

Lab 2. Our First C Program

GOAL: Practice the procedure to create a source code file using the vim editor, compile and build it, and get it to work. This is the general procedure for creating, editing, compiling C code, which will be used a lot this semester.

Step 1: Login to your Linux system

If using your Mac, open a Terminal window.

if using an ecs system, then:

- Make sure your VPN, GlobalProtect is connected **if** you are not on the campus network.
- Log in using your Sac State credentials to a Coding computer using Putty, Mobaxterm, ...

STEP 2: Getting Set Up

Create a sub-directory under your home directory, if it doesn't already exist, named "**csc60**". Change to that new directory and create a sub-directory named "**lab2**", then change to that new lab2 directory.

(Hint: use **mkdir** and **cd** commands)

STEP 4: Create your source code file

Create the new source file by beginning to edit the file with vim. The program will print your favorite quote.

Add the following lines of text (which are in the box below) to the file:

<code>/*-----*/</code>	<i>Don't count the dashes. Approximate.</i>
<code>/* Your Name Here */</code>	<i>Put Your-Name here, both <u>first & last</u>.</i>
<code>/* Lab 2 */</code>	
 <code>#include <stdio.h></code>	<i>Preprocessing directives for the compiler.</i>
<code>#include <stdlib.h></code>	
 <code>int main (void) {</code>	<i>"main" function definition required for all C programs.</i>
<code>printf("\nLab 2. \n\n");</code>	
<code>printf("Hello World.\n");</code>	<i>Be sure to indent for clarity</i>
<code>printf("Hi, Your Name. \n\n");</code>	<i>Put your First & Last name here!</i>
<code>printf("Quote \n");</code>	<i>Put a quote here instead of the word "Quote" Find a good one!</i>
 <code>return EXIT_SUCCESS;</code>	<i>Capitalize EXIT_SUCCESS</i>
<code>}</code>	

While in the editor, try a few vi commands:

- While in Command mode, move around the file with the following commands, see what they do:
 - <spacebar>, Ctrl-h, w, b, \$, 0 (zero),
 - H, M, L
 - j, k
 - /printf, n, ? (search for the word "printf", then repeat the search, and again but backwards)
 - % (move cursor to a "(" or "{" and then press %)

Step 5: Save your work, compile it and run it

To save your work and quit, (hit escape key to get out of insert mode), then type: **:wq**

The shell prompt returns.

Compile the program with the following command: **cc lab2.c** (you may need to use *gcc* rather than *cc*)

If the compile is successful, the executable output is sent to a file called **a.out**

If you have compile errors, they will appear, and you will need to fix them and recompile.

The prompt returns.

If you have no compile errors, type **a.out** and the output of your program will display.

Note: If typing **a.out** does NOT work, try: **./a.out**

Immediately after your run, enter the command **"echo \$?"** to check the exit status of the program.

```
$ ./a.out
```

```
Lab 2.
```

```
Hello World.
```

```
Hi, Mark Ainsley.
```

```
Be yourself; everyone else is already taken.
```

```
- Oscar Wilde
```

```
$ echo $?
```

```
0
```

```
$
```

NOTICE: the attribution (Who created the quote) is required

Check the exit status of your program

It should be 0, indicating Successful Exit

This number came from the "return()"

Standardize your indentation of your source file:

If you are not sure about your indentation, get out of Insert Mode (hit: **Esc key**), then:

type: **:1** which takes you to line 1 (Note: *"gg"* will also take you to line 1)

type: **=G** which will standardize your indentation

If you have Errors:

If you have errors, it is OK, a normal course of events. Examine the Error Message list. Sometimes the second or third message makes more sense than the first error message. One code error can cause SEVERAL error messages.

Important Reminder

Every time you change the code, you must **redo** the COMPILE (which is the **gcc** line) before you run the program, or you will NOT see any changes when you run the program (a.out)!!!

Step 6: Update the source code to include a conditional statement

Edit your source code file using vim

In the main function, add the definitions of two integer variables named “num1” and “num2” resp. Initialize **num1** to 5 as part of its definition. Assign the value of **num2** to 7 after its definition.

Before the “Hello World” printf() statement, add a **conditional** to the program that will print either the statement “Num1 is larger” or “Num2 is larger” depending on which variable holds the larger value.

Now, recompile your program and run it again to see that it correctly displays the result of the conditional.

STEP 7. PREPARE YOUR FILE FOR GRADING.

When all is working well, and you are still on the Linux machine...

- type: **script StudentName_lab2.txt** Script will keep a log of your interactive session.
Please use your name instead of “StudentName”
- At the prompt, type: **cc -o lab2 lab2.c** To compile the program into an executable named “lab2”
- At the prompt, type: **lab2** or **./lab2** as needed to run the program
- After the program run is complete,
type: **exit** To leave the script session.

NOTE: If you forget to type **exit** to leave *script*, your script file will be empty!!!

(How can you tell if the script file is empty?

Type: **ls -l**

If the script file shows a length of zero, it is empty.)

STEP 8: Copy your files to be accessible to your browser for upload into Canvas.

Use the method you prefer and works for your configuration.

STEP 9: Turn in your work.

Go to Canvas to complete the assignment and turn in your two files:

1. **lab2.c** ...the source code file
2. **StudentName_lab2.txt** ...the script file

STEP 10: LOG OFF EVERYTHING.

Type “**exit**” when you are ready to leave the Linux computer.

Close as much software and hardware as necessary for safety, depending on your location.