Department of Computer Science & Engineering

Semester III

OOPs with C++(PCS-307) TermWork

INSTRUCTIONS TO BE FOLLOWED

1.Use Bond paper sheets (one side ruled and other blank) for writing the experiments.

2. A4 sheets should be used for the C++ Program and Program output. Both the program

and Program output should be on different pages and code should be in properly indented.

Font size should be 12pt TIMES NEW ROMAN.

3. For output screenshot is preferred (but change the background to white instead of black)

as it covers the IDE also, and arrange it in such a way so that it is visible to evaluator.

4. The problem statements should be in same order as it is given and must be written in

the same order in the INDEX also.

5. In the Program page only write your name and roll no in comments. Don’t write question

in A4 pages, it must be written in Bond sheets with Problem statement as heading

6. There should be no footer on the A4 page adjust your work accordingly

In Bond sheets you must write

1.Problem statement

2.Theory (about the programs, what the syntax you are using, methods, attributes, draw

diagrams where required using pencil on the left size of sheet)

3.Observation (what you observe from the logic of the program, where it is used in real

life applications)

**PROBLEM STATEMENTS**

|  |  |  |
| --- | --- | --- |
|  | An electricity board charges the following rates to domestic users to discourage large consumption of energy.  For the first 100 units: - 60 P per unit  For the next 200 units: -80 P per unit  Beyond 300 units: -90 P per unit  All users are charged a minimum of Rs 50 if the total amount is more than Rs 300 then an additional surcharge of 15% is added.  Implement a C++ program to read the names of users and number of units consumed and display the charges with names |  |
|  | Printing an array into Zigzag fashion. Suppose you were given an array of integers, and you are told to sort the integers in a zigzag pattern. In general, in a zigzag pattern, the first integer is less than the second integer, which is greater than the third integer, which is less than the fourth integer, and so on. Hence, the converted array should be in the form of e1 < e2 > e3 < e4 > e5 < e6. **Test cases:** Input 1:  7  4 3 7 8 6 2 1  Output 1:  3 7 4 8 2 6 1  Input 2:  4  1 4 3 2  Output 2:  1 4 2 3 |  |
|  | Implement a C++ program to find the non-repeating characters in string.  Typical Input: graphic era university  Typical Output: c g h n p s t u v y |  |
|  | Define a class Hotel in C++ with the following specifications **Private members** • Rno Data member to store room number • Name Data member to store customer name • Tariff Data member to store per day charges • NOD Data member to store number of days of stay • CALC() Function to calculate and return amount as NOD\*Tariff ,and if the value of days\* Tariff >10000, then total amount is 1.05\* days\*Tariff. **Public members** • Checkin() Function to enter the content Rno, Name, Tariff and NOD • Checkout() Function to display Rno, Name, Tariff, NOD and Amount (amount to be displayed by calling function) CALC() |  |
|  | Write a java program to create a class named 'Bank ' with the following data members:  Name of depositor  Address of depositor  Account Number  Balance in account Class 'Bank' has a method for each of the following:  1 - Generate a unique account number for each depositor For first depositor, account number will be 1001, for second depositor it will be 1002 and so on 2 - Display information and balance of depositor 3 - Deposit more amount in balance of any depositor 4 - Withdraw some amount from balance deposited 5 - Change address of depositor After creating the class, do the following operations 1 - Enter the information (name, address, account number, balance) of the depositors. Number of depositors is to be entered by user. 2 - Print the information of any depositor. 3 - Add some amount to the account of any depositor and then display final information of that depositor 4 - Remove some amount from the account of any depositor and then display final information of that depositor 5 - Change the address of any depositor and then display the final information of that depositor 6 - Randomly repeat these processes for some other bank accounts. |  |
|  | Write a C++ program to create a class called StaticDemo and overload funcDemo() static member function.  **static void funcDemo(string,int)**🡺 truncate a string to a certain number of words.  Example  "The quick brown fox jumps over the lazy dog" , 4  Output :  The quick brown fox  **static void funcDemo(string)**🡺 count and print number of palindrome in a passing string.  **static void funcDemo(int,string)**🡺 chop a string into chunks of a given length.  Example:  2, "INFORMATION"  Output :  IN FO RM AT IO N  Example:  3, "INFORMATION"  Output :  INF ORM ATI ON  In main function invoke functions and perform the above task. |  |
|  | Imagine a tollbooth with a class called TollBooth. The two data items are of type unsigned int and double to hold the total number of cars and total amount of money collected. A constructor initializes both of these data members to 0. A member function called payingCar( )increments the car total and adds 0.5 to the cash total. Another function called nonPayCar( ) increments the car total but adds nothing to the cash total. Finally a member function called display( )shows the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car and another to count a non paying car. Pushing the ESC key should cause the program to print out the total number of cars and total cash and then exit. |  |
|  | Create a class called Time that has separate int member data for hours, minutes and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. A member function should display it in 11:59:59 format. A member function named add() should add two objects of type time passed as arguments. A main ( ) program should create two initialized values together, leaving the result in the third time variable. Finally, it should display the value of this third variable. |  |
