```
 1 /*****************************************************************************
 2  * PROGRAMMER : Ali Eshghi
 3  * STUDENT ID : 1112261
 4  * CLASS      : CS1C
 5  * SECTION    : MW 5pm
 6  * Assign #4  : Enhanced Employee
 7  * DUE DATE   : 26 Febuary 2020
 8  *****************************************************************************/
 9 #ifndef MYHEADER_H_
10 #define MYHEADER_H_
11
12 //Preprocessor directives
13
14 #include <iostream> //for input and output
15 #include <cmath>//for math equations
16
17 //using the name space standard
18 using namespace std;
19
20 //class shape: base class for the sub classes with public attributes
21 class Shape
22 {
23 //public parts containing the method functions of the class
24 public:
25
26     //virtual method for calculating perimeter
27     virtual double calcPerimeter() = 0;
28
29     //virtual method for calculating Are
30     virtual double calcArea() = 0;
31 };
32
33 //class Square: class for the square attributes
34 //inherit from the shape class
35 class Square : public Shape
36 {
37
38 //private attributes of class aquare
39 private:
40     double length;
41
42 //public functions of class Square
43 public:
```

```
44
45      //Constructor
46      Square(double _length = 0)
47      {
48      length = _length;
49      }
50
51      //method for setting the length
52      void setLength(double l)
53      {
54          length = l;
55      }
56
57      //method for claculating perimeter
58      double calcPerimeter()
59      {
60          return (length * 4);
61      }
62
63      //method for calculating area
64      double calcArea()
65      {
66          return length * length;
67      }
68  };
69
70  //Class Triangle: contains the attributes of triangle
71  //inherit from the class Shape
72  class Triangle : public Shape
73  {
74
75  //private attributes of the class
76  private:
77      double sideA, sideB, sideC;
78
79  //public functions of the class triangle
80  public:
81
82      //constructor
83      Triangle(double _sideA = 0, double _sideB = 0, double _sideC = 0)
84      {
85      sideA = _sideA;
86      sideB = _sideB;
```

```
    sideC = _sideC;
    }

    //Method for setting first side of the triabgle
    void setSideA(double sideA)
    {
        Triangle::sideA = sideA;
    }

    //method for setting the second side of the triangle
    void setSideB(double sideB)
    {
        Triangle::sideB = sideB;
    }

    //method for setting the third side of the triangle
    void setSideC(double sideC)
    {
        Triangle::sideC = sideC;
    }

    //method for calculating the perimeter
    double calcPerimeter()
    {
        return sideA + sideB + sideC;
    }

    //method for calculating the area
    double calcArea(){
        double p = calcPerimeter()/2;//Get half perimeter
        return sqrt(p * (p - sideA) * (p - sideB) * (p - sideC));
    }
};

//function for printing the perimeter
void printPerimeter(Shape &obj){
    cout << "Perimeter: " << obj.calcPerimeter() << endl;
}

//function for printing area
void printArea(Shape &obj){
    cout << "Area: " << obj.calcArea() << endl;
}
```

```
130
131
132 #endif /* MYHEADER_H_ */
133
```