**Lab Experience 3**

**Lab Objectives**

1. Interactive I/O, cin and cout
2. Formatting output.

**Background Information**

All input/output in C++ is considered a stream of data/information that is routed to/from the output/input device that is specified by an object. The istream object cin specifies the input stream is from the keyboard (also called standard input) and the ostream object cout specifies the output is being sent to the monitor/console (also called standard output).

Output is generated by inserting items (converted into their ASCII format) into the output stream specified by the ostream object. In this lab, cout is the ostream object which means all output will be directed to the monitor. The operator that inserts items onto the output stream is the stream insertion operator which is <<.

Input is accomplished by extracting items (converted from ASCII format into the binary representation specified by the data type of the variable) from the input stream specified by the istream object. In this lab, cin is the istream object which means all input is from the keyboard. The operator that extracts items from the input stream is called the stream extraction operator which is >>.

Each stream is associated with an input/output buffer (a buffer is just a segment of memory) and the buffer is associated with the appropriate peripheral device. Some common peripheral devices are the computer monitor, keyboard, and internal/external drives. Peripheral devices are much slower than the processor, thus the usage of the buffer to extract/insert information. Therefore the CPU does not have to wait until the operations are completed to continue executing.

**Interactive Input**

The cin object is used to transfer information from the standard keyboard into the declared identifiers (variables).

The syntax for interactive input from the standard input device is:

cin >> var1 [>> var2 [>> var3….]]; // where anything in [ ] is considered optional.

Everything the user types at the standard keyboard is type char which is in ASCII format. The stream extraction operator, >>, converts the user’s input into the data type specified by the variables data type.

Consider the following program segment:

int main() {

int purchasePrice;

cout << “Enter the price of the item ----> “;

cin >> purchasePrice;

// more statements

return 0;

}// end main

The statement: cin >> purchasePrice;

will extract characters from the standard input data stream a character is encountered that does not match the datatype of purchasePrice. The stream extraction operator will convert the ASCII values into a binary integer representation and store that into the variable purchasePrice.

All characters that are digits will be extracted and converted into their integer binary representation. When a character is encountered that is not digit, the extraction process will stop. The remaining characters will still be in the input buffer. This could be a letter, decimal point, special character, or whitespace.

A cout statement is used to provide directions to the user about what type of data is expected. There should be a cout statement preceding each cin statement to provide directions to the user.

**Output**

The cout object is used to send information to the computer monitor. The stream insertion operator converts the binary representation of the items into ASCII format. The syntax for the cout statement is:

cout << expressionormanipulator [<< expressionormanipulator [<< expressionormanipulator ….]];

where anything in [ ] is optional.

Consider the following program segment:

int main() {

double purchasePrice;

cout << “Enter the price of the item ----> “;

cin >> purchasePrice;

// more statements

cout << fixed << showpoint << setprecision(2) << endl;

cout << “The price of the item purchased is $” << purchasePrice << endl

<< endl;

return 0;

}// end main

The above statements will display the contents of the variable purchasePrice using fixed decimal format, a decimal point (guarantees the decimal point will be displayed if the value after the decimal point is zero), with two decimal places of accuracy.

**Formatting Numeric Output:**

To format numeric data in C++ special manipulators are used to align decimal points, how many decimal places to display, to override the scientific notation, etc. To truly understand how these work it is really necessary to practice, practice, and practice.

The table below is a partial listing of the common manipulators and a description of each:

|  |  |  |
| --- | --- | --- |
| **Manipulator** | **Description** | **Header File Required** |
| fixed | Forces floating point values to be displayed in decimal format with six digits after the decimal point instead of scientific notation or exponential format. | iostream |
| right | |  | | --- | | Right justifies the output | | iostream |
| left | Left justifies the output | iostream |
| showpoint | Forces the decimal point to display if the floating point number is actually a whole number. | iostream |
| setprecision(n) | When used in conjunction with fixed, a flag is set to display the output of all floating point numbers rounded to n decimal places.  **Displays only significant digits when used without fixed.** | iomanip |
| setw(n) | Specifies how many spaces (columns), also called field width, to use to display the contents of the variable. If the number of characters in the number is greater than the width specified, the field width is ignored.  Only affects the next item after the insertion operator.  n can be an unsigned integer or an integer literal or an arithmetic expression that results in an unsigned integer.  Useful to align decimal points and numeric data, strings, etc.  All information displayed is right justified in the field. | iomanip |

**Lab Exercises:**

1. Answer the questions below based on the code segment given below:

int x;

double y;

char ch;

cout << "Enter a value " ;

cin >> x;

cout << "Enter a value ";

cin >> ch;

cout << "Enter a value ";

cin >> y;

a. What is stored in x, y, and ch if the user inputs the following on a single line for the first cin statement?

