School of Computing

End Semester Examination – July 2021

Course Code: CSE201 – (Semi-Lab)
Course Name: Object Oriented Programming in C++

Duration: 120 minutes
Max Marks: 50
Number of Questions: 02

Instructions

- You will be given two questions with 25 marks each
- Execute the programs and prepare a single PDF file containing both the programs and its output
    - Create a word file.
    - Type your Name and register number.
    - Copy the programs and paste it in the word file (Screenshots of the code will not be considered for evaluation)
    - Take a snapshot of the output & paste it after the corresponding program.
    - Convert the word file to pdf file, then upload
- Name your file as follows: 12010001_Lab.pdf (Give your reg.no)
- Submissions with multiple files will not be evaluated.
- Exam duration is between 9:30 am to 11:30 am
- Upload time: 11:30 am to 11:45 am

In the heyday of the British empire, Great Britain used a monetary system based on pounds, shillings, and pence. There were 20 shillings to a pound, and 12 pence to a shilling. The notation for this old system used the pound sign, £, and two decimal points, so that, for example, £5.2.8 meant 5 pounds, 2 shillings, and 8 pence. (Pence is the plural of penny.) The new monetary system, introduced in the 1950s, consists of only pounds and pence, with 100 pence to a pound (like U.S. dollars and cents). We'll call this new system decimal pounds. Thus £5.2.8 in the old notation is £5.13 in decimal pounds (actually £5.1333333).

Create a class **OldPound** with three attributes for pounds, shillings and pence.

Add operator overloading function for converting OldPound object into float type which is the decimal pound equivalent.

An example of the user's interaction with the program would be
Enter pounds: 7
Enter shillings: 17
Enter pence: 9
Decimal pounds = 7.89

Hint: Conversion logic has to be derived from the numerical example given in the question.

Develop a class called **GeometricShape**. Use this class to store double type values that would be used to compute the area of geometric shapes.

Derive three specific classes called **Rhombus**, **Rectangle**, and **Trapezoid** from GeometricShape.

Add base class member functions called getdata() to initialize base class data members and displayarea() to compute and display the area of shapes. Make displayarea() as a pure virtual function and redefine this function in the derived classes to suit their requirements.

Write a main() to illustrate virtual functions.

(Hint: Rhombus: Area= (x ∗ y)/2, Rectangle: Area= x ∗ y, Trapezoid: Area= $\frac{x+y}{2}$ ∗ h).