**CS 60, Lab 1**

**Introduction to Linux and C++**

**You may wish to refer to**[**this program**](https://camino.instructure.com/courses/111345/files/10017953?wrap=1)[**Download this program**](https://camino.instructure.com/courses/111345/files/10017953/download?download_frd=1)**as a C++ syntax reference.**

Note:  You are required to use a text editor (Sublime) and a terminal as your programming environment.  You may NOT use xcode, vscode, or any other interactive development environment.

**ONE member of your pair should start a new document in Word or similar.  During the lab, you will see instructions to add things to the  lab report; add these to the document.**

**Part A:  Linux**

**Ensure that as you do part A you and your partner are communicating and going at the same pace.**

Linux is a way to control what your computer does; it's an alternative to using a mouse and pointing and clicking.  Instead, you type commands to do things like create files.  You will use Linux to compile and run your code.  Don't worry, you can still do most things with the mouse.  :)

You will type your commands into a command prompt.  Depending on whether you're on PC or Mac, you'll use a different program as your command prompt; "Cygwin terminal" (PC) or "terminal" (Mac).  At any moment in time, that program is pointed at a particular folder on your computer, and you can enter commands that affect the contents of that particular folder.  For example, you can give a command to compile a C++ code file that is in that folder.  Choose the instructions for either PC or Mac below.  These instructions will walk you through creating a file for your code, and then show you the commands to enter so that your command prompt is pointed at the folder you just made.

1.  Compile and run the sample code.

[PC instructions](https://camino.instructure.com/courses/111345/files/9951731?wrap=1)[Download PC instructions](https://camino.instructure.com/courses/111345/files/9951731/download?download_frd=1)

[Mac instructions](https://camino.instructure.com/courses/111345/files/9951752?wrap=1)[Download Mac instructions](https://camino.instructure.com/courses/111345/files/9951752/download?download_frd=1)

cd is the most important Linux command you will use this quarter.  Using the folder app,  create a new folder called "test" nested inside the folder you made for this class.  If you type "ls" (short for list) you should see the contents of the current folder listed.  Use cd to navigate inside the test folder you just made.  To get back to the previous folder (the folder one level "up"), you can type "cd .."

2.  Type "cd" to get back to your default folder.  Use the cd command to navigate to the "test" folder you made.  **What sequence of commands did you use?  Put the answer in your lab report.**

When you're done, **raise your hand to get your work checked**

If you are interested in learning more Linux commands, feel free to work through [this supplemental optional tutorial](https://camino.instructure.com/courses/111345/files/9951754?wrap=1)[Download this supplemental optional tutorial](https://camino.instructure.com/courses/111345/files/9951754/download?download_frd=1)after lab.

**Part B:  Variables and Expressions in C++**

Consider the following C++ program (but **do NOT run it**):

#include <iostream>

using namespace std;

int main()

{

cout << 1 + 2 \* 3 << endl;

cout << 1 + (2 \* 3) << endl;

cout << 10 / 4 % 3 << endl;

cout << 10 / (4 % 3) << endl;

cout << ((1 < 2) || (0 > 10) && (0 > 100)) << endl;

int a(1), b(5), c(7);

a = a + 1;

cout << a << endl;

a += b;

cout << a << endl;

cout << a++ << endl;

cout << ++a << endl;

cout << ((c % 2) ? b : a) << endl;

return 0;

}

1. Review C++ precedence rules to determine which operator is performed first when there are multiple operators in an expression.   Note that a unary operator is an operator that has only one input; unary - is used like so: int x = -5;
2. WITHOUT running the code: What is the output of this program?  put your answer in your lab report.  After you have made your prediction, run the code and see if you are right.  See if you can figure out any mistakes you made; ask the TA or instructor if you're not sure.
3. What is the difference between pre-increment and post-increment operators ?  Put your answer in your lab report.

Include your answers in your lab report and **raise your hand to get your work checked**

**C.  Conditionals & Loops**

**You should be working together on ONE program.** I suggest you exchange phone numbers or email addresses with your lab partner.**Before you leave lab, make sure both of you have a copy of the code.**

A prime number is a whole number whose only divisors are 1 and itself. So we know that 7 is prime because none of 2, 3, 4, 5, or 6 divide evenly into 7. Write a program that prompts the user for a number, and then tests whether that number is prime. Your program should print out the result. You must use a for loop.

1. Discuss how to do this.  If you’re feeling stuck, try to think of how you would check if a number is prime using paper and pencil, then try to think of how you would automate that process.  Write one or two sentences in your lab report, or some pseudocode, explaining your approach. **Raise your hand to get your code checked**
2. Write the code you discussed in 1.  Run your code with several outputs to be sure it’s right.  Copy-paste your code into your lab report and **raise your hand to get your code checked**
3. Change your code so that you now have a function isPrime, which takes in an integer and returns true if it's prime, and false otherwise.  Make sure to call your function from main.  Run your code with several outputs to be sure it’s right.  Copy-paste your code into your lab report and **raise your hand to get your code checked**

**D.  Nested Loops**

What is the output of each double-for loop?   **Trace it on paper BEFORE you run it.** At each step, **raise your hand to get your work checked** before you run the code.

