**COURSE PLAN**

|  |  |
| --- | --- |
| Target | 50% (marks) |
| Level-1 | 40% (population) |
| Level-2 | 50% (population) |
| Level-3 | 60% (population) |

1. **Method of Evaluation**

|  |  |
| --- | --- |
| **UG** | **PG** |
| Quizzes/Tests, Assignments (30%) | Quizes/Tests, Assignments, seminar (50%) |
| Mid Examination (20%) | End semester (50%) |
| End examination (50%) |  |

1. **Passing Criteria**

|  |  |  |
| --- | --- | --- |
| **Scale** | **PG** | **UG** |
| **Out of 10 point scale** | SGPA – “6.00” in each semester  CGPA – “6.00”  Min. Individual Course Grade  –  “C”  Course Grade  Point –  “4.0” | SGPA – “5.0” in each semester  CGPA – “5.0”  Min. Individual Course Grade  –  “C”  Course Grade  Point –  “4.0” |

\*for PG, passing marks are 40/100 in a paper

\*for UG, passing marks are 35/100 in a paper

1. **Pedagogy**
2. Problem Solving
3. Class Test/Quiz
4. Assignments
5. Digital and analog Presentations
6. Concept diary (needs to be maintained by students-short and concise notes which include course concepts that he/she has understood.)
7. **References:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Text Books** | **Web resources** | **Journals** | **Reference books** |
| 1. Alfred V. Aho, Ravi Sethi Jeffrey D. Ullman, “Compilers- Principles, Techniques, and Tools”, 2nd Edition, Pearson Education Asia 2. Robin Hunter, “The Essence of Compiler”, 2nd Edition, Pearson Publication | 1. <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-035-computer-language-engineering-spring-2010/lecture-notes/> 2. <http://nptel.ac.in/courses/106108052/1> |  | 1. Randy Allen, Ken Kennedy, “Optimizing Compilers for Modern Architectures: A Dependence-based Approach”, Morgan Kaufmann Publishers, 2002. 2. Steven S. Muchnick, “Advanced Compiler Design and Implementation, “Morgan Kaufmann Publishers - Elsevier Science, India, Indian Reprint 2003. 3. Keith D Cooper and Linda Torczon, “Engineering a Compiler”, Morgan Kaufmann Publishers Elsevier Science, 2004. 4. Charles N. Fischer, Richard. J. LeBlanc, “Crafting a Compiler with C”, Pearson Education, 2008. |

**GUIDELINES TO STUDY THE SUBJECT**

**Instructions to Students:**

1. Go through the 'Syllabus' in the Black Board section of the web-site(https://learn.upes.ac.in) in order to find out the Reading List.
2. Get your schedule and try to pace your studies as close to the timeline as possible.
3. Get your on-line lecture notes (Content, videos) at Lecture Notes section.  These are our lecture notes. Make sure you use them during this course.
4. Check your blackboard regularly
5. Go through study material
6. Check mails and announcements on blackboard
7. Keep updated with the posts, assignments and examinations which shall be conducted on the blackboard
8. Be regular, so that you do not suffer in any way
9. C**ell Phones and other Electronic Communication Devices:** Cell phones and other electronic communication devices (such as Blackberries/Laptops) are not permitted in classes during Tests or the Mid/Final Examination. Such devices MUST be turned off in the class room.
10. **E-Mail and online learning tool:** Each student in the class should have an e-mail id and a pass word to access the LMS system regularly. Regularly, important information – Date of conducting class tests, guest lectures, via online learning tool. The best way to arrange meetings with us or ask specific questions is by email and prior appointment. All the assignments preferably should be uploaded on online learning tool. Various research papers/reference material will be mailed/uploaded on online learning platform time to time.
11. **Attendance:** Students are required to attend at least 75% synchronous sessions. Students with less than said percentage shall NOT be allowed to appear in the end semester examination.

