

# Lab 0: An Introduction to C

CSE/IT 113L

NMT Department of Computer Science and Engineering

---

“A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.”

— Douglas Adams

“The most exciting phrase to hear in science, the one that heralds new discoveries, is not ‘Eureka!’ but ‘That’s funny...’”

— Isaac Asimov

“If Java had true garbage collection, most programs would delete themselves upon execution.”

— Robert Sewell

---

## Contents

<b>1</b>	<b>Glossary</b>	<b>1</b>
<b>2</b>	<b>Introduction</b>	<b>1</b>
<b>3</b>	<b>Buddy Programming</b>	<b>1</b>
<b>4</b>	<b>Requirements</b>	<b>2</b>
4.1	Your First C Program . . . . .	2
4.2	ASCII Art . . . . .	3
4.3	Extra Credit . . . . .	5
	<b>Submitting</b>	<b>6</b>

## 1 Glossary

**ascii:** American Standard Code for Information Interchange is a character encoding format for text data in computers and on the internet. In standard ascii, there are unique values for 128 alphabetic characters, numeric characters, special characters, and control codes.

**code review:** methodical assessments of code designed to identify bugs, increase code quality, and help developers learn the source code. Code reviews for this course will begin with Lab 2.

## 2 Introduction

In this lab you will be introduced to the C programming environment, buddy programming, and creating your first program.

## 3 Buddy Programming

We will be using Buddy Programming for all of the labs in this class. Buddy programming involves discussing solutions to problems you are having with your pair programming partner from the associated prelab, but not sharing code or implementation details. You are welcome to sit together for this lab, just like you did during the prelab, and you are encouraged to discuss problems you are having and possible high-level solutions with your pair programming partner. Just remember, when sharing these solutions, do not share your code!

In addition to buddy programming, you will be scheduling appointments with your grader for each lab assignment to discuss what went well with your program and what could use improvement. In the industry, this practice is referred to as a *code review*. You can read more about the *code review* process in the course syllabus.

We would like to make it clear that even though you have a buddy programming partner for each assignment, each student should submit their own original work. Do not share your code with other students in the class, either written on paper, typed out in an email or other messaging service, or any other method that exists to disseminate information.

We do run all the labs through a program that **will** inform us of any cheating. Yes, this has happened in the past. Be good, do your own work, and be a supportive buddy!

## 4 Requirements

### 4.1 Your First C Program

Create and navigate into a lab0 directory. Open VS Code, Gedit, Atom, or your preferred code editor from the terminal.

```
$ gedit&
or
$ atom&
or
$ code .
```

**Note Bene:** You can run VS Code from the terminal by typing 'code' after adding it to the path. In order to easily add VS Code to your path, Launch VS Code. Then Open the Command Palette and type 'shell command'. Select Install 'code' command in PATH from the options. On some OS, you do not need to perform this extra step in order to use the 'code' command, but it is a common requirement for Apple OS.

VS Code, Gedit, and Atom are popular GUI based text editors that provide excellent syntax highlighting and error-checking. However, if you are a Computer Science Major, you may be interested in the classic UNIX text editors Vim and/or Emacs. However, there is a learning curve to using either one as they involve a great deal of keyboard shortcuts.

#### Exercise 1 (hello.c, lab0.script).

In your favorite editor, create a new c file called hello.c and type in the following "Hello world!" program.

```
1  #include <stdio.h>
2
3  int main (void)
4  {
5      printf ("Hello, world!\n");
6      return 0;
7  }
```

Remember to save the file before attempting to compile it.

Return to your terminal in order to compile and run your code.

**Note Bene:** In VS Code, you can open a terminal below your editor window by clicking the caution/warning icons in the bottom left of the screen. It will default to the Problem tab; simply click on

*Terminal and you should see a terminal window appear. If you are using WSL, make sure your terminal is the bash terminal from your Linux subsystem and not a Windows terminal.*

Compile:

```
$ gcc -g -Wall hello.c -o hello
```

Run:

Since you have now saved your executable to a file named 'hello', you can run it using the following command.

```
$ ./hello
```

Make a script called lab0.script that shows hello.c compiling and running.

