Name of the Course: Computer Science Practical 2

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The Computer Science Practical Course covering Design and Analysis of Algorithms and Object-Oriented Programming (OOP) using C++ is a comprehensive exploration into fundamental computer science concepts and practical programming skills. It integrates the study of algorithmic design with hands-on application using the C++ programming language. **Relevance:** In the dynamic field of computer science, the integration of algorithmic design and object-oriented programming is highly relevant. This course equips students with essential skills to solve complex problems, design efficient algorithms, and implement practical solutions using the OOP paradigm in C++. **Usefulness:** The course is invaluable for developing a strong foundation in algorithmic thinking and software design. Students learn to analyze algorithm efficiency, apply OOP principles for code modularity, and create robust software solutions, enhancing their overall programming proficiency. **Application:** The concepts acquired in this practical course find direct application in real-world scenarios. Students engage in hands-on projects where they design and implement algorithms, analyze their performance, and develop software applications using object-oriented principles in C++. **Interest:** The practical nature of the course often captivates students. Through project-based learning, participants apply algorithmic strategies, design class hierarchies, and implement solutions in C++, fostering a deep interest in problem-solving and software development. **Connection with Other Courses:** This practical course establishes a strong connection with other computer science courses. It lays the groundwork for advanced studies in algorithmic complexity, data structures, software engineering, and advanced topics in object-oriented programming, providing a well-rounded education.

		Demand in the Industry: Professionals with proficiency in algorithmic design and object-oriented programming in C++ are in high demand. Industries spanning software development, technology, and finance actively seek individuals who can apply these skills to create efficient and scalable software solutions. **Job Prospects:** Graduates from this practical course have diverse job prospects. Roles may include software engineer, algorithm developer, systems analyst, or application developer. These professionals are valued for their ability to contribute to algorithmically optimized, modular, and maintainable software.
2	**Vertical:**	Major
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** **CO 1.** Analyze and implement algorithms for common computational problems. **CO 2.** Implement algorithms using divide and conquer strategies. **CO 3.** Apply dynamic programming techniques to solve optimization problems. **CO 4.** Implement and analyze algorithms based on greedy strategies. **CO 5.** Comprehend the principles of object-oriented programming. **CO 6.** Design and implement classes and objects in C++. **CO 7.** Implement single, multiple, and hierarchical inheritance. **CO 8.** Implement operator overloading for user-defined types. **CO 9.** Understand the impact of access specifiers on class members.	
8	**Course Outcomes (OC):** **OC 1.** Design and implement algorithms for various problem domains. **OC 2.** Evaluate and compare the time and space complexities of algorithms. **OC 3.** Apply divide and conquer strategies to solve computational problems. **OC 4.** Utilize dynamic programming techniques for optimization problems. **OC 5.** Implement and analyze algorithms based on greedy strategies. **OC 6.** Design and implement classes and objects in C++. **OC 7.** Apply inheritance and polymorphism concepts in program development. **OC 8.** Implement operator overloading for enhanced class functionality. **OC 9.** Utilize advanced features like friend functions, inline functions, and this pointer. **OC 10.** Understand the impact of scope specifiers on class members.	

9	**Modules:-** **Module 1 (30 hours):**
	Design & Analysis of Algorithms – Practical **Array Operations:** Implement programs for 1-d arrays, Implement programs for 2-d arrays. **List-Based Stack Operations:** Create a list-based stack and perform stack operations. **Linear and Binary Search:** Implement linear and binary search algorithms on a list. **Sorting Algorithms:** Implement sorting algorithms (e.g., bubble, selection, insertion). **Nth Max/Min Element:** Implement algorithms to find Nth Max/Min element in a list. **String Pattern Matching:** Implement algorithms to find a pattern in a given string. **Recursion:** Implement recursive algorithms (e.g., factorial, Fibonacci, Tower of Hanoi). **Greedy Algorithm:** Solve problems like file merging and coin change using the Greedy Algorithm. **Divide and Conquer:** Implement algorithms like merge sort and Strassen's Matrix Multiplication. **Dynamic Programming:** Implement algorithms for Fibonacci series and Longest Common Subsequence using dynamic programming.
	Module 2 (30 hours):
	OOPs using C++ – Practical **Introduction to Classes:** Create a simple class with data members and member functions. Demonstrate the use of class instances to access data and invoke member functions. **Branching and Looping with Classes:** Implement programs utilizing branching and looping statements within class methods. **Arrays and Classes:**

	Develop a program that employs one and two-dimensional arrays within a class. Illustrate how classes can handle array-based data structures. **Scope Resolution Operator:** Use the scope resolution operator to declare variables at different scope levels. Display and compare the values of variables with different scopes. **Constructors and Destructors:** Implement programs showcasing various types of constructors and destructors. Explore default, parameterized, copy constructors, and destructor functionalities. **Access Specifiers:** Demonstrate the use of public, protected, and private scope specifiers within a class. Understand the impact of different access specifiers on class members. **Inheritance:** Implement classes to demonstrate single and multilevel inheritance scenarios. Showcase how derived classes inherit properties from the base class. Develop programs illustrating multiple and hierarchical inheritance. Create programs that demonstrate the interaction between inheritance and derived class constructors. Understand the order of constructor invocation in the inheritance hierarchy. **Advanced Concepts:** Implement programs showcasing friend functions, inline functions, and the use of the this pointer within classes. **Function Overloading and Overriding:** Develop programs to demonstrate function overloading and overriding within classes. **Pointers and File Handling:** Explore the use of pointers within classes, emphasizing dynamic memory allocation. Develop programs for both text and binary file handling within a class context.
10	**Text Books** 1. Data Structure and Algorithm Using Python, Rance D. Necaise, Wiley India Edition, 2016. 2. Object Oriented Programming with C++, Balagurusamy E., 8th Edition, McGraw Hill Education India.

<table>
<tr><td>11</td><td colspan="2">Reference Books
1. Data Structures and Algorithms Made Easy, Narasimha Karumanchi, CareerMonk Publications, 2016.
2. Let Us C++ by KanetkarYashwant, Publisher: BPB Publications, 2020</td></tr>
<tr><td>12</td><td>Internal Continuous Assessment: 40%</td><td>Semester End Examination: 60%</td></tr>
<tr><td>13</td><td>The internal evaluation will be determined by the completion of practical tasks and the submission of corresponding write-ups for each session. Each practical exercise holds a maximum value of 5 marks. The total evaluation, out of 100 marks, should be scaled down to a final score of 20 marks.

Total: 20 marks</td><td>A Semester End Practical Examination of 2 hours duration for 30 marks as per the paper pattern given below.

Certified Journal is compulsory for appearing at the time of Practical Exam

Total: 30 Marks</td></tr>
<tr><td>14</td><td colspan="2">Format of Question Paper:

Total Marks: 30 Duration: 2 Hours
<table>
<tr><th>Question</th><th>Practical Question Based On</th><th>Marks</th></tr>
<tr><td>Q. 1</td><td>Module 1</td><td>12</td></tr>
<tr><td>Q. 2</td><td>Module 2</td><td>12</td></tr>
<tr><td>Q. 3</td><td>Viva</td><td>06</td></tr>
</table></td></tr>
</table>