2017 Fall CIS 150 – Lab 2

Release date: October 2nd, 2017

Due date: October 9th, 2017

Explore the use of output “manipulation” for formatting.

Include iomanip library - #include <iomanip>

**setw([int])** – used in cout, will set next output to be in filled size of int value

cout << setw(7) << 5;

output is bbbbbb5 where b represents a space

**left** and **right** – use left and right to direct where in the setw field to justify the value to print.

cout << left << setw(7) << 5;

NOTE: left and right are STICKY which means once set they apply to every field until changed unlike setw which only applies to the next field

**setprecision([int])** - sets the total number of digits to be displayed, when floating point numbers are printed.

cout << setprecision(5) << 1234.537;

output is 1234.5

On the default floating-point notation, the precision field specifies the maximum number of meaningful digits to display in total counting both those before and those after the decimal point. Notice that it is not a minimum and therefore it does not pad the displayed number with trailing zeros if the number can be displayed with less digits than the precision.

In the fixed notation, the precision field specifies exactly how many digits to display after the decimal point, even if this includes trailing decimal zeros. The number of digits before the decimal point does not matter in this case.

How is this done? IT DEPENDS …..

In some versions of C++ you would need to use setf method

**setf([flag value],[field bitmask])**

flag value – scientific or fixed field bitmask - floatfield

cout<<"USING fixed .......................\n" << setf(floatfield,fixed) << setprecision(5) <<1234.537 << endl;

USING fixed .......................

1234.53700

For Visual Studio you just use the reserved word **fixed**

cout<<"USING fixed .......................\n" << fixed << setprecision(5) <<1234.537 << endl;

**setfill([char])** - used in output insertion operations to fill character when results have to be padded to the field width. Note: using left and right will change which spaces are filled by the char.

cout<<"USING setw() & setfill() ...\n";

cout<< setfill('-')<< setw(10) <<11<<"\n";

cout<< setfill('\*')<< setw(10) <<2222<<"\n";

cout<< setfill('@')<< setw(10) <<33333<<"\n";

cout<< setfill('#')<< setw(10) <<4<<"\n";

USING setw() & setfill() ...

--------11 NOTE: not to scale because of Word font

\*\*\*\*\*\*2222

@@@@@33333

#########4

Write a C++ program according to the following algorithm:

Declare a string variable productName

Declare integer variable productQuantity

Declare real number variables productUnitPrice, productTotalPrice

Display welcome message to the user – “Welcome to the Inventory Calculation Program”

Ask the user to enter the product name – “Please enter the product name”

Store name entered into productName

Ask the user to enter the quantity of the product – “Please enter product quantity”

Store integer entered into productQuantity

Ask the user to enter the unit price of the product – “Please enter product unit price”

Store real number entered into productUnitPrice

Calculate productTotalPrice – productQuantity multiplied by productUnitPrice

Display result to the user in the following format:

**Product Quantity Unit Price ($)**

**Pen 100 1.05**

**The total price is: $ 105.00**

Create THREE additional variations of the output using the include<iomanip> operations.

Display thank you message to user – “THANK YOU for using the program written by [your name]”

Provide turn-in document, making sure to include screenshots of program build and execution, as well as file of the C++ code (the .cpp file) to Canvas.