```
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi
 * STUDENT ID : 1112261
 * CLASS      : CS1C
 * SECTION    : MW 5pm
 * Assign #7  : Exception
 * DUE DATE   : 23 March 2020
 *****************************************************************************/

#ifndef MYHEADER_H_
#define MYHEADER_H_


//Preprocessor directives

#include <iostream> //for input and output
#include <math.h>   //for math equations

//using the name space standard
using namespace std;

//class shape: base class for the sub classes with public attributes
class Shape
{
//public parts containing the method functions of the class
public:

    //virtual method for calculating perimeter
    virtual float calcPerimeter() = 0;

    //virtual method for calculating Are
    virtual float calcArea() = 0;

    // to check perimeter of different types of shapes
    friend bool equalPer(Shape &x, Shape &y);
};

//function equalPer: checks if the perimeter of two shapes are equal
//return type: bool
bool equalPer(Shape &x, Shape &y)
{
    if(x.calcPerimeter() == y.calcPerimeter())
        return true;
    else
        return false;
}

//class Square: class for the square attributes
//inherit from the shape class
class Square : public Shape
{
    float length;

//public functions of class Square
public:

    //Constructor
    Square(float l)
    {
```

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

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

```
178
179
180 #endif /* MYHEADER_H_ */
181
```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi
 * STUDENT ID : 1112261
 * CLASS      : CS1C
 * SECTION    : MW 5pm
 * Assign #7  : Exception
 * DUE DATE   : 23 March 2020
 *****************************************************************************/

#include "MyHeader.h"

int main(int argc, char *argv[])
{
    /*************************************************************
     * Perimeter and Area of shapes comparing using
     * overloading frined functions and class methods
     * ___________________________________________________________
     * This program prompts the user to input information for a
     * shape of square and a shape of triangle. then using the
     * virtual methods and inheritance of the classes, calculates
     * the perimeter and area of those shapes, then using the
     * frined class functions and class methods, the program
     * compares the area and perimeter of the squares with the
     * area and perimeter of the triangle. then outputs if any
     * attributes of these shapes were equal, then again promts the
     * user how many units they want to increase the length of both
     * shapes and again print outs the attribute of the shapes
     *
     *Also, using the try/catch blocks, the program determines if the
     *input numbers by the user are correct or no.
     * ___________________________________________________________
     * INPUT: square length, triangle sides
     *
     * OUTPUT: prints out the perimeter and area of the both shapes
     *         based on the information that the user put in, the
     *         result of the comparison of the attributes, and the
     *         new attributes after the modification
     *
     *************************************************************/


    cout << "/*************************************************************\n"
         << "* Perimeter and Area of shapes comparing using\n"
         << "* overloading frined functions and class methods\n"
         << "* ___________________________________________________________\n"
         << "* This program prompts the user to input information for a\n"
         << "* shape of square and a shape of triangle. then using the\n"
         << "* virtual methods and inheritance of the classes, calculates\n"
         << "* the perimeter and area of those shapes, then using the\n"
         << "* frined class functions and class methods, the program\n"
         << "* compares the area and perimeter of the squares with the\n"
         << "* area and perimeter of the triangle. then outputs if any\n"
         << "* attributes of these shapes were equal, then again promts the\n"
         << "* user how many units they want to increase the length of both\n"
         << "* shapes and again print outs the attribute of the shapes\n"
         << "*\n"
         << "* ___________________________________________________________\n"
         << "* INPUT: square length, triangle sides\n"
         << "*\n"
```

```
60          << "* OUTPUT: prints out the perimeter and area of the both shapes\n"
61          << "*         based on the information that the user put in, the\n"
62          << "*         result of the comparison of the attributes, and the\n"
63          << "*         new attributes after the modification\n"
64          << "*\n"
65          << "*****************************************************************/\n\n";
66
67
68 // check if 3 arguments are provided are not
69 // argv[0] is program name
70 // argv[1] is first name, argv[2] is last name
71 if(argc < 3)
72 {
73 cout<<"Arguments not provided correctly\n";
74 return 1;
75 }
76
77
78 // display first and last name
79 cout<<"First Name : "<<argv[1];
80 cout<<"\nLast Name : "<<argv[2];
81
82
83 // display second characters
84 // since argv[] is a char array we can directly access it's second char by [1] index
85 cout<<"\nSecond Character of First Name : "<<argv[1][1];
86 cout<<"\nSecond Character of Last Name : "<<argv[2][1];
87 // PART B ends
88
89 // input 3 squares and triangles for sample testing
90 float length,side1,side2,side3;
91
92 //begin try block
93 try
94 {
95     cout << "\n\nTRY BLOCK #1: ";
96
97     cout<<"\n\nEnter the length of 1st Square:";
98     cin>>length;
99
100    throw length;
101 }
102
103 catch(float length)
104 {
105     cout << "\n\nCATCHBLOCK #1: ";
106
107     while(length < 0 || length > 50)
108     {
109
110
111         if(length < 0)
112         {
113             cout << "\nThe side length: " << length
114                  << " is too small.\nPlease enter a value greater than 0.\n\n";
115         }
116
117         else if(length > 50)
118         {
```