## 4.2 ASCII Art

This week, you will be creating something fun -**ASCII Art**!

**Exercise 2** (name.c, n-out.script, ascii.c, a-out.script).

Again, open up an editor of your choice and save a blank document as name.c

Add the basics for a C file:

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5
6      return 0;
7  }
```

Now you can use printf statements to make “art” using letters, numbers, and characters.

First, you are going to write your name in ascii art (this will go in name.c). If you have a long name, simply use the first three or four letters of your name. Uppercase letters must minimally be 10 lines tall and lowercase characters must minimally be 5 lines tall. Fill the screen with as many characters of your name as possible. The letters in the name should appear horizontally

one after the other. Remember to put newline characters at the end of each line!

For example, here is the letter "K" in ascii art.

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      printf("k      k\n");
6      printf("k      k\n");
7      printf("k      k\n");
8      printf("k      k\n");
9      printf("k      k\n");
10     printf("k     k\n");
11     printf("k    k\n");
12     printf("k   k\n");
13     printf("kk\n");
14     printf("k  k\n");
15     printf("k   k\n");
16     printf("k    k\n");
17     printf("k     k\n");
18     printf("k      k\n");
19     printf("k      k\n");
20     printf("k      k\n");
21     printf("k      k\n");
22
23     return 0;
24 }
```

Next, you will create some ascii art. Name this file `ascii.c`. Your designs will be graded on effort, so get creative!

If you need some inspiration see <http://asciiart.website>.

Also, feel free to make more than one design. You have no limitations for this assignment.

After you have completed your designs (`name.c` and `ascii.c`), make a script of running your ASCII art programs. You will create two script files here, `n-out.script` and `a-out.script`.

To create the `n-out.script` file showing the compilation and execution of your `name.c` file, you will type the following into the command line:

```
$ script n-out.script
$ gcc -g -Wall name.c -o name
$ ./name
...
$ exit
```

To create the `a-out.script` file showing the compilation and execution of your `ascii.c` file, you will type the following into the command line:

```
$ script a-out.script
$ gcc -g -Wall ascii.c -o ascii
$ ./ascii
...
$ exit
```

You can verify that your scripts have been created and contain your terminal sessions with the following commands:

```
$ cat n-out.script
$ cat a-out.script
```

### 4.3 Extra Credit

The CS Department has a server on Discord which is used by multiple classes for communication and dissemination of information.

#### Exercise 3 (`discord.EXT`).

You can get **10% extra credit** for this lab by joining the department server and especially, the 113 channels.

Use <https://discord.gg/BvDT4DC> to join the server. Once you have joined the NMT CSE discord server, navigate to the Fresher Courses category and locate the `intro-programming-c` forum channel. I have created a Lab 0 - Introduction Extra Credit forum post.

Steps needed to get extra credit after joining the server:

1. Post an *answer* in the aforementioned forum post answering the question below:
  - If you could choose an age to remain forever, which age would you choose? Why? (The age you choose need not be one you have already experienced).
2. Take a screenshot of your post, rename it `discord-NAME.EXT` where `NAME` is your name and `EXT` is whatever the original file extension of your screenshot was (i.e. `png`, `jpeg`, `gif`).

## Submitting

You should submit your code as a tarball file that contains all the exercise files for this lab. The submitted file should be named (**note the lowercase**)

`cse113_firstname_lastname_lab0.tar.gz`

If you require a refresher on generating a tar archive file, instructions can be found in Section 6 - Creating Tar archives of Prelab 0.

**Upload your .tar.gz file to Canvas.**

## List of Files to Submit

1	Exercise (hello.c, lab0.script) . . . . .	2
2	Exercise (name.c, n-out.script, ascii.c, a-out.script) . . . . .	3
3	Exercise (discord.EXT) . . . . .	5

Exercises start on page 2.