Name of the Course: Design and Analysis of Algorithms

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The Design and Analysis of Algorithms course is a fundamental exploration into the systematic study of algorithms, their design principles, and the analysis of their efficiency. It forms the backbone of computer science education, providing essential skills for solving complex computational problems. **Relevance:** In the ever-evolving landscape of computer science, the Design and Analysis of Algorithms course is highly relevant. It equips students with the intellectual tools necessary to address challenges in diverse areas, from software development to artificial intelligence. **Usefulness:** This course is instrumental in cultivating algorithmic thinking. Participants learn to devise efficient algorithms, analyze their correctness, and evaluate their performance, essential skills for creating optimized solutions in various computing applications. **Application:** The knowledge gained from this course finds application in a myriad of scenarios, from developing efficient search and sorting algorithms to optimizing resource utilization in network design and artificial intelligence. **Interest:** The course often captivates students due to its intellectual challenges and problem-solving nature. Participants engage in dissecting complex problems, devising algorithmic solutions, and analyzing their efficiency, fostering a deep appreciation for algorithmic thinking. **Connection with Other Courses:** The Design and Analysis of Algorithms course establishes vital connections with other computer science disciplines. It forms the basis for advanced courses in data structures, algorithmic complexity, and computational theory, providing a holistic understanding of computation.

		Demand in the Industry: Professionals well-versed in algorithm design and analysis are in high demand. Industries ranging from technology and finance to healthcare actively seek individuals who can develop efficient algorithms to solve complex problems and enhance system performance. **Job Prospects:** Graduates from a Design and Analysis of Algorithms course find themselves well-positioned for various roles, including software engineer, algorithm developer, data scientist, and research scientist. These professionals are valued for their ability to devise elegant and efficient solutions to computational challenges.
2	**Vertical:**	Major
3	**Type:**	Theory
4	**Credits:**	2 credits (1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** **CO 1.** To make students understand the basic principles of algorithm design **CO 2.** To give idea to students about the theoretical background of the basic data structures **CO 3.** To familiarize the students with fundamental problem-solving strategies like searching, sorting, selection, and recursion and help them to evaluate efficiencies of various algorithms. **CO 4.** To teach students the important algorithm design paradigms and how they can be used to solve various real world problems	
8	**Course Outcomes (OC):** **OC 1.** Students should be able to understand and evaluate efficiency of the programs that they write based on performance of the algorithms used. **OC 2.** Students should be able to appreciate the use of various data structures as per need **OC 3.** To select, decide and apply appropriate design principle by understanding the requirements of any real life problems.	
9	**Modules:-** **Module 1 (15 hours):**	
	Introduction to algorithms - What is algorithm, analysis of algorithm, Types of complexity, Running time analysis, How to Compare Algorithms, Rate of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega-Ω Notation, Theta-Θ Notation, Asymptotic Analysis, Performance characteristics of algorithms,	

	Estimating running time / number of steps of executions on paper, Idea of Computability **Introduction to Data Structures -** What is data structure, types, Introduction to Array(1-d & 2-d), Stack and List data structures, operations on these data structures, advantages disadvantages and applications of these data structures like solving linear equations, Polynomial Representation, Infix-to-Postfix conversion. **Recursion -** What is recursion, Recursion vs Iteration, recursion applications like Factorial of a number, Fibonacci series & their comparative analysis with respect to iterative version, Tower of Hanoi problem. **Basic Sorting Techniques** - Bubble, Selection and Insertion Sort & their comparative analysis
	Module 2 (15 hours):
	Searching Techniques - Linear Search and its types, Binary Search and their comparative analysis, Selection Techniques - Selection by Sorting, Partition-based Selection Algorithm, Finding the Kth Smallest Elements in Sorted Order & their comparative analysis, String Algorithms - Pattern matching in strings, Brute Force Method & their comparative analysis **Algorithm Design Techniques** - Introduction to various types of classifications/design criteria and design techniques, Greedy Technique - Concept, Advantages & Disadvantages, Applications, Implementation using problems like - file merging problem. Divide-n-Conquer - Concept, Advantages & Disadvantages, Applications, Implementation using problems like - merge sort, Strassen's Matrix Multiplication **Dynamic Programming -** Concept, Advantages & Disadvantages, Applications, Implementation using problems like - Fibonacci series, Factorial of a number, Longest Common subsequence **Backtracking Programming -** Concept, Advantages & Disadvantages, Applications, Implementation using problems like N-Queen Problem
10	**Text Books** 1. Data Structure and Algorithm Using Python, Rance D. Necaise, Wiley India Edition, 2016. 2. Data Structures and Algorithms Made Easy, Narasimha Karumanchi, CareerMonk Publications, 2016. 3. Introduction to Algorithms, Thomas H. Cormen, 3rd Edition, PHI.
11	**Reference Books** 1. Introduction to the Design and Analysis of Algorithms, Anany Levitin, Pearson, 3rd Edition, 2011. 2. Design and Analysis of Algorithms, S. Sridhar, Oxford University Press, 2014.

<table>
<tr><td><b>12</b></td><td><b>Internal Continuous Assessment: 40%</b></td><td><b>Semester End Examination: 60%</b></td></tr>
<tr><td><b>13</b></td><td><b>Continuous Evaluation through:</b>
Class Test on Module 1: 10 marks
Class Test on Module 2: 10 marks
<b>Average of 2 Class Tests: 10 marks</b>
Assignment on Module 1: 5 marks
Assignment on Module 2: 5 marks
<b>Total of 2 Assignments: 10 marks</b>
<b>Total: 20 marks</b></td><td><b>Evaluation through:</b>
A <b>Semester End Theory Examination</b> of <b>1 hour duration</b> for <b>30 marks</b> as per the paper pattern given below.
<b>Total: 30 marks</b></td></tr>
<tr><td><b>14</b></td><td colspan="2"><b>Format of Question Paper:</b>
<b>Total Marks: 30</b> <b>Duration: 1 Hour</b>
<table>
<tr><td><b>Question</b></td><td><b>Based On</b></td><td><b>Options</b></td><td>Marks</td></tr>
<tr><td><b>Q. 1</b></td><td>Module 1</td><td><i>Any 2 out of 4</i></td><td>10</td></tr>
<tr><td><b>Q. 2</b></td><td>Module 2</td><td><i>Any 2 out of 4</i></td><td>10</td></tr>
<tr><td><b>Q. 3</b></td><td>Module 1 &amp; 2</td><td><i>Any 2 out of 4</i></td><td>10</td></tr>
</table>
</td></tr>
</table>