Rochester Institute of Technology
Golisano College of Computing and Information Sciences
Department of Information Sciences Technologies (IST)

Name: _____

ISTE100 – Computational Problem Solving in the Network Domain I
Lab 3: Exploring Functions and Selection

Preliminary Setup

1) From myCourses, download **Lab03.zip** archive.
2) From **Lab03.zip** archive extract three project folders, **Lab3Ex01**, **Lab3Ex02**, and **Lab3Ex03**.

Exercise 1 – Basic use of functions and first exposure to selection. (3 points)

Must be completed during the lab period.

1) Open the **Lab3Ex01** folder.
2) Double click (or single-click if you are running Linux or using a Mac) on the  file.
3) The Code::Blocks project is now open with the program in **`main.cpp`** file.

4) What you see in front of you is a program that calculates the hypotenuse of a right triangle, given the two legs. The program proceeds as follows:
   a) The program prompts the user for the values of the legs.
   b) The program validates user's input, i.e. it will not continue if the user provides a bad value.
   c) The program calculates the hypotenuse using Pythagorean Theorem: $H = \sqrt{L_1^2 + L_2^2}$.
   d) The program displays the value of the hypotenuse.

   However, this program is not complete, and it is your task to complete it.

5) Observe the **code line 15** in the **`main.cpp`** file.
   In the space below, briefly state, what you think this line of code does.

   _____

   Without running the code, examine **code lines 30 – 37.** This code construct is new to you. Explain what it does. Read it and take it literally, as you see it. What constitutes the "bad value"?

   _____

   _____

   _____

6) Is a similar code construct as you saw in **code lines 30 – 37** is needed anywhere else in this program?_____ Where? _____

   If it is needed, copy and paste the **code lines 30 – 37** into the appropriate place within the file. Note the variables within this construct and how they are used. **Change the variable, if needed, in the constructor that you pasted.**

7) Observe the following lines of code, where H^2 is calculated. The "squaring" is done by multiplying a number by itself.

   Open the web browser and go to the http://www.cplusplus.com/reference/cmath/ page. Click on the **pow** function in the lower left-hand pane of the screen. Study how the function is used.

   Replace the line `dfHypotenuse = dfLeg1 * dfLeg1 + dfLeg2 * dfLeg2;` with its analogue, that uses the **pow** function, instead of multiplying each known leg by itself.

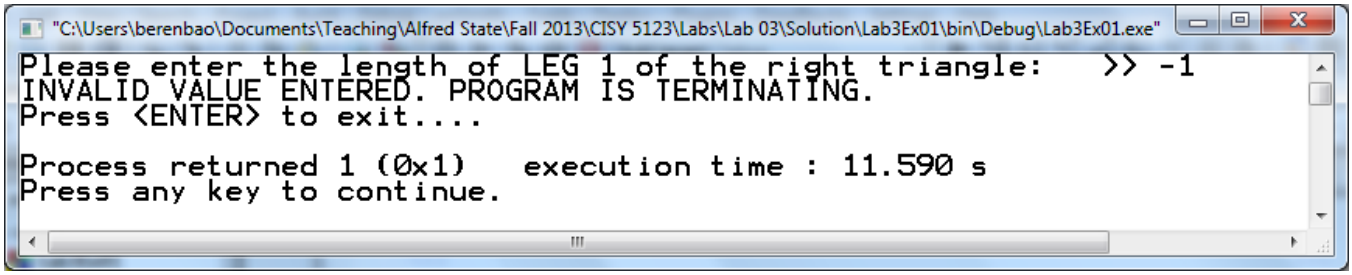
8) Add a line of code that will set the `dfHypotenuse` variable to its own square root. Find an appropriate function to take the square root on the http://www.cplusplus.com/reference/cmath/ page.

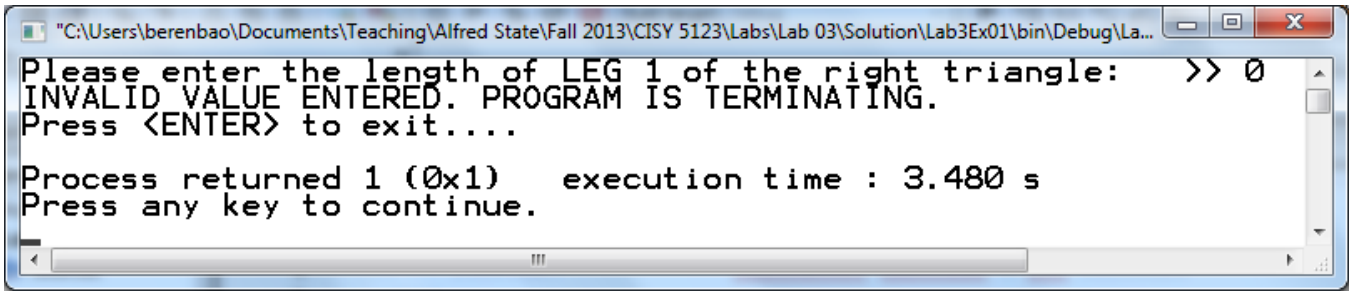
9) Run the program. What does the program do? Try it with bad input.