```
119             cout << "\nThe side length: " << length
120                  << " is too big.\nPlease enter a value smaller than 50.\n\n";
121         }
122 
123         cout<<"\n\nEnter the length of 1st Square:";
124         cin>>length;
125         cin.ignore(10000,'\n');
126     }
127 
128 }
129 
130 cout << "\n<<Valid side length input>>\n\n";
131 
132 Square r1(length);
133 r1.printPerimeter();
134 r1.printArea();
135 cout<<"Enter the three sides of 1st triangle:";
136 cin>>side1>>side2>>side3;
137 Triangle s1(side1,side2,side3);
138 s1.printPerimeter();
139 s1.printArea();
140 
141 
142 //begin try block
143 try
144 {
145     cout << "\n\nTRY BLOCK #2: ";
146 
147     cout<<"\n\nEnter the length and width of 2nd square:";
148     cin>>length;
149 
150     throw length;
151 }
152 
153 catch(float length)
154 {
155     cout << "\n\nCATCHBLOCK #1: ";
156 
157     while(length < 0 || length > 50)
158     {
159 
160 
161         if(length < 0)
162         {
163             cout << "\nThe side length: " << length
164                  << " is too small.\nPlease enter a value greater than 0.\n\n";
165         }
166 
167         else if(length > 50)
168         {
169             cout << "\nThe side length: " << length
170                  << " is too big.\nPlease enter a value smaller than 50.\n\n";
171         }
172 
173         cout<<"\n\nEnter the length of 2nd Square:";
174         cin>>length;
175         cin.ignore(10000,'\n');
176     }
177 }
```

```
178
179 cout << "\n<<Valid side length input>>\n\n";
180
181 Square r2(length);
182 r2.printPerimeter();
183 r2.printArea();
184 cout<<"Enter the three sides of 2nd triangle:";
185 cin>>side1>>side2>>side3;
186 Triangle s2(side1,side2,side3);
187 s2.printPerimeter();
188 s2.printArea();
189
190 //begin try block
191 try
192 {
193     cout << "\n\nTRY BLOCK #3: ";
194
195     cout<<"\n\nEnter the length of 3rd square:";
196     cin>>length;
197
198     throw length;
199 }
200
201 catch(float length)
202 {
203     cout << "\n\nCATCHBLOCK #3: ";
204
205     while(length < 0 || length > 50)
206     {
207
208
209         if(length < 0)
210         {
211             cout << "\nThe side length: " << length
212                  << " is too small.\nPlease enter a value greater than 0.\n\n";
213         }
214
215         else if(length > 50)
216         {
217             cout << "\nThe side length: " << length
218                  << " is too big.\nPlease enter a value smaller than 50.\n\n";
219         }
220
221         cout<<"\n\nEnter the length of 3rd square:";
222         cin>>length;
223         cin.ignore(10000,'\n');
224     }
225 }
226
227 Square r3(length);
228 r3.printPerimeter();
229 r3.printArea();
230 cout<<"Enter the three sides of 3rd triangle:";
231 cin>>side1>>side2>>side3;
232 Triangle s3(side1,side2,side3);
233 s3.printPerimeter();
234 s3.printArea();
235
236
```

```
237 // Testing for equal perimeters
238 cout<<"\nSquare 1 and Triangle 1 Perimeter Check : ";
239 if(equalPer(r1, s1))
240 {
241 cout<<"Equal\n";
242 }
243 else
244 cout<<"Unequal\n";
245
246 cout<<"\nSquare 2 and Triangle 2 Perimeter Check : ";
247 if(equalPer(r2, s2))
248 {
249 cout<<"Equal\n";
250 }
251 else
252 cout<<"Unequal\n";
253
254
255 // Testing rectangles for equal area
256 cout<<"\nSquare 1 and Rectangle 2 Area Check : ";
257 if(r1 == r2)
258 {
259 cout<<"Equal\n";
260 }
261 else
262 cout<<"Unequal\n";
263
264 cout<<"\nSquare 2 and Rectangle 3 Area Check : ";
265 if(r2 == r3)
266 {
267 cout<<"Equal\n";
268 }
269 else
270 cout<<"Unequal\n";
271
272
273 // testing addition member function
274 int x;
275 cout<<"\nIncrease length of Square 1 by how much : ";
276 cin>>x;
277 r1.addition(x);
278
279
280 // testing overloaded + operator
281 cout<<"\nAfter doing r2 = r2 + 8 :\n";
282 r2 = r2 + 8;
283
284
285 // testing >> overloaded operator
286 cout<<"\nAfter doing cin>>r3 :\n";
287 cin>>r3;
288
289 // testing << overloaded operator
290 cout<<"\nAfter doing cout<<r3 :\n";
291 cout<<r3;
292
293 return 0;
294 }
295
```

296
297
298