This much should be enough to get you organized and on your way to having a great semester! If you need us for anything, send your feedback through e-mail [abhijit.kumar@ddn.upes.ac.in](mailto:abhijit.kumar@ddn.upes.ac.in). Please use an appropriate subject line to indicate your message details.

There will no doubt be many more activities in the coming weeks. So, to keep up to date with all the latest developments, please keep visiting this website regularly.

**RELATED OUTCOMES**

1. **The expected outcomes of the Program are:**

|  |  |
| --- | --- |
| PO1 | **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| PO2 | **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| PO3 | **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. |
| PO4 | **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| PO5 | **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. |
| PO6 | **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. |
| PO7 | **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| PO8 | **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| PO9 | **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| PO10 | **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. |
| PO11 | **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| PO12 | **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

1. **The expected outcomes of the Specific Program are:**

The engineering graduates will be able to

|  |  |
| --- | --- |
| PSO1 | Perform system and application programming using computer system concepts, concepts of Data Structures, algorithm development, problem solving and optimizing techniques, |
| PSO2 | Apply software development and project management methodologies using concepts of front-end and back-end development and emerging technologies and platforms. |
| PSO 3 | Design solutions to challenging and ever growing real world data engineering problems and examine it to uncover hidden patterns, correlations, insights and make better data driven decisions. |

1. **The expected outcomes of the Course are:**

On completion of the course, the students would be able to

|  |  |
| --- | --- |
| CO 1 | Comprehend different phases of compiler. |
| CO 2 | Use concepts of regular grammar to build lexical analyzer. |
| CO 3 | Build parsers for a context free grammar. |
| CO 4 | Synthesize syntax directed translations rules. |
| CO 5 | Assess code and memory optimization techniques to improve the performance of a program. |

1. **Co-Relationship Matrix**

Indicate the relationships by1- Slight (low) 2- Moderate (Medium) 3-Substantial (high)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO 3** |
| **CO 1** | 2 | 2 | 3 |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO 2** | 2 | 2 | 3 |  | 1 |  |  |  |  |  |  |  | 2 |  |  |
| **CO 3** | 2 | 2 | 3 |  | 1 |  |  |  |  |  |  |  | 2 |  |  |
| **CO 4** | 2 | 2 | 3 |  | 1 |  |  |  |  |  |  |  | 2 |  |  |
| **CO 5** | 2 | 2 | 3 |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **Average** | 2 | 2 | 3 |  | 0.6 |  |  |  |  |  |  |  | 2 |  |  |

1. **Course outcomes assessment plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **components**  **Course Outcomes** | **Assignment** | **Test/Quiz** | **Mid Semester** | **End Semester** | **Any other** |
| **CO 1** | √ | √ | √ | √ |  |
| **CO 2** | √ | √ | √ | √ |  |
| **CO 3** | √ | √ | √ | √ |  |
| **CO 4** | √ | √ |  | √ |  |
| **CO5** | √ | √ |  | √ |  |

**BROAD PLAN OF COURSE COVERAGE**

**Course Activities:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Description** | **Planned** | | | **Remarks** |
| **From** | **To** | **No. of Sessions** |
| **1.** | INTRODUCTION |  |  | 8 |  |
| **2.** | BASIC PARSING TECHNIQUES |  |  | 14 |  |
| **3.** | SYNTAX-DIRECTED TRANSLATION |  |  | 8 |  |
| **4.** | SYMBOL TABLE |  |  | 7 |  |
| **5.** | INTRODUCTION TO CODE OPTIMIZATION |  |  | 8 |  |

Sessions: Total No. of Instructional periods available for the course

**SESSION PLAN**

**UNIT-I**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lecture No.** | **Topics to be Covered** | **Mode of Delivery** | **CO Mapped** |
| 1 | Introduction to the course and course-objectives | Synchronous | CO1 |
| 2 | Compiler Structure, Phases and Passes, Bootstrapping | Synchronous | CO1 |
| 3 | Compiler Structure, Phases and Passes, Bootstrapping (Contd..) | Synchronous | CO1 |
| 4 | Finite State Machines and Regular Expressions and their Applications to Lexical Analysis | Asynchronous | CO1 |
| 5 | Implementation of Lexical Analyzer, LEX | Synchronous | CO2 |
| 6 | Implementation of Lexical Analyzer, LEX (Contd...) | Synchronous | CO2 |
| 7 | Formal Grammars and their Applications to Syntax Analysis | Asynchronous | CO3 |
| 8 | The Syntactic Specification of Programming Languages | Synchronous | CO3 |

