```cpp
/*
Lab 2
CS 2
Date: FIXME
Updated By: FIXME

Object Oriented Programming about triangles using Object Oriented Design.

This program reads three sides of triangles from an input file.
The program calculates various properties of triangles such as
area, perimeter, type, etc. using class.
The program creates and output report file with all the information calculated.
*/

#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
#include "Triangle.h"

using namespace std;
#define MAX_TRIANGLES 100

// function to check the data stored in triangles array
void checkData(Triangle ts[], int arrayLength);

// This function reads triangle information from input file into
// the triangles array provided
// returns total number of triangles' read from the file
int readTriangles(Triangle ts[]);

// This function write triangle information as a report to an output file
// provided by the user
void writeReport(Triangle ts[], int arrayLen);

// This function draws a line with give char ch of length len
void drawLine(ofstream &fout, char ch, int len);

// Function to sort triangles in ascending order
// based on their area using bubble sort algorithm
void sortTriangles(Triangle triangles[], int arrayLen);

int main(int argc, char *argv[])
{
	// Declare array of triangles that can hold upto MAX_TRIANGLES triangles
	Triangle triangles[MAX_TRIANGLES];
	// set output for floating point numbers to always display up to 2 decimal points
	cout << fixed << showpoint << setprecision(2);
	//checkData(triangles, MAX_TRIANGLES);

	int triangleCount = readTriangles(triangles);
	// Data before sorting
	cout << "Before sorting\n\n";
	checkData(triangles, triangleCount);
	// FIXME
	// Call sortTriangles function by passing proper arguments

	cout << "After sorting\n";
	checkData(triangles, triangleCount);
	writeReport(triangles, triangleCount);
	cout << "All done! Hit enter to exit...";
	cin.get();
	return 0;
}
```

```
65
66  void checkData(Triangle ts[], int arrayLength)
67  {
68        double side1, side2, side3;
69        for (int i = 0; i < arrayLength; i++)
70        {
71              ts[i].getSides(side1, side2, side3);
72              cout << i + 1 << ". " << side1 << " " << side2 << " " << side3 << " " <<
73  ts[i].getArea() << endl;
74        }
75  }
76
77  int readTriangles(Triangle ts[])
78  {
79        int count = 0;
80        double side1, side2, side3;
81        string file;
82        cout << "Enter file name with triangels' info in it:\n";
83        getline(cin, file);
84        ifstream fin;
85        fin.open(file);
86        if (!fin)
87        {
88              cout << "Input file does not exist. Program Terminates!";
89              cin.get();
90              return 1;
91        }
92        int i=0;
93        while(i < MAX_TRIANGLES && !fin.eof())
94        {
95              fin >> side1 >> side2 >> side3;
96              // FIXME
97              // set there sides' length for i^th triangle in ts array
98              ++i;
99        }
100       return i; // return total number of triangles read from input file
101 }
102
103 void sortTriangles(Triangle triangles[], int arrayLen)
104 {
105       // Use improved version of bubble sort to sort
106       // Triangle objects stored in triangles array based on area
107       bool inOrder = true;
108       for (int i = 0; i < arrayLen; i++)
109       {
110             inOrder = true;
111             for (int j = 0; j < arrayLen - i - 1; j++)
112             {
113                   cout << triangles[j].getArea() << endl;
114                   if (triangles[j].getArea() > triangles[j+1].getArea())
115                   {
116                         inOrder = false;
117                         // FIXME
118                         // swap two triangles as they're not in order
119                   }
120             }
121             if (inOrder) // data is all sorted
122                   break;
123       }
124 }
125
126 void writeReport(Triangle ts[], int arrayLen)
127 {
128       string file;
```

```
129         ofstream fout;
130         double side1, side2, side3;
131         do
132         {
133                 cout << "Enter a file name to write output to:\n";
134                 getline(cin, file);
135                 fout.open(file);
136                 if (!fout)
137                         cout << "Output file could NOT be opened. Try again…\n";
138         } while (!fout);
139
140         fout << fixed << showpoint << setprecision(2);
141         int len = 65;
142         drawLine(fout, '*', len);
143         fout << setw(40) << "Triangle Information" << endl;
144         // FIXME
145         // Draw line with * character
146         fout << setw(5) << "#" << setw(8) << "side 1" << setw(8) <<
147                 "side 2" << setw(8) << "side 3" << setw(8) << "area" <<
148                 setw(10) << "perimeter" << setw(17) << "type" << endl;
149         drawLine(fout, '=', len);
150         for (int i = 0; i < arrayLen; i++)
151         {
152                 // FIXME
153                 // Output each triangle's information as shown in the sample output.
154                 // Use proper formatting
155                 fout << "FIXEME" << endl; // comment this out when fixed
156         }
157         // FIXME
158         // Draw line with = character
159         cout << "Done writing data to output file!\n";
160 }
161
162 void drawLine(ofstream &fout, char ch, int len)
163 {
164         fout << setfill(ch);
165         fout << setw(len) << "" << endl;
166         fout << setfill(' '); //reset fill character to space
167 }
```