|  | Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest() to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12.This interest should be added tosavingsBalance. Provide a static method modifyInterestRate() that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs2000.00 and Rs3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month’s interest and print the new balances for both savers |  |
|  | Define a cla Define a class named **UserOne** with following description:   |  |  | | --- | --- | | **Data Members** | **Description** | | Name | To store person name | | FatherName | To store Father Name | | MotherName | To store Mother Name | | gender | To store gender | | **Member Functions** | **Description** | | InputInfo() | Input Name, Father name and Mother name and gender | | **Friend Function** | **Description** | | Userchecker(UserOne,UserTwo) | Compare data |   Define another class named **UserTwo** with following description:   |  |  | | --- | --- | | **Data Members** | **Description** | | Name | To store person name | | FatherName | To store Father Name | | MotherName | To store Mother Name | | gender | To store gender | | **Member Functions** | **Description** | | InputInfo() | Input Name, Father name and Mother name and gender | | **Friend Function** | **Description** | | Userchecker(UserOne,UserTwo) | Compare data |   Write a C++ program to compare Father Name, Mother Name of both classes using friend function. If Father name and Mother Name of both classes are equal then print the message “Belongs to Same Family” and display message “We are Brothers” or “We are Sisters” or “We are brother and sister”, otherwise print the message “Belongs to different Family”. |  |
|  | Create a class Complex having two int type variable named real & img denoting real and imaginary part respectively of a complex number. Overload +, - , = = operator to add, to subtract and to compare two complex numbers being denoted by the two complex type objects |  |
|  | Using the concept of operator overloading. Implement a program to overload the following:  a. Unary –  b. Unary ++ preincrement, postincrement  c. Unary -- predecrement, postdecrement |  |
|  | Using the concept of operator overloading. Implement a program to overload the following:  With the help of friend function  a. Unary –  b. Unary ++ preincrement, postincrement  c. Unary -- predecrement, postdecrement |  |
|  | Create a Base class that consists of private, protected and public data members and member functions. Try using different access modifiers for inheriting Base class to the Derived class and create a table that summarizes the above three modes (when derived in public, protected and private modes) and shows the access specifier of the members of base class in the Derived class. |  |
|  | Create a class called Student that contains the data members like age, name, enroll\_no, marks. Create another class called Faculty that contains data members like facultyName, facultyCode, salary,deptt, age, experience, gender. Create the function display() in both the classes to display the respective information. The derived Class Person demonstrates multiple inheritance. The program should be able to call both the base classes and displays their information. Remove the ambiguity (When we have exactly same variables or same methods in both the base classes, which one will becalled?) by proper mechanism |  |
|  | Implement a real case scenario by a proper C++ code to provide the solution to Diamond Problem in C++ |  |
|  | Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from base shape. Add to the base class, a member function get\_data() to initialize base class data members and another member function display\_area() to compute and display the area of figures. Make display\_area() as a virtual function and redefine this function in the derived class to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area. Remember the two values given as input will be treated as lengths of  two sides in the case of rectangles and as base and height in the case of triangle and used as follows:  Area of rectangle = x \* y  Area of triangle = ½ \*x\*y |  |
| 18 | Create a base class called CAL\_AREA(Abstract). Use this class to store float type values that could be used to compute the volume of figures. Derive two specific classes called cone, hemisphere and cylinder from the base CAL\_AREA. Add to the base class, a member function getdata ( ) to initialize base class data members and another member function display volume( ) to compute and display the volume of figures. Make display volume ( ) as a pure virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a cone, cylinder and hemisphere interactively  and display the volumes. Remember values given as input will be and used as follows:  Volume of cone = (1/3)πr2h  Volume of hemisphere = (2/3)πr3  Volume of cylinder = πr2h |  |
| 19 | Write a C++ file handling program to count and display the number of palindrome present in a text file "myfile.txt".  Example: If the file "myfile.txt" contains the following lines,  My name is aba dad  Hello aba dad  How are you  aba is my friend  Output will be => 5 |  |
| 20. | Create a class MyCalculator which consists of a single method power(int, int).  This method takes two integers, n and p, as parameters and finds n p .  If either n or p is negative, then the method must throw an exception which says "n and p should be non-negative".  Input Format  Each line of the input contains two integers, n and p .  Output Format  Each line of the output contains the result ,if neither of n and p is negative.  Otherwise the output contains "n and p should be non-negative".  Sample Input  3 5  2 4  0 0  -1 -2  -1 3  Sample Output  243  16  C++Exception: n and p should not be zero.  C++Exception: n or p should not be negative.  C++Exception: n or p should not be negative.  Explanation  In the first two cases, both n and p are positive. So, the power function returns the answer correctly.  In the third case, both n and p are zero. So, the exception, "n and p should not be zero.” is printed.  In the last two cases, at least one out of n and p is negative. So, the exception, "n or p should not be negative.” is printed for these two cases. |  |
| 21. | Construct a C++ program to demonstrate different methods of List, Vector and Map in STL (Standard Template Library) |  |
|  |  |  |