25.34 a 3.5

x contains: 25 y contains: 34 ch contains: ‘.’

b. What is stored in x, y, and ch if the user inputs the following on a single line for the first cin statement?

25 a 3.5?

x contains: 25 y contains: 3.5 ch contains: ‘a’

c. What is stored in x, y, and ch if the user inputs the following on a single line for the first cin statement?

25a34?

x contains: 25 y contains: 34 ch contains: ‘a’

1. Write the single cout statement with stream manipulators to do each of the following:
   1. Display the number 34.789 in a field of six spaces with two decimal places of precision.

cout << fixed << setprecision(2) << setw(6)<< 34.789;

* 1. Display the number 57.0 in a field of five spaces with three decimal places of precision.

The decimal point and any trailing zeroes should be displayed.

cout << fixed << setprecision(3) << setw(5) << 57.0;

* 1. Display the number 5.789e + 12 in fixed point notation with 2 decimal places.

cout fixed << setprecision(2) << 5789000000000;

* 1. Display the number 67 left justified in a field of seven spaces.

cout << left << setw(7) << 67;

1. Given the following program segment

int main(){

int dollars, cents;

double number ;

cout << “Enter a monetary value “;

cin >> number;

// add the statements here

**dollars = static\_cast<int>(number); // Should drop the decimal places automatically.**

**cents = (number – dollars) \* 100;**

cout << “The number of dollars entered was “ << dollars << endl

<< “The number of cents entered was “ << cents << endl << endl;

return 0;

}// end main

Add the C++ statements that will store the dollar amount input by the user into the variable dollars and the number of cents entered by the user in the variable cents. As an example, if the user enters 15.25, then dollars would contain 15 and cents would contain 25.

1. Download the program **rightTrig.cpp** from D2L and complete the program as directed by the comments. To test your program use 9.56 and 3.25 for the sides. Your program should display the sides of the triangle and the hypotenuse with only 2 decimal digits with all decimal points aligned. **Copy and paste your program into your word document. Below your program, capture the output window, console window and paste it below your program.**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Programmer: Johnathan Lee

//

// Description: This program will ask the user for the two sides of a right

// triangle and determine the length of the hypotenuse.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

int main(){

double side1, // one of the legs of the triangle

side2, // one of the legs of the triangle

hypotenuse; // the hypotenuse of the triangle

cout << setprecision(2) << fixed << showpoint; // format the output

cout << "Enter the length of one side of the triangle: " ;

//Write the input statement for the user to input a value for side1

cin >> side1;

//Write the prompt for the user for the second side

cout << "Enter the length of the next side of the triangle: ";

//Write the input statement for the user to input a value for side2

cin >> side2;

//Write the assignment statement to calculate the hypotenuse

//(cmath has a built-in function called sqrt. The syntax is sqrt(variable\_or\_number)

hypotenuse = sqrt(side1\*side1 + side2\*side2);

//Write an output statement(s) to display the legs of the triangle and the hypotenuse of the triangle

//in column format (appropriately labeled) with all decimal points lining up.

cout // Wasn't sure how wide you wanted it exactly, so I chose a number that

// would fit most inputs.

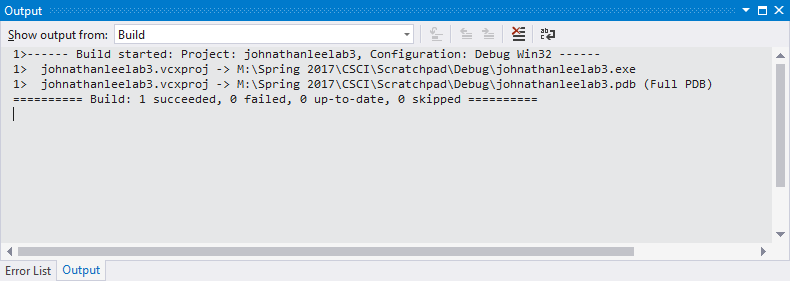
<< "Leg one : " << setw(7) << side1 << endl

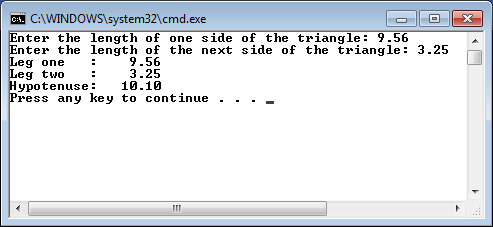
<< "Leg two : " << setw(7) << side2 << endl

<< "Hypotenuse: " << setw(7) << hypotenuse << endl;

return 0;

}// end main





**Due Dates:** According to the due date posted for the assignment folder.

**What to hand in:**

1. Place the word document and the program **rightTrig.cpp** into the Lab Experience Three assignment folder . Save the word document as: yourNameLab3.doc or docx.
2. Hand in a print out of your word document.
3. Hand in a print out of your rightTrig program.