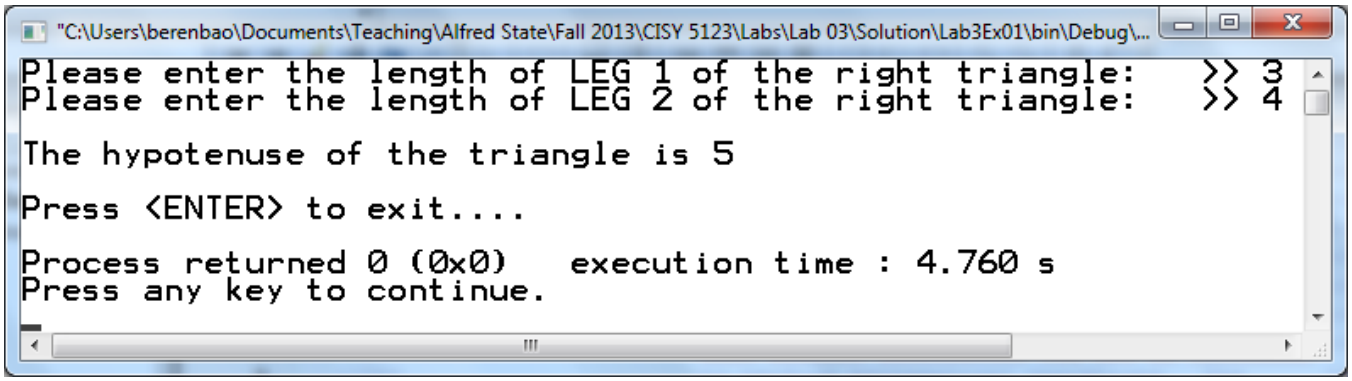
   _____

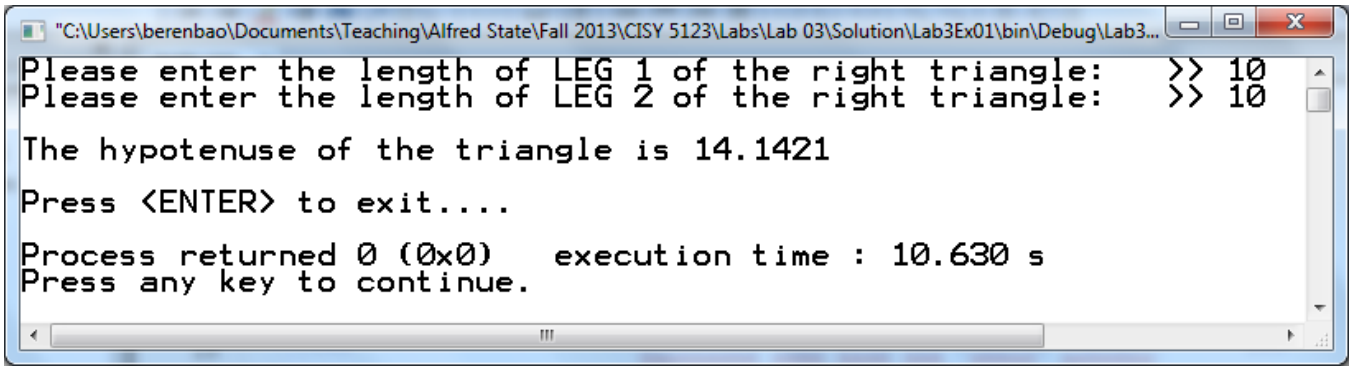
   _____

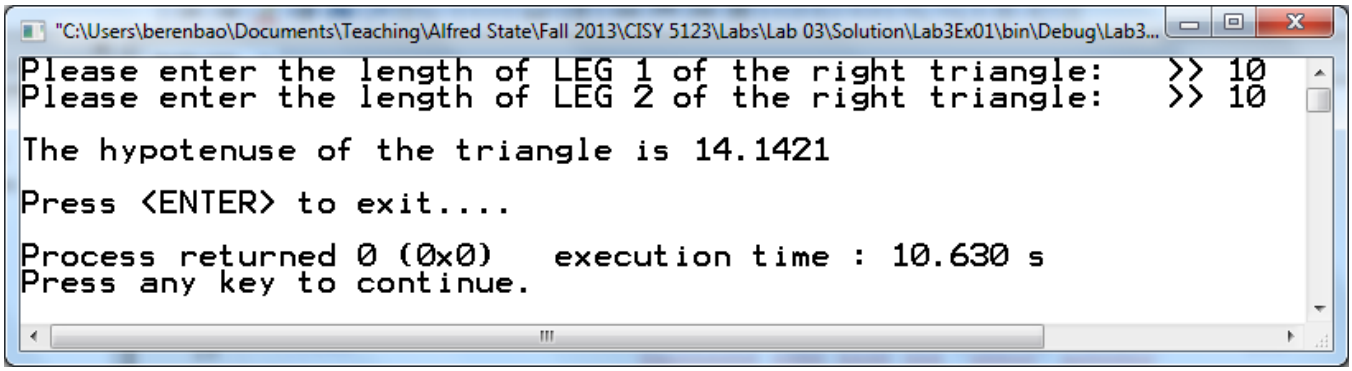
Use the following examples to verify your results:











Signature: _____ Date: _____

Have your instructor sign here when Exercise 1 works correctly and you have answered <u>all</u> questions.

