PSG COLLEGE OF TECHNOLOGY, COIMBATORE.

Department of Applied Mathematics and Computational Sciences

M.Sc. Cyber Security (2nd Semester)

20XC27 – Object Computing Lab

Problem Sheet – I (Classes & Objects)

---

ws1_p1. Write a C++ program that prompts the user for the radius of a circle then calls an inline function `funCircleArea()` to calculate the area of that circle.

ws1_p2. Write a complete C++ program with the three alternate functions as specified below, of which each function simply triples the variable '`count`' defined in main. The functions are: (a) `funTripleByValue()` that passes a copy of '`count`' by value, triples the variable and returns the new value and (b) `funTripleByAddress()` that passes the address of the variable '`count`' and returns the new value, and (c) `funTripleByReference()` that passes count by reference via a reference parameter and triples the original value of count. For all the function calls, the tripled values should be printed in `main()` function.

ws1_p3. Write a C++ program read the units for square, rectangle, triangle and circle and implement overloaded functions to calculate the area of each.

ws1_p4. Create a '`Date`' class, with day, month and year as data members, and the following function members. (Hint – Date validation is not required)

a. `getDate ()` – to read the value of a date object
b. `setDate ()` – to assign a date value
c. `printDate ()` – to display the date object value

ws1_p5. Create a class called `Account` that a bank might use to represent customer's bank accounts. `Account` class should include data members to store the account number, customer name, and balance. Your class should provide a member function to initialize the data member. The class should also provide three other member functions.

a. `funCredit()` should add an amount to the current balance.
b. `funDebit()` to withdraw money from the Account and should ensure that the debit amount does not exceed the Account's balance.
c. `funGetBalance()` should return the current balance.

ws1_p6. Create a class called **Rational** for performing arithmetic operation with fractions. Write a program to test your class. Use integer variable to represent the private data of the class- the numerator and the denominator. Provide a member function that enables an object of this class to be initialized the data members and should store the fraction in reduced form. For example, the fraction 2/4 would be stored in the object as 1 in the numerator and 2 in the denominator. Also provide public member functions that perform each of the following tasks:

- Adding two Rational numbers. The result should be stored in reduced form.
- Subtracting two Rational numbers. The result should be stored in reduced form.
- Multiplying two Rational numbers. The result should be stored in reduced form.
- Dividing two Rational numbers. The result should be stored in reduced form.
- Printing Rational numbers in the form a/b.
- Printing Rational numbers in floating-point format.

ws1_p7. Given that an **Employee** class contains following members:

Data members: Employee_Number, Employee_Name, Basic Pay, DA, IT, Net Salary

Member Functions: to read the data, to calculate Net Salary and to print data members.

Write a C++ program to read the data of employee and compute Net Salary of the employee (DA= 52% of Basic Pay and Income Tax (IT)=30% of the Basic Pay, Net Salary = BP+DA-IT).

ws1_p8. Define a **Student** class with RollNo, Name and Marks in 3 subjects. Declare an array of 10 **Student** objects. Using appropriate member functions to read the values, and find the average of best two marks for each student. Print the RollNo, Name and average marks of all the students.

ws1_p9. Write a C++ program to create a class called **Octal** which has the characteristics of an octal number. Implement the following operations by writing an appropriate member functions.

OCTAL h=k;

int y=h+k; int y=h-K; int y = h*k; int y = h/k;

where h is an octal number and k is an integer. And display the results in octal representation.

ws1_p10. Create a class called **Time** that has integer data elements for hours, minutes, and seconds, also write the following member functions. A member function should initialize these data elements to specified value, if given and otherwise to 0. A member function should display it, in 11:50:45 format. Define other two member functions to add and subtract two time object.

[Hint – The problem number should be your file name, eg. ws1_p1.cpp]