**Lab Experience 3**

**Lab Objectives**

1. Understanding the differences between using **get** and the extraction operator, >> for getting information from any input stream.
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 the object. cin is the object that specifies the input stream is coming from the keyboard and cout is the output stream that specifies the output is being sent to the monitor/console. Items are either inserted onto the output stream or extracted from the input stream. Insertion into the output stream is accomplished by the insertion operator (<<) and extraction from the input stream is accomplished by the extraction operator (<<).

Each stream is associated with an input/output buffer (a segment of memory) and the buffer is associated with an appropriate peripheral device. Some common peripheral devices are the computer monitor, keyboard, and internal/external drives.

There are basically two input streams where the programmer can extract data. The first is using the cin object which extracts information from the keyboard buffer and is used to query the user for information to execute the program.

The second is using sequential files by using file stream objects which are programmer-defined. What is inherent is using sequential files is the program does not ask the user for information, except which file to use, but gets all of the information from a peripheral device like a flash drive, cd, hard drive, etc.

An object represents data and has functions built in which allow the programmer to manipulate the information. These functions are called methods. The most common methods used for input streams are: get, and ignore.

**Input Instructions**

The cin object is used to transfer information from the keyboard into the declared identifiers (variables). Everything typed at the keyboard is in ASCII format, hence it is a character. The extraction operator, >>, is used to convert the input data into the data type specified by the identifier.

For example:

cin>>purchasePrice;

This statement will extract characters from the data stream until it encounters a character that does not match the data type of purchasePrice.

Suppose purchasePrice is of type int, all whole numbers will be extracted until a character that is not a whole number is encountered. This could be a letter, decimal point, special character, or whitespace. All of the characters extracted that are integer will be converted from their ASCII representation into a binary integer representation by the extraction operator.

A coutstatement is used to precede all cin statements since this is interactive input and the user needs directions on what type of data is expected in the program.

**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 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. Only is in effect for the next value specified in the cout statement. If the number of characters contained in the variable is greater than n, then setw(n) is ignored. All information displayed is right justified in the field. | iomanip |

**The get method**

The extraction operator skips all leading whitespace characters. It is sometimes necessary not to skip any characters in the input stream. When this is necessary the get method is used to extract a single character at time since it won’t skip whitespace characters. When using the get method you still must press enter after the character has been entered.

The syntax for the get method is:

get(ch) // where ch is of type character

Example of usage:

#include <iostream>

using namespace std;

int main()

{

charch;

cout<<"This program has paused. Press Enter to continue.";

cin.get(ch);

cout<<"Thank you!"<<endl;;

return 0;

}

As part of the lab exercises you will find out the hidden dangers of mixing the extraction operator and the get method within the same program.

**The ignore method**

The ignore method is used to ignore or discard characters in the input stream. That is, to flush the buffer of extraneous characters that would affect the programs execution. The syntax for the ignore method is:

To skip the next character use: ignore()

To skip n number of characters or until the character c is encountered use: ignore (n, c)

Examples when used with the cin object:

cin.ignore(); // will skip the next character

cin.ignore(20, ‘B’); // will either skip 20 characters or will skip all characters

// up to and including the first occurrence

// of the letter ‘B’

**Lab Exercises:**

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

int x;

double y;

charch;

cout<<"Enter a value " ;

cin>> x;

cout<<"Enter a value ";

cin>>ch;

cout<<"Enter a value ";

cin>> y;

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

25.34 a 3.5

**25 >>x .>>ch 34>>y**

b. What would be stored in x, y, and ch if the input the following on a single line for the first cin statement?

25 a 3.5?

**25 >>x a >>ch 3.5 >>y**

c. What would be stored in x, y, and ch if the input the following on a single line for the first cin statement?

25a34?

**25 >>x a >>ch 3.5 >>y**

1. Write cout statements with stream manipulators that perform the following:
   1. Display the number 34.789 in a field of six spaces with two decimal places of precision.

**cout<<setw(6) <<setprecision(4) << 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<<setw(5) <<setprecision(5) << 57.0**

* 1. Display the number 5.789e + 12 in fixed point notation.

**cout<< fixed <<5.789e + 12**

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

**cout<< left << set(7) << 67**

Given the following data definitions

int dollars, cents;

double number ;

cout<< “Enter a monetary value “;

cin>> number;

Write 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.

