|  |  |  |  |
| --- | --- | --- | --- |
| **Practical-III** | | **BCAP 184**  **C++ LAB** | **48 Hrs** |
| **Practical/Week: 4 Hrs**  **Credits: 2** | | **Exercises on C++ Programming** | **I.A.: 20**  **Exam: 80** |
| **PART A** | | | |
| **1** | Write a program with a class POLYMORPH to calculate the volume of sphere, cylinder and rectangular prism using function overloading concept. | | |
| **2** | Write a program to perform the following operations on two complex numbers:   1. Addition using a member function 2. Subtraction using a friend function | | |
| **3** | Write a program to compute the total marks and declare the results using an array of objects. Assume that the class contains the data members - roll no, name, marks in 3 subjects. Result is calculated as follows.  If student gets <35 in any of the subjects, Fail. Otherwise various results are calculated on the basis of average as   1. >=70 Distinction 2. >=60 and <70 First Class 3. >=50 and <60 Second Class else Pass Class.   Use member functions to accept the data, compute, and display the result in tabular form. | | |
| **4** | Write a program to create a class DISTANCE with the data members feet and inches. Use a constructor to read the data and a member function Sum ( ) to add two distances by using objects as function arguments and show the result. (Input and output of inches should be less than 12.) | | |
| **PART B** | | | |
| **1** | Using constructors and proper methods, design a class graphics which stores shapes, area, back colour and fore colours. Use this class in the main program to input any ‘N’ shapes and perform the following operations and print the list in a neat format.   1. Sort according to area 2. Search for a specified shape. | | |
| **2** | Create a class ‘Bank’ which includes data members – Acno, Name, Balance and a parameterized constructor to initialize the data members and other methods like deposit, withdrawal, and display the details of the customer.  (Hint: Check for minimum balance of Rs. 500/- while opening the account and during the withdrawal. Also, amount should be positive integer. Otherwise show appropriate message.) | | |
| **3** | Write a program to accept two strings and using operator overloading perform the following.   1. Concatenation of two strings. 2. Comparison of two strings alphabetically.   (Note : For concatenation (+), for comparison (==, > or <) | | |
| **4** | Create a class ‘Time’ which includes the data members – hours, minutes and seconds. Write a menu driven program with the following methods to   1. accept time 2. display time 3. increment time by one second by overloading unary operator ++ 4. decrement time by one second by overloading unary operator - -   (Hint: Validate minutes and seconds to be in the range of 0-59 in input and output). | | |

|  |  |
| --- | --- |
| **PART C** | |
| **1** | Using single inheritance, create a class ELECTRICITY that includes Tariff code, Zone, Meter number. Tariff code can be LT1, LT2 or LT3. Zone is either RURAL or URBAN.  Create another class CUSTOMER that inherits ELECTICITY, and includes AccountID, CustName, Address, Previous reading and Present reading. Validate for Present reading >= Previous reading. A fixed amount of Rs. 200/- to be paid by all the customers. Prepare an electricity bill with all the details in a neat format using the following data: |
| **2** | Using hierarchical inheritance, create a base class ‘ITEM’ with data members item number, title and price. Derive the following items for base class ITEM.   1. ‘Book’ with author name, publication and pages as the data members. 2. ‘CD’ with data members - category, time of play and speed.   Issue desired number of items and print the list of books and CD’s separately. Also print the number of books, CDs and total number of items purchased. |
| **3** | Using multiple inheritance, write a program to create a class ‘Personnel Information’ which includes name, address and gender as the data members. Another class for ‘Physical Information’ with data members height, weight, blood group. Derive a class called ‘Salary’ which inherits from the above two classes, with employee number, department and salary. Find increment in salary for an employee as follows.  For Male: In department S or P - 10% For Female: In department S or P - 11%.  (Hint: S for Sales and P for Purchase; for any other department, no increment.) |
| **4** | Create a class Employee containing name and EmpNo. Create two more classes **Manager** with data members department name and number of employees under that department, and **Scientist** with data members year and number of publications. Using the concept of **containership**, read all the information of a Scientist and Manager and display the information in a neat format. |

Scheme of Examination

|  |  |  |  |
| --- | --- | --- | --- |
| Tariff code | UNITS | RURAL | URBAN |
| LT1 | 0 to 40 | 6.80 | 6.80 |
|  | Above 40 | 7.00 | 7.00 |
| LT2 | 0 to 30 | 3.40 | 3.55 |
| 31 to 100 | 4.65 | 4.95 |
| Above 100 | 6.20 | 6.70 |
| LT3 | 0 to 50 | 7.25 | 7.75 |
| 51 & above | 8.55 | 8.95 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Details** | | | **Marks** | **Total** |
| 1 | PART A | i. | Problem solving and coding | 8 | **18** |
| ii. | Compiling the code and debugging | 6 |
| iii | Execution and testing | 4 |
| 2 | PART B | i. | Problem solving and coding | 10 | **22** |
| ii. | Compiling the code and debugging | 7 |
| iii | Execution and testing | 5 |
| 3 | PART C | i. | Problem solving and coding | 11 | **25** |
| ii. | Compiling the code and debugging | 8 |
| iii. | Execution and testing | 6 |
| **4** | **Class Records** | | | | **10** |
| **5** | **Viva – Voce** | | | | **5** |
| **Total Marks** | | | | | **80** |