

Please complete labs 1 & 2. Like today's exercise, they are trivial and meant to give you an idea of C++ code layout and the use of variables.

Create a new file, and simply count to 1,000,000. Counting to one million is not something we would ever do by hand, but computers do it tirelessly. To accomplish this, you might take these steps:

a. Create the program structure

```
#include <iostream>  
using namespace std;  
int main( )  
{  
// your program here  
}
```

b. Create a counting variable. "counting" means integer. Name it something appropriate, like "loopCounter".

c. After creating and initializing your variable, create a "while" loop. The structure of a while loop is as follows:

```
while ( condition )  
{  
// do something  
}
```

in this case, the condition is (`yourVariable < 1000000`), and the "do something" is two lines: print out the value of your variable,

use **cout** and don't forget the **endl;**

and then INCREMENT your variable by 1. The line of code will be something like

```
yourVariable = yourVariable + 1;
```

d. To get your code up and running, start simply.... perhaps only count to 10, and answer these important questions:

- What should the initial value of your variable be? 0 or 1? Try each, and see what value is first displayed.

- Should the condition be "**while (yourVariable < 10)**" or "**while (yourVariable <= 10)**" ? Try each, and see what the last displayed value is.

e. When you are satisfied that your loop boundaries are correct, replace the condition with 1000000 instead of 10, and watch your program run. How long does it take to reach 1000000?

f. Now try an experiment - move the "**cout << yourVariable << endl;**" statement from inside the loop to after the loop. In this case, the program will only print the final value when the loop is completed. NOW, how long does it take to run? You will note that the difference in execution time is due entirely to the act of printing to the screen.

g. What is the final value of your variable outside of the loop? Is it different than when the cout statement was inside the loop? Why?

the answer is: moving the cout statement to outside the while loop meant that it was printed only after the final increment. The final increment was actually 1,000,001, when the while condition failed, and then was bypassed.

h. As mentioned in the syllabus, not all the labs will be graded and hence you do not need to submit this lab. We will inform you which labs need to be turned in. However, you are encouraged to complete the assignments and attend labs for a better understanding of the subject.