**Advanced Data Structures and Algorithm Analysis Lab**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | 23CS3351 | **Year** | II | **Semester** | II |
| **Course Category** | PC | **Branch** | CSE/IT | **Course Type** | Practical |
| **Credits** | 3 | **L-T-P** | 0-0-3 | **Prerequisites** | Data Structuresthrough C / Object Oriented Programming |
| **Continuous Internal** **Evaluation :** | 30 | **Semester End Evaluation:** | 70 | **Total Marks:** | 100 |

|  |  |
| --- | --- |
| **Course Outcomes** | **Blooms Level** |
| Upon successful completion of the course, the student will be able to: |
| **CO1** | Implement programs as an individual on different IDEs/ online platforms. | **L3** |
| **CO2** | Apply different algorithm design techniques for solving problems. | **L3** |
| **CO3** | Develop an effective report based on various programs implemented. | **L3** |
| **CO4** | Apply technical knowledge for a given problem and express with an effective oral communication. | **L3** |
| **CO5** | Analyze outputs using given constraints/test cases. | **L4** |

|  |
| --- |
| **Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:Substantial,2:Moderate,1:Slight)** |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO2** |  | 2 |  |  |  |  |  |  |  |  |  |  | 1 |  |
| **CO3** |  |  |  |  |  |  |  |  | 2 | 2 |  |  |  |  |
| **CO4** |  |  |  |  |  |  |  |  | 2 | 2 |  |  |  |  |
| **CO5** |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |
| **Avg.** | **3** | 2 |  |  |   |  |  |  | 2 | 2 |  |  | 1 |  |

|  |  |
| --- | --- |
| **CO 1** | **“**Implement programs as an individual on different IDEs/ online platforms**”** |
| **PO1** | **Engineering Knowledge** **Justification:** By implementing programs on various IDEs and online platforms, students apply fundamental engineering principles, computational thinking, and modern software tools to develop solutions, aligning with the Engineering Knowledge program outcome |
| **CO 2** | “Apply different algorithm design techniques for solving problems” |
| **PO2** | **Problem Analysis****Justification:** In the lab, students gain hands-on experience in identifying, designing, implementing, and analyzing problem-solving techniques, directly supporting the Problem Analysis. |
| **PSO1** | **Apply the Knowledge of Computing Skills in building the Software Systems that meet the requirements of Industry and Society.****Justification:** Students able to acquire a computing skills helps build industry-relevant and socially beneficial software, while design techniques provide structured methodologies to solve problems efficiently in the development process. |
| **CO3** | Develop an effective report based on various programs implemented. |
| **PO9 & PO10** | **Individual and Team Work, Communication Skills****Justification:** The students keep a day-to-day evaluation observation book and a lab record for each experiment they execute in the lab. The faculty may evaluate these documents, which are used to help students improve their written communication skills and teamwork. |
| **CO4** | Apply technical knowledge for a given problem and express with an effective oral communication. |
| **PO9 & PO10** | **Individual and Team Work, Communication Skills****Justification:** The faculty conducts a daily assessment of each student over lab experiments by asking relevant questions. This evaluation process assesses students' conceptual understanding, ability to apply theoretical knowledge in practical scenarios, and problem-solving skills. By asking targeted questions, the faculty encourages active participation, reinforces learning, and provides immediate feedback to improve students' understanding of the subject. |
| **CO5** | Analyze outputs using given constraints/test cases. |
| **PO2** | **Problem Analysis****Justification:** Comparing actual outputs with expected results to verify correctness. Utilizing debugging techniques to identify logical or implementation errors. Assessing software performance under various conditions, including boundary values and extreme cases. Ensuring adherence to performance constraints like time complexity and memory limits. Optimizing code for efficient handling of edge cases. |

Course Coordinator:

Module Coordinator: Signature of HOD