Exercise 2 – Construct a program based on a working example. (3 points)
Must be completed during the lab period

1) Close the **Lab3Ex01** project.

2) Open the **Lab3Ex02** folder and open the project, similarly to the way you have opened the **Lab3Ex01** project.

3) You will now complete the program that calculates leg 2 of the triangle, given the leg 1 and the hypotenuse. The formula is: $L_2 = \sqrt{H^2 - L_1^2}$.

4) Enable this program to use the math library. You can use the solution to Exercise 1 as an example. (Who's to say you can't do some fancy copying and pasting here? ☺)

5) Locate the following comment:

```
//***
//*** Prompt the user for leg 1.
//***
```

Add the code to prompt the user for the value of leg 1. Again, you can work based on the solution to Exercise 1. Be sure to validate user's input based on the criteria set in Exercise 1.

6) Locate the following comment:

```
//***
//*** Prompt the user for Hypotenuse.
//***
```

Add the code to prompt the user for the value of hypotenuse. You can work based the solution to item 5) of this exercise. Be sure to validate user's input based on the criteria set in item 5).

7) Sometimes it is more manageable to break a formula into steps instead of trying to implement the entire thing in one line. Since you have the values for the leg 1 and the hypotenuse, you need to square them. You can place the results of squaring right back into their corresponding variables. For example, to square the value of variable **X** and to put it back into **X** would be done as:
X = X².

Locate the following comment:

```
//***
//*** Get square of hypotenuse and leg 1.
//***
```

Below this comment, add the code to square the values of the leg 1 and the hypotenuse and to leave the values in their corresponding variables.

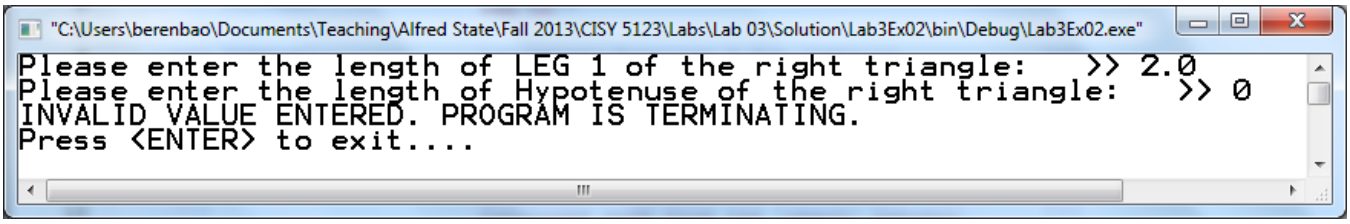
8) Locate the comment:

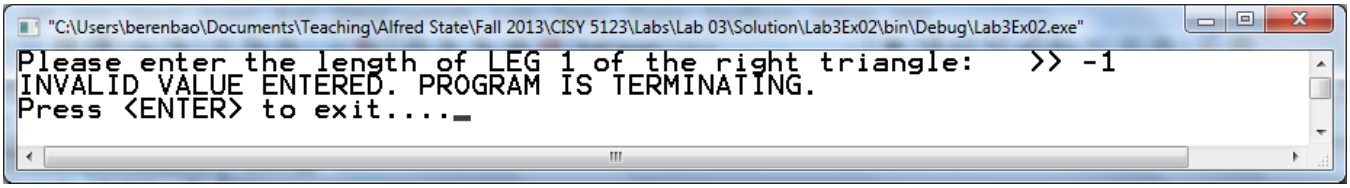
```
//***
//*** Calculate square of leg 2, then calculate leg 2
//***
```

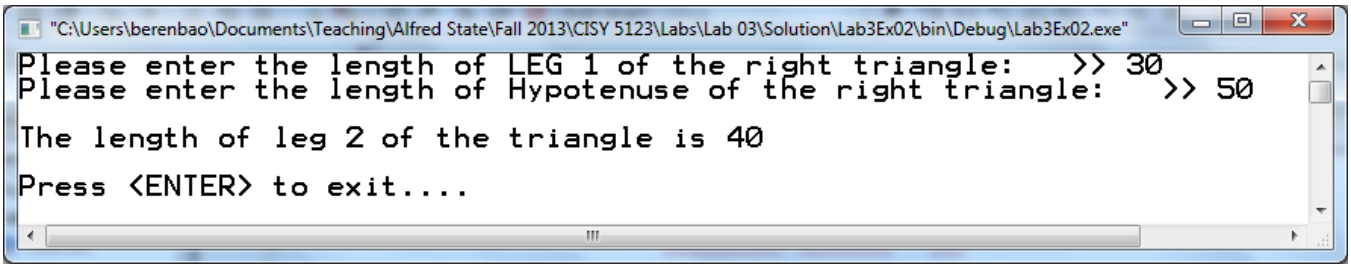
Add the code to calculate Leg 2 of the triangle based on the formula given in item 3) of this exercise. Note the use of the function for finding the square root in Exercise 1 of this lab. Find the Leg 2 using the same function.

