Date:

Assignment No.:

• Problem Statement:

Program in C++ to create a class POINT having a member function which takes coordinates of a point from the user as the input. Using that class, derive three different classes line, triangle and square. The base class POINT has a member function display() which prints the x and y coordinate of the point. In the derived class line, triangle and square we override the display function of the base class and display the length of the line, perimeter of the triangle and perimeter of square in their respective display function. Use the concept of function overloading.

• Algorithm:

➔ **Algorithm for method len(x1, y1, x2, y2)**:

Return sqrt(((x1 - x2) * (x1 - x2)) + ((y1 - y2) * (y1 – y2)))

//sqrt() is a function to calculate the square root of a number
//passed to it as argument

➔ **Name of the class:** Point

Private data members: x, y

Public member functions of the class:

get() : To read the coordinate of a point from the user.

display() : To display the coordinate of the point read from the user.

Algorithm for method get():

Step 1: Print "Enter coordinates of the point : "
Step 2: Print " x : "
Step 3: Read x
Step 4: Print " y : "
Step 5: Read y

Algorithm for method display()

Step 1: Print "The point is : " x ", " y

➔ **Name of the class:** Line

Derived from: class Point

Private data members: x1, y1, x2, y2

Public member functions of the class:

get() : To read the coordinates of a line from the user.. //Overridden the
//get() method of the base class

display() : To display the coordinates of the line read from the user.
//Overridden the display()
//method of the base class

Date:

Algorithm for method get():

Step 1: Print "Enter coordinates of the line : "
Step 2: Print " x1 : "
Step 3: Read x1;
Step 4: Print " y1 : "
Step 5: Read y1
Step 6: Print " x2 : "
Step 7: Read x2;
Step 8: Print " y2 : "
Step 9: Read y2

Algorithm for method display():

Step 1: Set length = len(x1, y1, x2, y2)
Step 2: Print "The length of the line is : " length

➔ **Name of the class:** Triangle

Derived from: class Point

Private data members: x1,y1,x2,y2,x3,y3

Public member functions of the class:

get() : To read the coordinates of a triangle from the user.
//Overridden the
//get() method of the base class

display() : To display the coordinates of the triangle read from the user.
//Overridden the display()
//method of the base class

Algorithm for method get():

Step 1: Print "Enter coordinates of the triangle : "
Step 2: Print " x1 : "
Step 3: Read x1
Step 4: Print " y1 : "
Step 5: Read y1
Step 6: Print " x2 : "
Step 7: Read x2
Step 8: Print " y2 : "
Step 9: Read y2
Step 10: Print " x3 : "
Step 11: Read x3
Step 12: Print " y3 : "
Step 13: Read y3

Algorithm for method display():

Step 1: l1 = len(x1, y1, x2, y2)
Step 2: double l2 = len(x1, y1, x3, y3)
Step 3: double l3 = len(x2, y3, x3, y3)
Step 4: peri= l1 + l2 + l3
Step 5: Print "Perimeter of the triangle : " peri

Date:

➔ **Name of the class:** Square

Derived from: class Point

Private data members: x1, y1, x2, y2, x3, y3, x4, y4

Public member functions of the class:

get() : To read the coordinates of a square from the user
//Overridden the
//get() method of the base class

display() : To display the coordinates of the square read from the user.
//Overridden the
//display() method of the base class

Algorithm for method get():

Step 1: Print "Enter coordinates of the square : "
Step 2: Print " x1 : "
Step 3: Read x1
Step 4: Print " y1 : "
Step 5: Read y1
Step 6: Print " x2 : "
Step 7: Read x2
Step 8: Print " y2 : "
Step 9: Read y2
Step 10: Print " x3 : "
Step 11: Read x3
Step 12: Print " y3 : "
Step 13: Read y3
Step 14: Print " x4 : "
Step 15: Read x4
Step 16: Print " y4 : "
Step 17: Read y4

Algorithm for method display():

Step 1: double l1 = len(x1, y1, x2, y2);
Step 2: double l2 = len(x2, y2, x3, y3);
Step 3: double l3 = len(x3, y3, x4, y4);
Step 4: double l4 = len(x4, y4, x1, y1);
Step 5: peri= l1 + l2 + l3 + l4
Step 6: Print "Perimeter of the square : " peri

➔ **Algorithm for method main():**

Step 1: Create an object p of class Point
Step 2: Call method get() of class Point for object p
Step 3: Call method display() of class Point for object p
Step 4: Create an object l of class Line
Step 5: Call method get() of class Point for object l

