**Lab Experience 4**

**Lab Objectives**

1. Writing programs from the beginning.
2. Understanding the differences between using **get** and the extraction operator, >> for getting information from any input stream.
3. 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()

{

char ch;

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. Given the following code snippet:

cin>> var1; // var1 is type int

cin.get(ch); // ch is type char

What would be stored in the variable ch after the above code has executed?

**The new line character ‘\n’ will be stored into ch after the user hits the enter key.**

1. Given the following code snippet:

cin>> var1; // var1 is type int

getline(cin,str); // str is type string

What would be stored in the variable str after the above code has executed?

**The new line character will be stored into str after the user hits the enter key.**

**Programming Exercises:**

1. The video game machines at a local establishment output coupons depending on how well you play the game. You can redeem 10 coupons for a candy bar or 3 coupons for a gumball. You prefer candy bars to gumballs. Write a program that inputs the number coupons you have won and outputs how many candy bars and gumballs you can get if you spend all of your coupons on candy bars first and any remaining coupons on gumballs. Use the following test data for input into your program: 56 coupons, 75 coupons, 9 coupons.

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

//testing data for problem 2

/\*

int main () {

int var1;

char ch;

string str;

cout << "Type in an integer: " ;

cin >> var1;

getline (cin,str);

cout << "Integer is: " << var1 <<'\n' ;

cout << " String is: " << str <<'\n';

return 0;

}

\*/

//end main

//Problem 3

const int CANDY\_BAR\_TICKET\_PRICE = 10;

const int GUMBALL\_TICKET\_PRICE = 3;

int main() {

int tickets, gumballTickets, gumballs, candybars;

cout << "Enter number of tickets: ";

cin >> tickets;

gumballTickets = tickets % CANDY\_BAR\_TICKET\_PRICE;

gumballs = gumballTickets / GUMBALL\_TICKET\_PRICE;

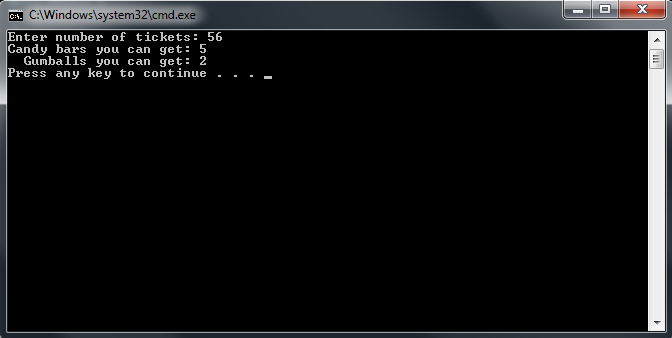
candybars = (tickets - gumballTickets) / CANDY\_BAR\_TICKET\_PRICE;

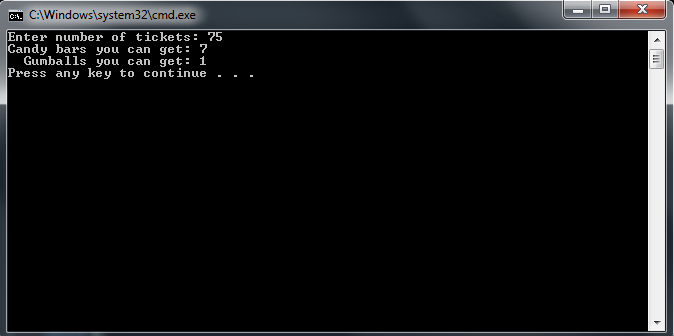
cout << "Candy bars you can get: " << candybars << '\n';

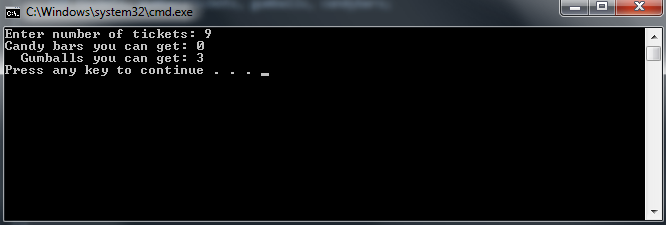
cout << " Gumballs you can get: "<< gumballs << '\n';

return 0;

}







Copy and paste your completed program in this word document. Capture the output window for each of the test inputs given below and paste each output window below your program.