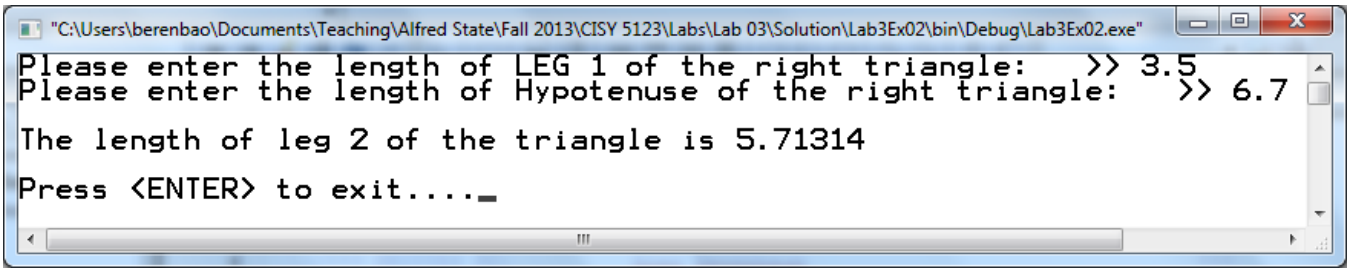
9) Compile and run your program. Try it with "bad input values".

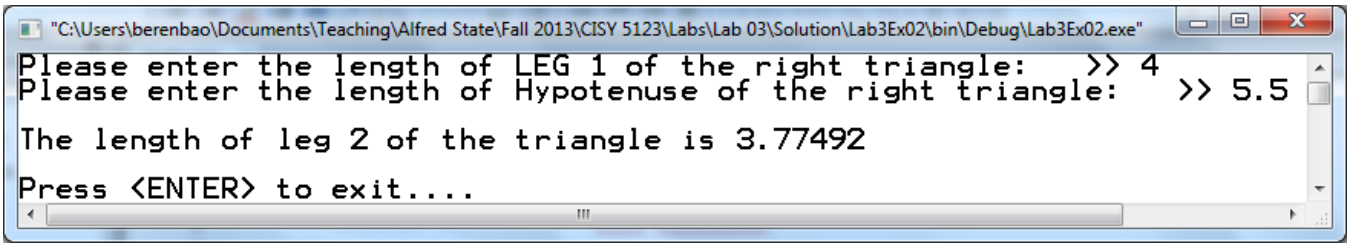
Use the following examples to verify your results:











Signature: _____ Date: _____

Have your instructor sign here when you have completed the table and answered the all questions.

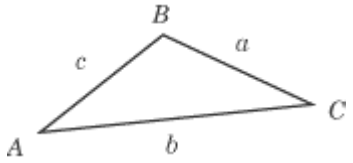
Exercise 3 – Using more functions for math. (4 points)

Must be completed during the lab period

What we have done so far will only work for right triangles. However, we want to have a program that will work with any triangle. To make things easier, the user will supply us with the angle opposite to the "missing" side. This gives us the opportunity to use the "Law of Cosines". If you are not familiar with the Law of Cosines, please read its formulation below:

The Law of Cosines

The Law of Cosines offers a different way of solving non-right triangles and can be used when you don't have the information necessary to use the Law of Sines. This is the Law of Cosines:



$$c^2 = a^2 + b^2 - 2ab\cos(C)$$

If you look carefully at the Law of Cosines, you should see a resemblance to the Pythagorean Theorem. In fact, for right triangles, the Law of Cosines simplifies to the Pythagorean Theorem. Try it yourself. The last term drops out (since cos(90) = 0) and you're left with the familiar formula of $c^2 = a^2 + b^2$. If you're curious, the $2ab\cos(C)$ term compensates for the lack of a right angle.

The Law of Cosines allows you to solve any triangle for which you know any three of the four unknowns in the formula. There are two ways you might know three of the four unknowns:

1. **If you know two sides and their included angle, use the Law of Cosines to find the length of the third side.** Then use the Law of Sines to complete the triangle.

2. If you know the lengths of all three sides, use the Law of Cosines to find the measure of one angle. Then use the Law of Sines to complete the triangle.

We will not use the Law of Sines, because all we are interested is in finding the third side.

1) Close the **Lab3Ex02** project.
2) Open the **Lab3Ex03** project.

3) Enable this program to use the math library. You can use the solutions to Exercises 1 and 2 as examples.

