

MyHeader.h

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1C
5  * SECTION    : MW 5pm
6  * Assign #4  : Friends and overloading
7  * DUE DATE   : 10 March 2020
8  *****/
9
10 #ifndef MYHEADER_H_
11 #define MYHEADER_H_
12
13 //Preprocessor directives
14
15 #include <iostream> //for input and output
16 #include <math.h>   //for math equations
17
18 //using the name space standard
19 using namespace std;
20
21 //class shape: base class for the sub classes with public attributes
22 class Shape
23 {
24 //public parts containing the method functions of the class
25 public:
26
27     //virtual method for calculating perimeter
28     virtual float calcPerimeter() = 0;
29
30     //virtual method for calculating Area
31     virtual float calcArea() = 0;
32
33     // to check perimeter of different types of shapes
34     friend bool equalPer(Shape &x, Shape &y);
35 };
36
37 //function equalPer: checks if the perimeter of two shapes are equal
38 //return type: bool
39 bool equalPer(Shape &x, Shape &y)
40 {
41     if(x.calcPerimeter() == y.calcPerimeter())
42         return true;
43     else
44         return false;
45 }
46
47 //class Square: class for the square attributes
48 //inherit from the shape class
49 class Square : public Shape
50 {
51     float length;
52
53 //public functions of class Square
54 public:
55
56     //Constructor
57     Square(float l)
58     {
59         length=l;
```

```

60     }
61
62     //method for claculating perimeter
63     float calcPerimeter()
64     {
65         return length * 4;
66     }
67
68     //method for calculating area
69     float calcArea()
70     {
71         return length*length;
72     }
73
74     //method for printing perimeter
75     void printPerimeter()
76     {
77         cout<<"Perimeter of Square is "<<calcPerimeter()<<"\n";
78     }
79
80     //method for printing area
81     void printArea()
82     {
83         cout<<"Area of Square is "<<calcArea()<<"\n";
84     }
85
86     //method for addition
87     void addition(int x)
88     {
89         length += x;
90         cout<<"Length increased by "<<x<<" units\n";
91     }
92     // overloading operator ==
93     friend bool operator==(Square &r1, Square &r2);
94
95     // overloading operator +
96     friend Square operator+(Square &r1, int x);
97
98     // overloading operator <<
99     friend ostream& operator<< (ostream &out, Square &point);
100
101     // overloading operator >>
102     friend istream& operator>> (istream &in, Square &point);
103 };
104
105 //function to overload the == operator
106 //return type : bool
107 bool operator==(Square &r1, Square &r2)
108 {
109     return (r1.calcArea() == r2.calcArea());
110 }
111
112 //function to overload the + operator
113 //return type : square class type variable
114 Square operator+(Square &r, int x)
115 {
116     r.length += x;
117     cout<<"Length increased by "<<x<<" units\n";
118     return r;

```

```

119 }
120
121
122 ostream& operator<< (ostream &out, Square &r)
123 {
124     out<<"Length of Square : "<<r.length;
125     out<<endl;
126     r.printPerimeter();
127     r.printArea();
128     return out;
129 }
130
131 istream& operator>> (istream &in, Square &r)
132 {
133     cout<<"Enter length : ";
134     cin>>r.length;
135     r.printArea();
136     r.printArea();
137 }
138
139 class Triangle : public Shape
140 {
141     float side1,side2,side3;
142
143 public:
144     Triangle(float a,float b,float c)
145     {
146         side1=a;
147         side2=b;
148         side3=c;
149     }
150
151     float calcPerimeter()
152     {
153         return side1+side2+side3;
154     }
155
156     float calcArea()
157     {
158         float s=calcPerimeter()/2;
159         float area = sqrt(s*(s-side1)*(s-side2)*(s-side3));
160         return area;
161     }
162
163     void printPerimeter()
164     {
165         cout<<"Perimeter of Triangle is "<<calcPerimeter()<<"\n";
166     }
167
168     void printArea()
169     {
170         cout<<"Area of Triangle is "<<calcArea()<<"\n";
171     }
172 };
173
174
175
176 #endif /* MYHEADER_H_ */
177

```