Date:

Step 6: Call method display() of class Point for object l
Step 7: Create an object t of class Triangle
Step 8: Call method get() of class Point for object t
Step 9: Call method display() of class Point for object t
Step 10: Create an object s of class Square
Step 11: Call method get() of class Point for object s
Step 12: Call method display() of class Point for object s

- **Source Code:**

```
#include <iostream>
#include <math.h>

using namespace std;

class Point {
   private:
      int x, y;
   public:
      virtual void get() {
         cout << "Enter coordinates of the point : " << endl;
         cout << " x : ";
         cin >> x;
         cout << " y : ";
         cin >> y;
      }

      virtual void display() {
         cout << "The point is : " << x << ", " << y << endl;
      }
};

#define LENGTH(x1, y1, x2, y2) sqrt(((x1 - x2) * (x1 - x2)) + ((y1 - y2) * (y1 -
y2)))

class Line : public Point {
   private:
      int x1, y1, x2, y2;
   public:
      virtual void get() {
         cout << "Enter coordinates of the line : " << endl;
         cout << " x1 : ";
         cin >> x1;
         cout << " y1 : ";
         cin >> y1;
         cout << " x2 : ";
         cin >> x2;
         cout << " y2 : ";
```

```
        cin >> y2;
    }

    virtual void display() {
        double length = LENGTH(x1, y1, x2, y2);
        cout << "The length of the line is : " << length << endl;
    }
};

class Triangle : public Point {
  private:
    int x1, y1, x2, y2, x3, y3;
  public:
    virtual void get() {
        cout << "Enter coordinates of the triangle : " << endl;
        cout << " x1 : ";
        cin >> x1;
        cout << " y1 : ";
        cin >> y1;
        cout << " x2 : ";
        cin >> x2;
        cout << " y2 : ";
        cin >> y2;
        cout << " x3 : ";
        cin >> x3;
        cout << " y3 : ";
        cin >> y3;
    }

    virtual void display() {
        double l1 = LENGTH(x1, y1, x2, y2);
        double l2 = LENGTH(x1, y1, x3, y3);
        double l3 = LENGTH(x2, y3, x3, y3);
        cout << "Perimeter of the triangle : " << (l1 + l2 + l3) << endl;
    }
};

class Square : public Point {
  private:
    int x1, y1, x2, y2, x3, y3, x4, y4;
  public:
    virtual void get() {
        cout << "Enter coordinates of the square : " << endl;
        cout << " x1 : ";
        cin >> x1;
        cout << " y1 : ";
        cin >> y1;
        cout << " x2 : ";
        cin >> x2;
        cout << " y2 : ";
        cin >> y2;
```

```
        cout << " x3 : ";
        cin >> x3;
        cout << " y3 : ";
        cin >> y3;
        cout << " x4 : ";
        cin >> x4;
        cout << " y4 : ";
        cin >> y4;
    }

    virtual void display() {
        double l1 = LENGTH(x1, y1, x2, y2);
        double l2 = LENGTH(x2, y2, x3, y3);
        double l3 = LENGTH(x3, y3, x4, y4);
        double l4 = LENGTH(x4, y4, x1, y1);
        cout << "Perimeter of the square : " << (l1 + l2 + l3 + l4) << endl;
    }

};

int main() {
    Point p;
    p.get();
    p.display();

    Line l;
    l.get();
    l.display();

    Triangle t;
    t.get();
    t.display();

    Square s;
    s.get();
    s.display();

    return 0;
}
```

- ## Input & Output:

Enter coordinates of the point :

x : 10

y : 20

The point is : 10, 20

Date:

Enter coordinates of the line :

x1 : 15

y1 : 15

x2 : 23

y2 : 30

The length of the line is : 17

Enter coordinates of the triangle :

x1 : 15

y1 : 15

x2 : 23

y2 : 30

x3 : 50

y3 : 25

Perimeter of the triangle : 80.4005

Enter coordinates of the square :

x1 : 10

y1 : 20

x2 : 10

y2 : 40

x3 : 50

y3 : 40

x4 : 50

y4 : 20

Perimeter of the square : 120

Date:

- ## Discussion:

1. Here we used Inheritance and Inherited functions work slower than normal function as there is indirection. Also,innheritance increases the coupling between base class and derived class. A change in base class will affect all the child classes.

2. The perimeter calculation can further be improved by rounding off the result.