4) Right before the `main()` function, declare a `PI` constant. The value of `3.14159265` for the `PI` constant is sufficient, precision-wise.

5) In the appropriate place in the code (use comments as a guide), add code to prompt the user for the values of **side a** and **side b** of the triangle, with validation. Use exercises 1 and 2 as a guide.

6) Similarly, locate the appropriate place in the code to prompt the user for the **known angle C** and add the code to prompt the user. In this case, we are validating the size of the angle, which cannot be less than or equal to 0 degrees and cannot be greater than or equal to 180 degrees. The "or" operator in C++ is ||. Thus, the expression, which will tell us that the angle value is invalid, i.e. "bad" is: `((dfAngleC <= 0) || (dfAngleC >= 180))`. Be sure to put it into the appropriate place of the **if statement**.

7) Using the math library, in the appropriate place in the source file, calculate the square of the **side C**. You may perform the entire calculation in one line or break it up into several steps.

   NOTE: The `cos` function is expecting its input in <u>radians</u>. The user provided input in <u>degrees</u>.

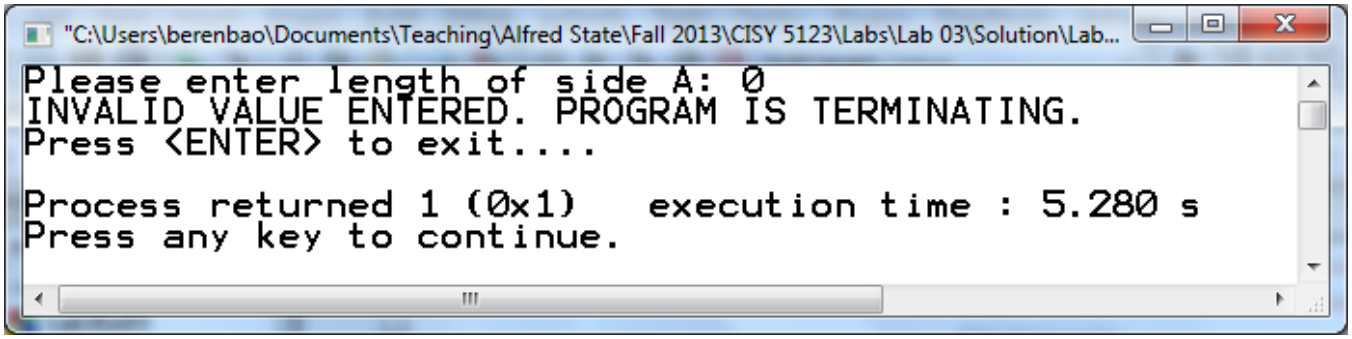
   You need to convert degrees to radians. The formula for conversion is: $rads = degs\,\frac{\pi}{180}$ **.**

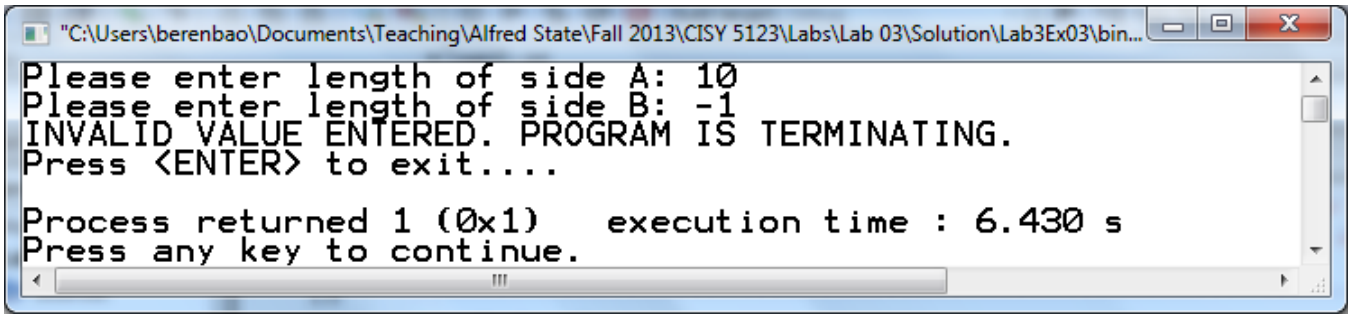
   Thus, you may call your cosine function as: `cos(dfAngleC * PI / 180.0);`

