ***CIE Marks****: 50* |

**Course Description:**

This course provides an introduction to Compiler Design and aims at solving problems in designing compilers, besides giving ideas on optimization techniques. Tools for compiler construction, green compilation techniques and also compiling for power optimization are the advanced topics dealt in this course.

Prerequisites: Knowledge of assembly level programming, Formal Languages and Automata Theory is essential.

**Course Objectives:**

* To identify features of phases of compiler, learn the design aspects and tools.
* To design parser automaton, construct parser table and do the program translation
* To apply various optimization techniques on programs and to extend it for green compilation and power optimization.

**Course Contents:**

|  |  |
| --- | --- |
| **UNIT – I** | **8 Hours** |

**Introduction to compilers**: Language processors, the structure of a compiler, Lexical Analysis, Syntax analysis, Semantic analysis, Intermediate Code generation, Optimization, Code generation, Compiler construction tools, The Science of building a compiler, Applications of compiler technology, **Lexical Analysis**: The role of lexical analyzer, A simple approach to the design of lexical analyzer, Lex tool.

|  |  |
| --- | --- |
| **UNIT – II** | **8 Hours** |

**Syntax Analysis & Parsing**: Top down parsing. Recursive descent parsing, Computation of FIRST & FOLLOW sets LL(1) grammar,Bottom up parsing- shift reduce parser & conflicts, Simple LR, The canonical collection of LR(0) items, LR Parsing algorithm, Constructing SLR parsing tables, LR(1) items, Constructing canonical LR(1) parsing tables, Constructing LALR parsing tables, efficient construction of LALR parsing table, Using ambiguous grammars – precedence, associativity, dangling else problem, Parser generators, YACC.

|  |  |
| --- | --- |
| **UNIT – III** | **8 Hours** |

**Syntax Directed Translation& Intermediate code generation**: Syntax directed translation schemes & implementations, Intermediate code, Syntax trees, DAG for expressions, Three address code, quadruples, triples, Translation of expressions, Control flow, Backpatching, Switch statements, Intermediate code for procedures.

|  |  |
| --- | --- |
| **UNIT – IV** | **8 Hours** |

**Runtime Environment and Code generation**: Static VS Dynamic storage, stack allocation scheme, Design issues in code generation, Basic blocks and flow graphs; Optimization in the basic blocks; implementation of simple code generator, code generation algorithm; Peep hole optimization; register allocation and assignments; optimal code generation for expressions.

|  |  |
| --- | --- |
| **UNIT – V** | **7 Hours** |

**Machine Independent Optimization:** The sources of optimization, common subexpression, copy propagation, dead code elimination, code motion; Data flow analysis, reaching definitions, live variables, constant propagation, partial redundancy elimination; Loops in flow graph, depth first, back edges & reducibility; region based analysis and the algorithm; Symbolic analysis; Optimization for power management, Green compilers, use of tools.

**Text Book**

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers : Principles, Techniques and Tools, Second Edition, Pearson Education, 2007.

**Reference Books**

1. Allen I.Holub, Compiler Design in C, PHI, 1990.
2. Jean Paul Tremblay and Paul G. Sorenson, The Theory and Practice of Compiler Writing,BS Publications, 2008.
3. K.C. Louden, Compiler Construction:Principles and Practice,Cengage Publications,2002.
4. Wissam Chedid, Chansu Yu, and Ben Lee, Power Analysis and Optimization Techniques for Energy Efficient Computer Systems, Electrical Engineering &Computer Science Faculty Publications118, 2005, <http://engagedscholarship.csuohio.edu/enece_facpub/118>
5. Ulrich Kremer, Low Power/Energy Compiler Optimizations, Dept. of Computer Science, Rutgers University, 2005, online <https://www.cs.rutgers.edu/~uli/CRC04.pdf> .

**Other Materials:**

1. Leland L. Beck, System Software, 3rd Edition, Addison Wesley, 1997.

2. <https://nptel.ac.in/course.php>

**Self Study Component: Loaders And Linkers :** Basic Loader Functions , Design of an Absolute Loader, A Simple Loader, Machine Dependent Loader Features , Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader; **Macro Processor:**Basic Macro Processor Functions, Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures.

**Teaching Methods**

* Lecture using Black board and chalk
* Power point presentations
* Exercises on Problem solving
* Self study/ Course Project Assignment

**Assessment Methods**

* Three internals – 30 Marks each will be conducted and the Average of best of the two will be considered.
* Rubrics for the evaluation of Self-study/Course Project component for 20 marks.
* Final examination will be conducted for 100 marks and evaluated for 50 Marks.

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

|  |  |
| --- | --- |
| **CO** | **Description** |
| **CO 1** | Explain Compilers& its phases, lexical analysis and demonstrate use of LEX tool. |
| **CO 2** | Compute LR(0), LR(1) and LALR sets of items,construct parse table for a given grammar and show the use of YACC tool. |
| **CO 3** | Perform syntax directed translation and generate intermediate code. |
| **CO 4** | Implement runtime procedures and code generation managing optimization in code generation phase. |
| CO 5 | Perform machine independent optimizations and develop optimizations for power management. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mapping of Course outcomes (COs) to Program outcomes (POs)** | | | | | | | | | | | | | | | |
| **Course Outcomes mapping to program outcomes** | | | | | | | | | | | | | **Program Specific Outcomes (PSOs)** | | |
| **POs/ COs** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **PSO1** | **PSO2** | **PSO3** |
| CO1 | 2 | 2 | 3 |  | 3 | 1 | 1 |  |  |  |  |  | 3 | 2 |  |
| CO2 | 3 | 3 | 3 |  | 3 |  | 1 |  |  |  |  |  | 3 | 3 |  |
| CO3 | 2 | 3 | 3 |  | 2 |  | 1 |  |  |  |  |  | 2 | 3 |  |
| CO4 | 2 | 2 | 3 |  | 2 | 2 | 3 |  |  |  |  |  | 3 | 3 |  |
| CO5 | 3 | 2 | 3 |  | 3 | 3 | 3 |  |  |  |  |  | 3 | 2 |  |
| Correlation Level | 2 | 2 | 3 |  | 3 | 2 | 2 |  |  |  |  |  | 3 | 3 |  |

\*3: Strong, 2: Medium, 1: Weak \*\*3: Highly related 2: Supportive