CSE 108

Offline Assignment 1

Consider the following code template and add necessary code as instructed. You are not allowed to change any code that has been already added in the template.

Task 1: Your goal is to produce the same output given at the end of the template.

Task 2: Add another class called “Line” that has two Point objects p1 and p2 as private members, a constructor in the form Line(int a1, int b1, int a2, int b2) and a public member function to print the length of the line constructed using p1 and p2. The constructor initializes the coordinates of p1 with a1 and b1, and p2 with a2 and b2, respectively. Create an object of “Line” from main function and print its length.

```
#include <iostream>
using namespace std;

class Point
{
      // x and y coordinate
            int x, y;
    public:
      //Add necessary constructor(s) to initialize x and y
      //Add your set and get functions for x and y

            void print()
            {
                  cout << "Coordinate: "<< x <<", " <<y <<endl;
            }
};

class Circle
{
    Point p;
    int radius;
    public:
        // Add necessary constructor(s) to initialize p and radius
        // Overload “upload” functions

        void print()
        {
            cout << "Center ";
            p.print();
            cout << "Radius: " << radius << endl;
        }
};
```

```
int main()
{
    Point p(5,5);
    Circle c(2, 3, 5);

    cout << endl << "Point Display" <<endl;
    p.print();
    cout << endl << "Circle Display" <<endl;
    c.print();
    cout << endl;


//First update
    cout << "First Update" << endl;

// call “update” function of Circle class to increase the center’s x
coordinate by 5 and y coordinate by 5;

    c.print();
    cout <<endl;

    //Second update
    cout << "Second Update" << endl;

// call “update” function of Circle class to increase the radius by 6;

    c.print();
    cout << endl;

    //Third update
    cout << "Third Update" << endl;

// call “update” function of Circle class to increase the center’s x
coordinate by 2 and y coordinate by 2 and the radius by 2;

    c.print();
    cout << endl;
    return 0;
}
```

Output:

Point Display

Coordinate: 5, 5

Circle Display

Center Coordinate: 2, 3

Radius: 5

First Update

Center Coordinate: 7, 8

Radius: 5

Second Update

Center Coordinate: 7, 8

Radius: 11

Third Update

Center Coordinate: 9, 10

Radius: 13