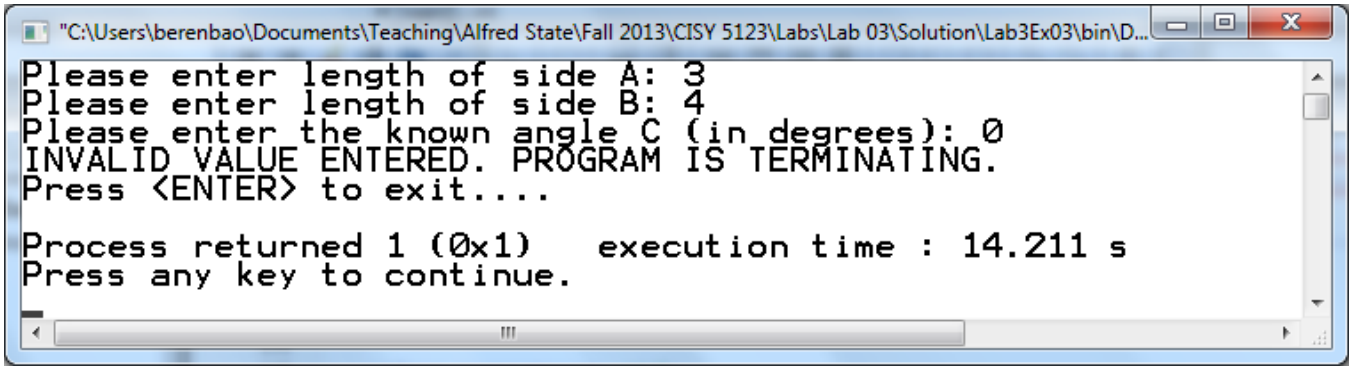
8) In the appropriate place in the source file, take the square root of the result you have obtained in item 7).

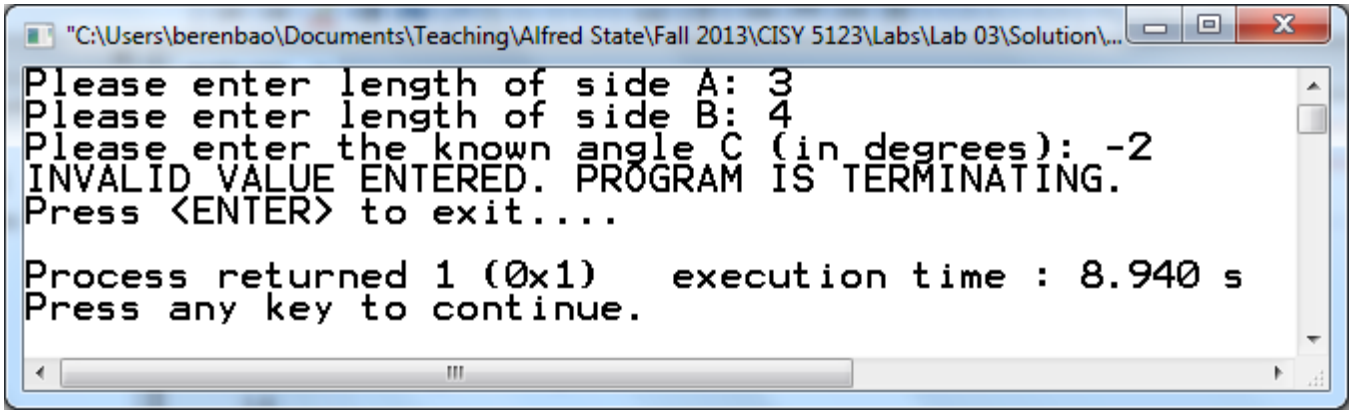
9) Run your program to verify that it produces the results as shown.

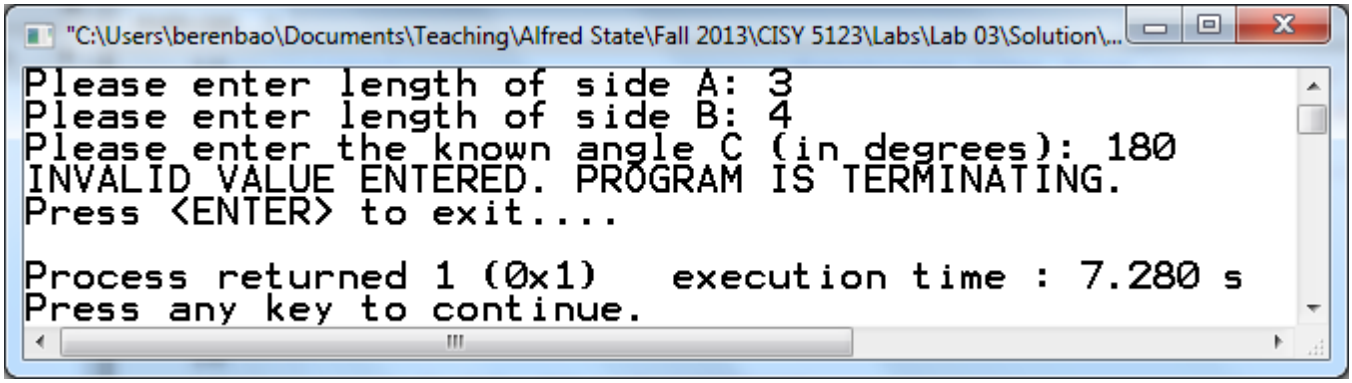
Use the following examples to verify your program.

