

## 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 `CircleData.txt`, for the program. The file specifies the center coordinates of the circle and its radius. The first line specifies the center coordinates and the second line specifies the radius.

```
Circle's center:    (40, 40)
Circle's radius:    40
```

### 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 some a few lines of output, each specifying a labeled value computed by the program:

- the coordinates of the top-right, top-left, bottom-right and bottom-left of the bounding box of the circle, expressed in point notation with each coordinate written in a field that is three columns wide.
- the area of the circle.
- the perimeter of the circle.
- the area of the bounding box.
- the perimeter of the bounding box.

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 Summer II 2004 Project 2

Bounding Box Coordinates
-----
Top Right:    ( 80, 80)
Top Left:     (  0, 80)
Bottom Left:  (  0,  0)
Bottom Right: ( 80,  0)

Areas and Perimeters
-----
Circle Area:           5024.00
Circle Perimeter:      251.20
Bounding box Area:     6400.00
Bounding box Perimeter: 320.00
```

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://eags01.cs.vt.edu:8080/CuratorS04/index.jsp>

**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 Summer II 2004
//
// Student:      <put your name here>
//
// Programmer:   Mir Farooq Ali
// OS:           Windows XP - Tablet Version
// System:       Tablet PC
// Compiler:     Visual Studio .NET 2002
// Last modified: July 15, 2004
//
// Purpose
// This program computes the coordinates of a bounding box of a circle
// given its center and radius. It also calculates the area and perimeter
// of the circle and the bounding box. 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() {

    const double PI = 3.14; // Set the value of PI

    ifstream dataFile("CircleData.txt"); // Open the input file stream
    ofstream logFile("Analysis.txt");     // Open the output file stream

    int xCenter,          // X-coordinate of the circle's center
        yCenter,          // Y-coordinate of the circle's center
        radius;           // Radius of the circle

    int xTopRight,       // X-coordinate of the top right corner of bounding box
        yTopRight,       // Y-coordinate of the top right corner of bounding box
        xTopLeft,        // X-coordinate of the top left corner of bounding box
        yTopLeft,        // Y-coordinate of the top left corner of bounding box
        xBottomLeft,     // X-coordinate of the bottom left corner of bounding box
        yBottomLeft,     // Y-coordinate of the bottom left corner of bounding box
        xBottomRight,    // X-coordinate of the bottom right corner of bounding box
        yBottomRight;    // Y-coordinate of the bottom right corner of bounding box

    double circleArea,   // Area of the circle
           circlePerimeter, // Perimeter of the circle
           bbArea,       // Bounding Box area
           bbPerimeter;  // Bounding Box perimeter

    // Write an identification header to the output file:
    logFile << "Programmer: "
             << "Mir Farooq Ali"           // Change this to YOUR name.
             << endl;
    logFile << "CS 1044 Summer II 2004 Project 2"
             << endl
             << endl;

    // Read the coordinates of the circle's center:
    dataFile.ignore(INT_MAX, '(');
    dataFile >> xCenter;
    dataFile.ignore(INT_MAX, ',');
    dataFile >> yCenter;
```

```
dataFile.ignore(INT_MAX, '\n');

// Read the radius of the circle
dataFile.ignore(INT_MAX, ':');
dataFile >> radius;

// Check for an invalid circle
if (radius < 0) {
    logFile << "The given radius for the circle is invalid." << endl;
    return 0;
}

// You must add the following calculations:

// Calculate the coordinates of the bounding box

// Calculate the circle's area

// Calculate the circle's perimeter

// Calculate the bounding box's area

// Calculate the bounding box's perimeter

// Write out the analysis:
logFile << "Bounding Box Coordinates" << endl;
logFile << "-----" << endl;
logFile << "Top Right:      " << '(' << setw(3) << xTopRight
        << ',' << setw(3) << yTopRight
        << ')' << endl;
logFile << "Top Left:       " << '(' << setw(3) << xTopLeft
        << ',' << setw(3) << yTopLeft
        << ')' << endl;
logFile << "Bottom Left:    " << '(' << setw(3) << xBottomLeft
        << ',' << setw(3) << yBottomLeft
        << ')' << endl;
logFile << "Bottom Right:   " << '(' << setw(3) << xBottomRight
        << ',' << setw(3) << yBottomRight
        << ')' << endl;

logFile << endl;
logFile << "Areas and Perimeters" << endl;
logFile << "-----" << endl;
logFile << setprecision(2) << showpoint << fixed;
logFile << "Circle Area:      "
        << setw(8) << right << circleArea << endl;
logFile << "Circle Perimeter:     "
        << setw(8) << right << circlePerimeter << endl;
logFile << "Bounding box Area:    "
        << setw(8) << right << bbArea << endl;
logFile << "Bounding box Perimeter: "
        << setw(8) << right << bbPerimeter << endl;

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

return 0;
}
```