1. Workers at a particular company have won a 7.6% pay increase retroactive for six months. Write a program that takes an employee’s previous annual salary as input and outputs the amount of retroactive pay due the employee, the new annual salary, and the new monthly salary. Use a const declaration for the percent pay increase. Use the following test data for input into your program: $40,000, $65,000, $20,300.

Copy and paste your completed program in this word document. Capture the output window for each of the test inputs given below and paste each output window below your program.

#include <iostream>

#include <iomanip>

using namespace std;

//testing data for problem 2

/\*

int main () {

int var1;

char ch;

string str;

cout << "Type in an integer: " ;

cin >> var1;

getline (cin,str);

cout << "Integer is: " << var1 <<'\n' ;

cout << " String is: " << str <<'\n';

return 0;

}

\*/

//end main

//Problem 4

const double PAY\_INCREASE = 0.076; //Workers get 7.6% increase

int main() {

double oldYearPay, newYearPay, newMonthPay, retroPay;

cout << "Please enter your annual salary: ";

cin >> oldYearPay;

retroPay = oldYearPay \* PAY\_INCREASE;

newYearPay = retroPay + oldYearPay;

newMonthPay = newYearPay / 12;

cout << setprecision(2) << fixed; //sets to 2 decimal points and fixes e notation for answers

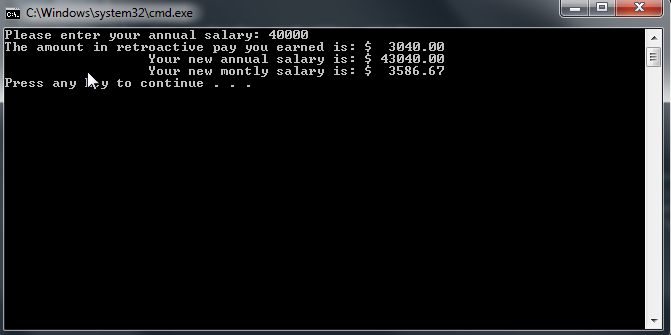
cout << "The amount in retroactive pay you earned is: $" << setw(9) << retroPay << '\n';

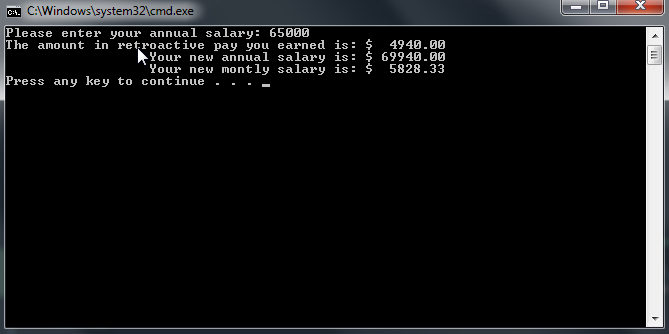
cout << " Your new annual salary is: $" << setw(9) << newYearPay << '\n';

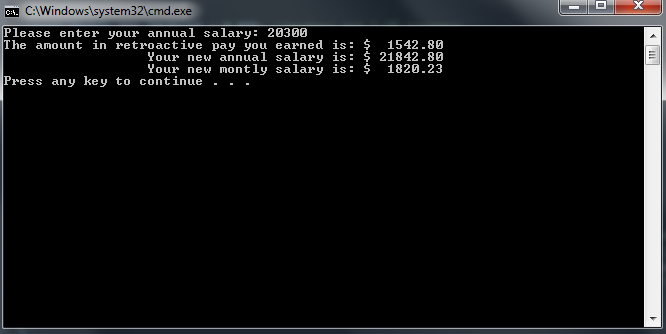
cout << " Your new montly salary is: $" << setw(9) << newMonthPay << '\n';

return 0;

}







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

**What to hand in:**

1. Compress the .cpp files from the programming exercises and the word document into a single file called **{yourname}Lab4.zip** e.g. timwrennlab4.zip Note: **If your name is not part of the zip filename, I will not open the zipped file.*Note: Only compress the .cpp files from the programming exercises.***
2. Hand in a print out of your word document.
3. Hand in a print out of your .cppprograms.