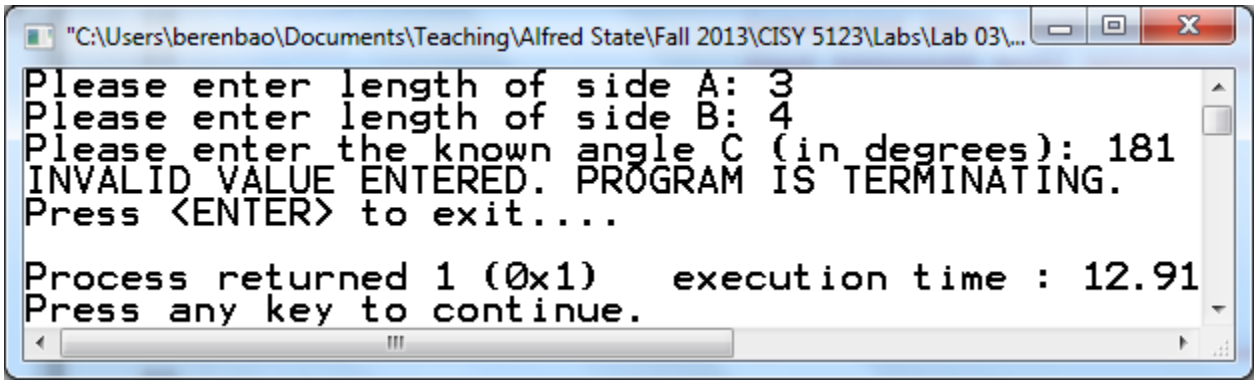


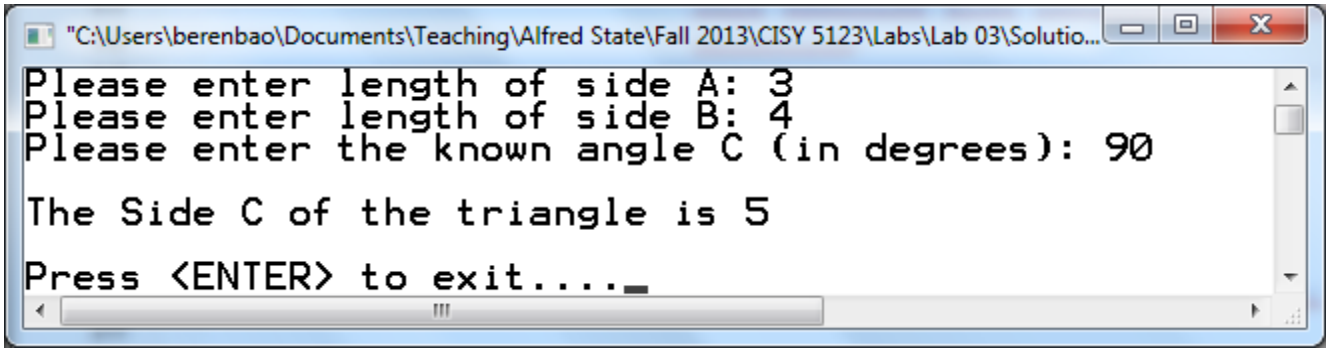


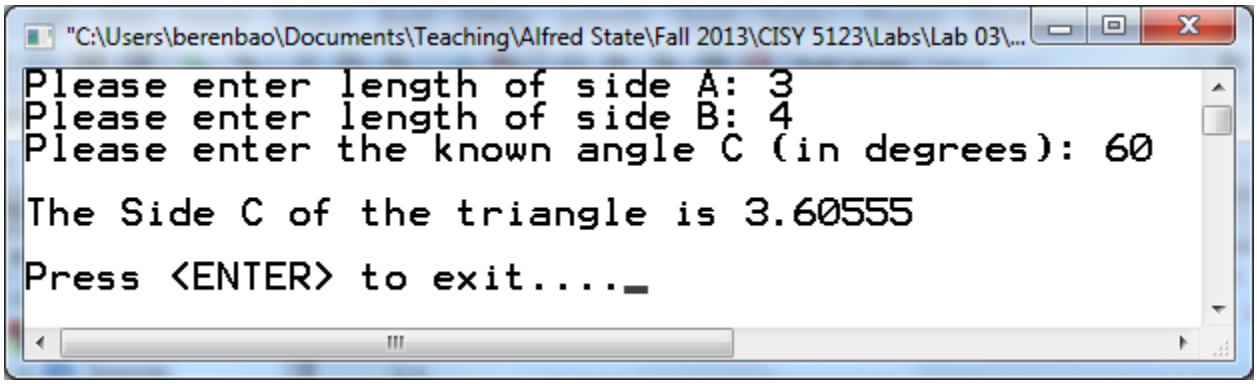

"C:\Users\berenbao\Documents\Teaching\Alfred State\Fall 2013\CISY 5123\Labs\Lab 03\Solution\Lab3Ex03\bin\D...
Please enter length of side A: 3
Please enter length of side B: 4
Please enter the known angle C (in degrees): 0
INVALID VALUE ENTERED. PROGRAM IS TERMINATING.
Press <ENTER> to exit....
Process returned 1 (0x1) execution time : 14.211 s
Press any key to continue.

