

Arithmetic Calculations

For the second programming project, you will be given most of the C++ source code for a working program. The given source code includes all the necessary variable declarations and the mandatory input and output statements. Your task is to complete the given program by adding the C++ statements needed to perform the specified calculations.

You may modify the given program source code as much as you like. At minimum you must add the calculations. Just be careful if you make other changes since your output must match the formatting of the sample output exactly.

Sample input data:

Here is a sample input file, named `RectangleData.txt`, for the program. The file specifies the coordinates of two opposite corners of a rectangle that is to be drawn on a video screen. Note that in this coordinate system, the x-axis runs along the top of the screen and the y-axis runs down the left side of the screen, with the origin at the upper-left (or northwest) corner of the screen. So, y-values increase as you go down, not up.

The first line specifies the northeast corner and the second the southwest corner of the rectangle. Each corner is expressed using the customary mathematical notation for a point, and is preceded by a descriptive label that ends with a colon character ': '.

```
NE corner:    ( 80,155)
SW corner:    ( 39,248)
```

Calculations and sample output data:

Here is a sample output file, named `Analysis.txt`, for the program. It begins by identifying the programmer (you) and the specific project. The remainder of the output file reports the results computed by the program.

There will be six lines of output, each specifying a labeled value computed by the program:

- the coordinates of the center of the rectangle, expressed in point notation with each coordinate written in a field that is three columns wide.
- the height of the rectangle.
- the width of the rectangle.
- the area of the rectangle.
- the perimeter of the rectangle.
- the aspect ratio of the rectangle, written with precision 2.

The aspect ratio of a rectangle is the ratio of the height to the width.

The labels must be precisely as shown in the sample output. The output should be aligned for easy readability.

```
Programmer: Mir Farooq Ali
CS 1044 Spring 2004 Project 2

Center      ( 59,201)
Height      93
Width       41
Area        3813
Perimeter   268
Aspect ratio 2.27
```

Additional samples of input and correct output are available on the course website.

Evaluation:

Everything that was said in the specification for Project 1 about testing still applies here. Do not waste submissions to the Curator in testing your program! There is no point in submitting your program until you have verified that it produces correct results on the sample data files that are provided. If you waste all of your submissions because you have not tested your program adequately then you will receive a low score on this assignment. You will not be given extra submissions.

Your submitted program will be assigned a score based upon the runtime testing performed by the Curator System. We will not be evaluating your submission of this program for documentation style. However, you should examine the given program as a guide to acceptable documentation, and include similar comments for the statements you add to it.

For a number of the later projects, we will also evaluate your submission for documentation, and for other requirements. It is best to begin preparing for that now.

Submitting your program:

You will submit this assignment to the Curator System (read the *Student Guide*), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guide*.

You will be allowed up to five submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest score you achieve will be counted.

The *Student Guide* and the submission link can be found at: <http://www.cs.vt.edu/curator>

Pledge:

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the header comment for your program:

```
// On my honor:
//
// - I have not discussed the C++ language code in my program with
//   anyone other than my instructor or the teaching assistants
//   assigned to this course.
//
// - I have not used C++ language code obtained from another student,
//   or any other unauthorized source, either modified or unmodified.
//
// - If any C++ language code or documentation used in my program
//   was obtained from another source, such as a text book or course
//   notes, that has been clearly noted with a proper citation in
//   the comments of my program.
//
// - I have not designed this program in such a way as to defeat or
//   interfere with the normal operation of the Curator System.
//
// <Student Name>
```

Failure to include this pledge in a submission is a violation of the Honor Code.

The program source code:

```
// Project 2 for CS 1044 Spring 2004
//
// Student:      <put your name here>
//
// Programmer:   William McQuain
// Modified by:  Mir Farooq Ali
// OS:          Windows XP - Tablet Version
// System:       Tablet PC
// Compiler:     Visual Studio .NET 2002
// Last modified: February 10, 2004
//
// Purpose
// This program computes some statistics for a rectangle, given
// the screen coordinates of its NW and SE corners.  The computed
// values are:
//
//   - coordinates of the center
//   - height and width
//   - area and perimeter
//   - aspect ratio
//
// The program then writes a summary of its findings to an output
// file.
//
#include <fstream>
#include <iostream>
#include <iomanip>
#include <string>
#include <climits>
using namespace std;

int main() {

    ifstream dataFile("RectangleData.txt");
    ofstream logFile("Analysis.txt");

    int xNE = 0,          // x coordinate of NE corner
        yNE = 0,          // y coordinate of NE corner
        xSW = 0,          // x coordinate of SW corner
        ySW = 0;          // y coordinate of SW corner

    int Height = 0,      // height of rectangle
        Width = 0,       // width of rectangle
        Area = 0,        // area of rectangle
        Perimeter = 0,   // perimeter of rectangle
        xCenter = 0,     // x coordinate of rectangle center
        yCenter = 0;     // y coordinate of rectangle center

    double aspectRatio = 0.0;

    // Write an identification header to the output file:
    logFile << "Programmer: "
              << "<insert your name here>"           // Change this to YOUR name.
              << endl;
    logFile << "CS 1044 Spring 2004 Project 2"
              << endl
              << endl;
```

```
// Read the coordinates of the NW corner:
dataFile.ignore(INT_MAX, '(');
dataFile >> xNE;
dataFile.ignore(INT_MAX, ',');
dataFile >> yNE;
dataFile.ignore(INT_MAX, '\n');

// Read the coordinates of the SE corner:
dataFile.ignore(INT_MAX, '(');
dataFile >> xSW;
dataFile.ignore(INT_MAX, ',');
dataFile >> ySW;
dataFile.ignore(INT_MAX, '\n');

// Check for an invalid rectangle:
if ( (xNE < 0) || (yNE < 0) || (xSW < 0) || (ySW < 0) ) {
    logFile << "The given rectangle coordinates are impossible." << endl;
    return 0;
}

if ( (xSW >= xNE) || (ySW <= yNE) ) {
    logFile << "The given rectangle coordinates are impossible." << endl;
    return 0;
}

// You must add the following calculations:
// Calculate the height and width:

// Calculate the center:

// Calculate the area:

// Calculate the aspect ratio:

// Calculate the perimeter:

// Write out the analysis:
logFile << endl;
logFile << "Center          " << '(' << setw(3) << xCenter
        << ',' << setw(3) << yCenter
        << ')' << endl;
logFile << "Height          " << setw(8) << Height << endl;
logFile << "Width           " << setw(8) << Width << endl;
logFile << "Area            " << setw(8) << Area << endl;
logFile << "Perimeter       " << setw(8) << Perimeter << endl;
logFile << fixed << showpoint;
logFile << "Aspect ratio  " << setw(8) << setprecision(2)
        << aspectRatio << endl;

dataFile.close();
logFile.close();

return 0;
}
```