

Objectives:

- Continue understanding of basic output and input (cin/cout)
- Continue understanding of C++ mathematical operators
- Mathematical Library Functions
- Understanding conditional branching - if statements

Procedures:

1. Boot up your Ubuntu Linux OS and log in.

2. Open a terminal. At the command prompt, examine the man page for g++. What command did you use to do this?

`man g++`

3. While in the g++ man page, hit the `/` key. A forward leaning slash should **appear** at the bottom left side of the screen. This enables the search function of the manual system. Type the word `Wall` at the forward leaning slash and hit enter. You will notice that the manual skips to the first instance of the word `Wall` in the man page. Hitting the `n` key will skip to the next instance of the searched for word.

4. Use the manual system to research the `-Wall` command line option. What does the `-Wall` option do, and what sort of flags does it enable?

`-wall enables all the warnings about constructions that some users consider questionable and that are easy to avoid.`

`the flags enabled. are about construction.`

5. Use the manual system to research the `-Wextra` command line option. What is the function of this option?

`The. function of -Wextra is to enable more warning flags that werent enables with -wall`

6. Use the manual system to research the `-O` command line options. What is the function of this option? Briefly explain the levels available for this option.

`-o is used so the compiler can try to reduce. code size and execution time w/o performing any optimizations.`

`With each level, O2, O3, Os, Ofast, etc., each one optimizes more and more.`

7. Examine the program listing below:

```
#include<iostream>
#include<cmath>
using namespace std;
```

```
int main()
{
```

```
    int x = 5, y = 8, z = 10;
```

```
    cout << ( x < y ) << endl;
```

```
    cout << ( y != y ) << endl;
```

```
    cout << ( x == z ) << endl;
```

```
    cout << ( y == y ) << endl;
```

```
    cout << ( x + y > z ) << endl;
```

```
    cout << ( y > x + 5 ) << endl;
```

```
    x = ( z * y - x * z ) / x; Im going based that all of these are
```

```
    cout << ( x != z ) << endl; using the newly assigned X
```

```
    cout << ( x / z > y - 5 ) << endl;
```

```
    cout << ( x + 4 <= z - 5 ) << endl;
```

```
    cout << ( x * y <= pow(x,2.0) ) << endl;
```

```
    cout << ( y * z <= pow(x,2.0) ) << endl;
```

```
    cout << ( x >= sqrt(z) ) << endl;
```

```
    cout << ( x * z % y < 5 ) << endl;
```

```
    if (x > 100)
```

```
        cout << "a" << endl;
```

```
        cout << "b" << endl;
```

```
    if (x < y)
```

```
        cout << "one" << endl;
```

```
    else
```

```
        cout << "two" << endl;
```

```
    if (x + y >= y + z)
```

```
    {
```

```
        cout << "three" << endl;
```

```
    }
```

```
    else
```

```
    {
```

```
        cout << "four" << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

OUTPUT

True (1)

false (0)

False (0)

True (1)

True (1)

False (0)

X = 150

True. (1)

True (1)

False (0)

True (1)

True (1)

True (1)

True (1)

a

b

nothing, it will skip and go to else

two

three

Nothing

Use the spaces to the right of the listing to show the output of the program. Only list output for statements that are actually executed by the program.

8. Create a new directory called prog5. Type in the program using a text editor and save the source file as 'prog5a.cpp' in the new directory. Compile the program with all warnings enabled. What command did you use to accomplish this?

g++. -o -Weverything prog5.cpp

9. Run the program, comparing its output to your estimate in the previous step. Explain any difference between your estimation and the actual output of the program.

I believe it didnt keep the newly assigned x
there were a few that were wrong

10. Copy the Makefile from the previous lab and alter it so that it will compile prog5a.cpp. List the contents of the new Makefile below:

```
all: prog5

prog5: prog5.o

g++ prog5.o -o prog5

prog5.o: prog5.cpp

g++ -c prog5.cpp

clean:

rm -rf *.o prog5
```

11. Examine the Fahrenheit to Celsius program you created in Lab 4. Alter this program so that it can convert both Fahrenheit to Celsius as well as Celsius to Fahrenheit. Prompt the user for which conversion to make, then input the temperature from the keyboard, make the proper conversion (whichever the user selects) and output the converted temperature. Supply a flowchart that shows the logic of your altered program and attach it to this lab sheet. As usual, make sure you write a complete C++ program and add copious comments. Print out the source code and attach it to this lab sheet.

12. Write a complete C++ program that prints all the real solutions to the quadratic equation:

$$ax^2 + bx + c = 0$$

for a given set of coefficients. Declare variables (of an appropriate type and using meaningful variable names) to represent the coefficients a, b, and c, as well as the solutions. Then read the values for a, b, and c from the keyboard and use the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

to generate the solutions. Print out a message when the user supplies values for which the roots are undefined.

Run the program several times, with different inputs. Try to explain any anomalous results.

Make sure you write a complete C++ program. You may use additional variables for the calculations as necessary. Add judicious comments to document your code.

Include a header, like the one from previous labs, displaying your name, the date, the course number, and a short description of the program.

When you have finished, print out the code and attach it to this lab sheet.

Objectives:

- Continue understanding of basic output and input (cin/cout).
- Decision structures and conditional branching.
- Nested if statements / selection structures.

Procedures:

1. Examine the program listing below:

```
#include<iostream>
#include<cmath>
using namespace std;

int main()
{
    int x = 5, y = 8, z = 10;

    cout << ( x < y || y > z ) << endl;
    cout << ( x < y && y > z ) << endl;
    cout << ( y < z ) << endl;
    cout << !( y < z ) << endl;
    cout << ( !( y < z ) || ( x < y ) ) << endl;
    cout << ( !( y > z ) && !( x > y ) ) << endl;
    cout << !x << endl;
    cout << ( x + y > z ) << endl;
    cout << !( x + y > z ) << endl;
    cout << ( ( x + y > z ) || !( z - x > y ) ) << endl;

    if ( x > y || y > x && z > y )
        cout << "a" << endl;
        cout << "b" << endl;

    if ( x + y && y + x > z )
        cout << "one" << endl;
    else
        cout << "two" << endl;

    if ( x + y >= y + z || z - y < x )
    {
        cout << "three" << endl;
    }
    else
    {
        cout << "four" << endl;
    }

    return 0;
}
```

OUTPUT

True

false

True

False

True

True

False

True

False

True

A

B

One

Three

Use the spaces to the right of the listing to show the output of the program. Only list output for statements that are actually executed by the program.

Compile and run the above program, comparing its output to your estimate above. Explain any differences between your estimates and the actual output of the program.

2. Write a complete C++ program that implements a game of *Rock, Paper, Scissors* between a user and the computer. Print out a prompt and have the program give the player the following menu:

1. Rock
2. Paper
3. Scissors

After reading in a number from the keyboard, use a selection structure to set the player's choice to one of the strings "Rock", "Paper", or "Scissors". Use the C++ `rand` method to generate a random number in the range 1-3 for the computer's turn, then use another decision structure to set the computer's choice to one of the three strings. Make sure to seed the random number generator appropriately. Finally, using some combination of decision structures, decide the winner of the game using the following rules:

- Rock crushes scissors (Rock wins)
- Scissors cut paper (Scissors wins)
- Paper covers rock (Paper wins)

Inform the user of the player choices, the rule used, and tell the user who won (the user or the computer).

Make sure you write a complete C++ program. You may use additional variables as necessary. Add judicious comments to document your code.

Include a header, like the one from previous labs, displaying your name, the date, and the course number.

When you have finished, print out the code and hand it in to your instructor.