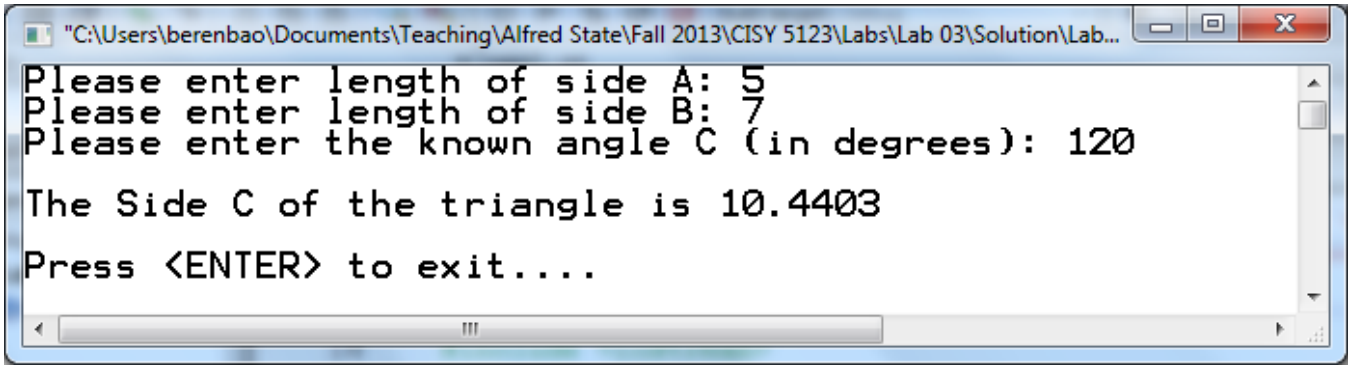

"C:\Users\berenbao\Documents\Teaching\Alfred State\Fall 2013\CISY 5123\Labs\Lab 03\Solution\...
Please enter length of side A: 3
Please enter length of side B: 4
Please enter the known angle C (in degrees): -2
INVALID VALUE ENTERED. PROGRAM IS TERMINATING.
Press <ENTER> to exit....
Process returned 1 (0x1) execution time : 8.940 s
Press any key to continue.

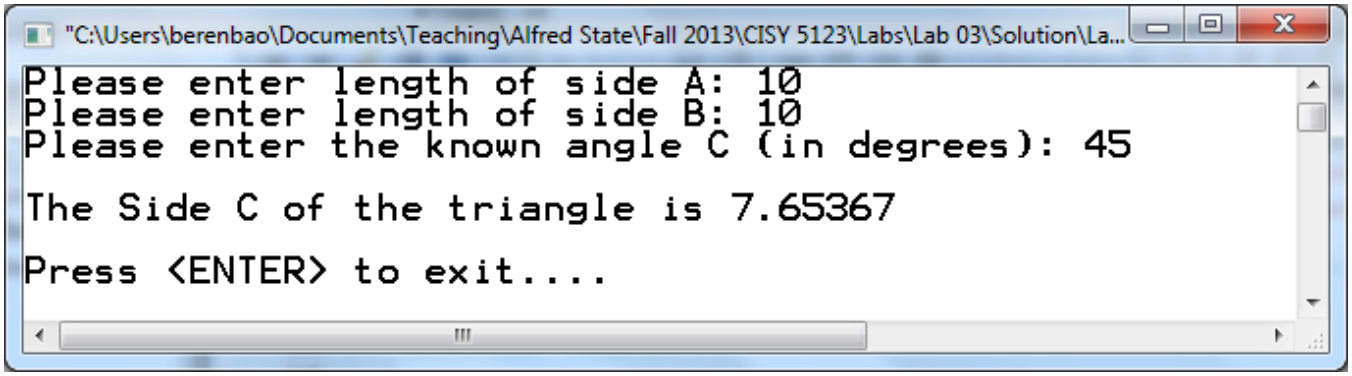

"C:\Users\berenbao\Documents\Teaching\Alfred State\Fall 2013\CISY 5123\Labs\Lab 03\Solution\...
Please enter length of side A: 3
Please enter length of side B: 4
Please enter the known angle C (in degrees): 180
INVALID VALUE ENTERED. PROGRAM IS TERMINATING.
Press <ENTER> to exit....
Process returned 1 (0x1) execution time : 7.280 s
Press any key to continue.


"C:\Users\berenbao\Documents\Teaching\Alfred State\Fall 2013\CISY 5123\Labs\Lab 03\...
Please enter length of side A: 3
Please enter length of side B: 4
Please enter the known angle C (in degrees): 181
INVALID VALUE ENTERED. PROGRAM IS TERMINATING.
Press <ENTER> to exit....
Process returned 1 (0x1) execution time : 12.91
Press any key to continue.









Signature: _____ Date: _____

Have your instructor or TA sign here when Exercise 3 works correctly.