

| Analysis & Design of Algorithms Lab                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                        | Semester   | 4  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|----|
| Course Code                                                                                                                                                                                                                                                                                                                                                                                                                                        | BCSL404                                                                                                                                                                                                                                                                                                                                                                | CIE Marks  | 50 |
| Teaching Hours/Week (L:T:P: S)                                                                                                                                                                                                                                                                                                                                                                                                                     | 0:0:2:0                                                                                                                                                                                                                                                                                                                                                                | SEE Marks  | 50 |
| Credits                                                                                                                                                                                                                                                                                                                                                                                                                                            | 01                                                                                                                                                                                                                                                                                                                                                                     | Exam Hours | 2  |
| Examination type (SEE)                                                                                                                                                                                                                                                                                                                                                                                                                             | Practical                                                                                                                                                                                                                                                                                                                                                              |            |    |
| <b>Course objectives:</b> <ul style="list-style-type: none"><li>• To design and implement various algorithms in C/C++ programming using suitable development tools to address different computational challenges.</li><li>• To apply diverse design strategies for effective problem-solving.</li><li>• To Measure and compare the performance of different algorithms to determine their efficiency and suitability for specific tasks.</li></ul> |                                                                                                                                                                                                                                                                                                                                                                        |            |    |
| Sl.No                                                                                                                                                                                                                                                                                                                                                                                                                                              | Experiments                                                                                                                                                                                                                                                                                                                                                            |            |    |
| 1                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm.                                                                                                                                                                                                                                 |            |    |
| 2                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.                                                                                                                                                                                                                                    |            |    |
| 3                                                                                                                                                                                                                                                                                                                                                                                                                                                  | a. Design and implement C/C++ Program to solve All-Pairs Shortest Paths problem using Floyd's algorithm.<br>b. Design and implement C/C++ Program to find the transitive closure using Warshal's algorithm.                                                                                                                                                            |            |    |
| 4                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Design and implement C/C++ Program to find shortest paths from a given vertex in a weighted connected graph to other vertices using Dijkstra's algorithm.                                                                                                                                                                                                              |            |    |
| 5                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Design and implement C/C++ Program to obtain the Topological ordering of vertices in a given digraph.                                                                                                                                                                                                                                                                  |            |    |
| 6                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Design and implement C/C++ Program to solve 0/1 Knapsack problem using Dynamic Programming method.                                                                                                                                                                                                                                                                     |            |    |
| 7                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Design and implement C/C++ Program to solve discrete Knapsack and continuous Knapsack problems using greedy approximation method.                                                                                                                                                                                                                                      |            |    |
| 8                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Design and implement C/C++ Program to find a subset of a given set $S = \{s_1, s_2, \dots, s_n\}$ of $n$ positive integers whose sum is equal to a given positive integer $d$ .                                                                                                                                                                                        |            |    |
| 9                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Design and implement C/C++ Program to sort a given set of $n$ integer elements using Selection Sort method and compute its time complexity. Run the program for varied values of $n > 5000$ and record the time taken to sort. Plot a graph of the time taken versus $n$ . The elements can be read from a file or can be generated using the random number generator. |            |    |
| 10                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Design and implement C/C++ Program to sort a given set of $n$ integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of $n > 5000$ and record the time taken to sort. Plot a graph of the time taken versus $n$ . The elements can be read from a file or can be generated using the random number generator.     |            |    |
| 11                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Design and implement C/C++ Program to sort a given set of $n$ integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of $n > 5000$ , and record the time taken to sort. Plot a graph of the time taken versus $n$ . The elements can be read from a file or can be generated using the random number generator.   |            |    |
| 12                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Design and implement C/C++ Program for N Queen's problem using Backtracking.                                                                                                                                                                                                                                                                                           |            |    |

**Course outcomes (Course Skill Set):**

At the end of the course the student will be able to:

1. Develop programs to solve computational problems using suitable algorithm design strategy.
2. Compare algorithm design strategies by developing equivalent programs and observing running times for analysis (Empirical).
3. Make use of suitable integrated development tools to develop programs
4. Choose appropriate algorithm design techniques to develop solution to the computational and complex problems.
5. Demonstrate and present the development of program, its execution and running time(s) and record the results/inferences.

**Assessment Details (both CIE and SEE)**

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

**Continuous Internal Evaluation (CIE):**

CIE marks for the practical course are **50 Marks**.

The split-up of CIE marks for record/ journal and test are in the ratio **60:40**.

- Each experiment is to be evaluated for conduction with an observation sheet and record write-up. Rubrics for the evaluation of the journal/write-up for hardware/software experiments are designed by the faculty who is handling the laboratory session and are made known to students at the beginning of the practical session.
- Record should contain all the specified experiments in the syllabus and each experiment write-up will be evaluated for 10 marks.
- Total marks scored by the students are scaled down to **30 marks** (60% of maximum marks).
- Weightage to be given for neatness and submission of record/write-up on time.
- Department shall conduct a test of 100 marks after the completion of all the experiments listed in the syllabus.
- In a test, test write-up, conduction of experiment, acceptable result, and procedural knowledge will carry a weightage of 60% and the rest 40% for viva-voce.
- The suitable rubrics can be designed to evaluate each student's performance and learning ability.
- The marks scored shall be scaled down to **20 marks** (40% of the maximum marks).

The Sum of scaled-down marks scored in the report write-up/journal and marks of a test is the total CIE marks scored by the student.



- SEE shall be conducted jointly by the two examiners of the same institute, examiners are appointed by the Head of the Institute.
- The examination schedule and names of examiners are informed to the university before the conduction of the examination. These practical examinations are to be conducted between the schedule mentioned in the academic calendar of the University.
- All laboratory experiments are to be included for practical examination.
- (Rubrics) Breakup of marks and the instructions printed on the cover page of the answer script to be strictly adhered to by the examiners. **OR** based on the course requirement evaluation rubrics shall be decided jointly by examiners.
- Students can pick one question (experiment) from the questions lot prepared by the examiners jointly.
- Evaluation of test write-up/ conduction procedure and result/viva will be conducted jointly by examiners.

General rubrics suggested for SEE are mentioned here, writeup-20%, Conduction procedure and result in -60%, Viva-voce 20% of maximum marks. SEE for practical shall be evaluated for 100 marks and scored marks shall be scaled down to 50 marks (however, based on course type, rubrics shall be decided by the examiners)

Change of experiment is allowed only once and 15% of Marks allotted to the procedure part are to be made zero.

The minimum duration of SEE is 02 hours

#### **Suggested Learning Resources:**

- Virtual Labs (CSE): <http://cse01-iiith.vlabs.ac.in/>

#### **Semester End Evaluation (SEE):**

- SEE marks for the practical course are 50 Marks.

  
**HOD's Signature**  
**H.O.D.**  
**Dept. Of Information Science & Engineering**  
**Alva's Institute of Engg. & Technology**  
**Miiar, MOODBIDRI - 574 225**