**dollars = number / 1** //integer division so that 15.25 drops off everything but the whole number.

**cents = 100 \* (number - dollars)**

1. Download the following program **inputError.cpp** from D2L by copying and pasting it into a new file associated with a project.
2. Execute the program.
3. Did the output match what you expected? What type of output did you get? What does the ASCII character displayed represent? **No, the output did not match what I expected. The output was ascii value 10. Ascii value 10 represents the new line character.**
4. Insert a single statement into the program to correct the input error. Copy and paste your program with the correction into your word document. Below your program, capture the output window and paste it below your program.

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

// Programmer: Tim Wrenn

//

// Description: Potential problems caused by mixing the

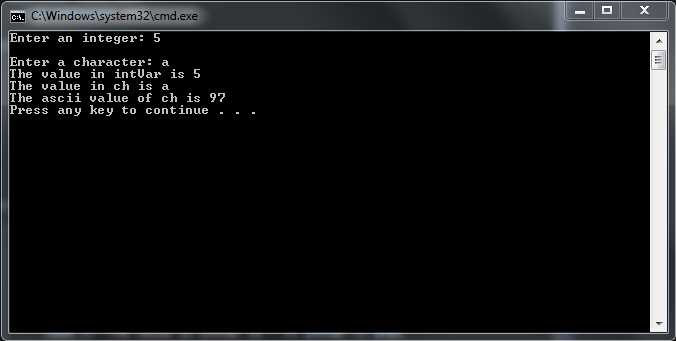
// extraction operator and the get method together

// in one program. NOTE: You are not allowed to change

// any of the original statements.

//

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

#include <iostream>

using namespace std;

int main(){

int intVar;

char ch;

cout <<"Enter an integer: " ;

cin >> intVar;

**//my SINGLE added statement**

**//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

**cin.ignore(80, '\n');**

**//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

cout <<"\nEnter a character: " ;

cin.get(ch);

cout <<"The value in intVar is "<< intVar << endl

<<"The value in ch is "<< ch << endl

<<"The ascii value of ch is "<< static\_cast<int>(ch)

<< endl;

return 0;

}// end main

**NOTE: THE ONLY CHANGE ALLOWED IN THE PROGRAM IS AN INSERTION OF A SINGLE STATEMENT, CHANGING THE ORDER OF EXECUTION OF THE STATEMENTS OR CHANGING A STATEMENT IS NOT ACCEPTABLE.**

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 and paste it below your program.

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

// Programmer: Chris Dang

//

// 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 that brings in side1

cin >> side1;

//Prompt the user for the second side

cout << "Enter the length of the second side. " << endl;

//Write the input statement that brings in side2

cin >> side2;

//Write the assignment statement to calculate the hypotenuse

hypotenuse = sqrt (pow(side1, 2.0) + pow(side2 , 2.0));

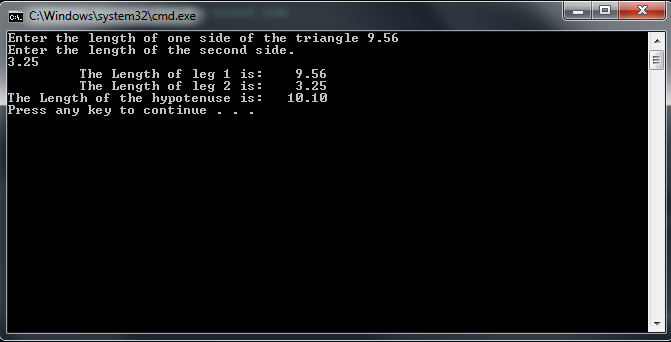
//Write an output statement to display the legs of the triangle and the hypotenuse of the triangle.

cout << " The Length of leg 1 is: " << setw(7) << side1 << endl;

cout << " The Length of leg 2 is: " << setw(7) << side2 << endl;

cout << "The Length of the hypotenuse is: " << setw(7) << hypotenuse << endl;

//All output should be in column format (appropriately labeled) with all decimal points lining up.

 return 0;

}// end main

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

**What to hand in:**

1. Place the word document and the program **rightTrig.cpp** into the Lab Experience Three drop box. 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.