**8**

**UNIT-II**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lecture No.** | **Topics to be Covered** | **Mode of Delivery** | **CO Mapped** |
| 1 | Introduction to Parsing- Parse Tree, Derivation, & Ambiguity | Asynchronous | CO3 |
| 2 | Top Down Parsing | Synchronous | CO3 |
| 3 | Computation of FIRST & FOLLOW | Synchronous | CO3 |
| 4 | LL Parsers | Synchronous | CO3 |
| 5 | LL Parsers (Contd...) | Synchronous | CO3 |
| 6 | Shift Reduce Parsing, Handle, and Handle Pruning | Synchronous | CO3 |
| 7 | Operator Precedence Parsing | Synchronous | CO3 |
| 8 | LR Parsing | Synchronous | CO3 |
| 9 | LR Parsing (continued) | Synchronous | CO3 |
| 10 | Canonical LR | Synchronous | CO3 |
| 11 | LALR Parser | Asynchronous | CO3 |
| 12 | Introduction to YACC | Synchronous | CO3 |
| 13 | Application of YACC | Synchronous | CO3 |
| 14 | Class Test- 1 | Asynchronous | CO1, CO2, & CO3 |

**UNIT-III**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lecture No.** | **Topics to be Covered** | **Mode of Delivery** | **CO Mapped** |
| 1 | Syntax-Directed Translation Schemes and Implementation | Synchronous | CO4 |
| 2 | Syntax-Directed Translation Schemes and Implementation (continued...) | Synchronous | CO4 |
| 3 | Intermediate Code, Postfix Notation, Parse Trees & Syntax Trees | Synchronous | CO4 |
| 4 | Three Address Code, Quadruple & Triples | Synchronous | CO4 |
| 5 | Translation of Expressions | Synchronous | CO4 |
| 6 | Translation of Control Flow Statements | Synchronous | CO4 |
| 7 | More About Translation: Array References in Arithmetic Expressions, Procedure Calls | Asynchronous | CO4 |
| 8 | More About Translation: Array References in Arithmetic Expressions, Procedure Calls | Asynchronous | CO4 |

**UNIT-IV**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lecture No.** | **Topics to be Covered** | **Mode of Delivery** | **CO Mapped** |
| 1 | Data Structure for Symbol Table | Synchronous | CO4 |
| 2 | Implementation of Simple Stack Allocation Scheme | Asynchronous | CO4 |
| 3 | Storage Allocation in Block Structured Language | Asynchronous | CO4 |
| 4 | Error Handling in Compiler Design | Synchronous | CO4 |
| 5 | Lexical Phase Errors | Asynchronous | CO4 |
| 6 | Syntactic Phase Errors | Asynchronous | CO4 |
| 7 | Semantic Phase Errors | Asynchronous | CO4 |

**UNIT-V**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lecture No.** | **Topics to be Covered** | **Mode of Delivery** | **CO Mapped** |
| 1 | Introduction to Code Generation & Need of Code Optimization | Asynchronous | CO5 |
| 2 | DAG- Basics & Construction Examples | Synchronous | CO5 |
| 3 | BASIC Block and Flow Graphs | Synchronous | CO5 |
| 4 | Optimization of Basic Blocks | Synchronous | CO5 |
| 5 | Global Data Flow Analysis | Asynchronous | CO5 |
| 6 | Target Code Generations- Introduction and Examples | Synchronous | CO5 |
| 7 | Peephole Optimization | Synchronous | CO5 |
| 8 | Class Test - 2 | Asynchronous | CO3, CO4, & CO5 |