* 1. for(int i=0; i<10; i++){  
            for(int j=0; j<5; j++){  
                cout<<"x";  
            }  
            cout<<endl;  
         }
  2. for(int i=0; i<5; i++){  
            for(int j=0; j<10; j++){  
                cout<<"x";  
            }  
            cout<<endl;  
         }
  3. for(int i=0; i<10; i++){  
            for(int j=0; j<10; j++){  
                if(i==j){  
                    cout<<"o";  
                }  
                else{  
                    cout<<"x";  
                }  
            }  
            cout<<endl;  
         }
  4. Write a program that creates the following output. You must use loops.

0 0 0 3 2 1 6 4 2 9 6 3 12 8 4

Hint:  use the line  cout<<i\*j<<" ";

You can go straight to C++ for this problem.  Copy-paste your code into your lab report and **raise your hand to get your code checked**

**E.  Arrays and file i/o**

In part E you will write a program to sort an array of ints.  You may wish to refer to [this program](https://camino.instructure.com/courses/111345/files/10017955?wrap=1)[Download this program](https://camino.instructure.com/courses/111345/files/10017955/download?download_frd=1)as a C++ syntax reference.

1.  Write a main function that creates an array, and reads values into the array from [this file](https://camino.instructure.com/courses/111345/files/9951730?wrap=1)[Download this file](https://camino.instructure.com/courses/111345/files/9951730/download?download_frd=1). Your program should read in the contents of the file, then print out the contents of the array.  [Here](https://camino.instructure.com/courses/111345/files/9951753?wrap=1)[Download Here](https://camino.instructure.com/courses/111345/files/9951753/download?download_frd=1)is an example of a program that reads in values from a file and prints them out.  You should copy-paste this program into Sublime and use it as your starting point.

Note: In C++ you must decide the size of your array when you declare it, and it can never change size .  if you're not sure how big your array should be, a common approach is to create an array that is larger than you think you will need, and keep track of how many items you put into it.  Make your array size 100.  Note that we will see better ways to handle this situation in a few weeks.

Troubleshooting:

* Remember that the input file must be in the same folder as your .cpp file, and its name should end in .txt.
* If you can't connect to the file and you've checked that it's in the right folder, try specifying the full path.  Instructions on how to do that [are here](https://camino.instructure.com/courses/111345/files/9951755?wrap=1)[Download are here](https://camino.instructure.com/courses/111345/files/9951755/download?download_frd=1).
* If you run into strange issues, try downloading the file instead of copy-pasting.

When you're done, **raise your hand to get your code checked**, and add this function to your lab report.

You've reached the Minimum Stopping Point.  This is as far as you need to complete the lab to get full credit.  Content after this point is extra practice you can work on if you have extra time in lab.

Put the contents of your whole C++ file into your lab report.

Submit the lab report on Camino.  P**lease copy-paste the contents of your lab report document into the Camino submission box,  with the last names of both members of your group at the top.**

I suggest you exchange phone numbers or email addresses with your lab partner.  Before you leave lab, make sure both of you have a copy of the code.

--------------------------------Minimum Stopping Point----------------------------------

2. Write a function called max that finds the **index**of the largest value in an array.

To get started:  How many parameters should the function have?  What should its return type be?

Be sure to test the function with a call from main.  Where should the main function be defined in relation to the max function?

When you're done, **raise your hand to get your code checked**, and add this function to your lab report.

3. Write a function that calls max and swaps the largest value in the array to the last spot in the array, for example:

If the array a contains:  5, 3, 8, 1, 9, 3, 2

Then after calling your function on a, it will look like:  5, 3, 8, 1, 2, 3, 9

That is, the 9 and 2 swapped places.

To get started:  How many parameters should the function have?  What should its return type be?

Be sure to test the function with a call from main.

When you're done, **raise your hand to get your code checked**, and add this function to your lab report.

4. How can you modify your function from part 2 so that it ALSO puts the second-to-biggest number in the second-to-last spot? for example:

If the array a contains:  5, 3, 8, 1, 9, 3, 2

Then after calling your function on a, it will look like:  5, 3, 3, 1, 2, 8, 9

Hint:  You can use the function you wrote in part 1 to find the second-largest number in the array (as long as you do it after swapping the largest into the last spot).  How?

Be sure to test the function with a call from main.

When you're done, **raise your hand to get your code checked**, and add this function to your lab report.

5. How can you modify your function again so that it sorts the whole array?  If you're having a hard time getting started, first think about how you would modify your function so that the last three items in the array are the largest three items, in order.

Be sure to test the function with a call from main.

When you're done, **raise your hand to get your code checked, a**nd add this function to your lab report..