



SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

CMPT 125 - Introduction to Computing Science and Programming II - Fall 2021

Lab 1.
September 15

- Basic commands in Linux
- VSCode Configuration
- Write Hello_Word.c
- Debug and Compile using VS code
- Compile using Terminal

0. If you need to remotely connect to csil lab computers check :

<http://www.sfu.ca/computing/about/support/csil/unix.html#remote-access-linux-system>

For problems about it contact helpdesk

<https://www.sfu.ca/computing/about/support/how-to-contact-helpdesk.html>

- **pwd** : will give you the address of directory you are in.
- **ls** : use it to show the files and directories that are in the directory you are in.

For opening Terminal use: **ctrl+alt+t**

```
sepidh@csil-cpu8:~$ pwd
/home/sepidh
sepidh@csil-cpu8:~$ ls
Android  Documents  Music      Public     Templates  'VirtualBox VMs'
Desktop  Downloads  Pictures   sfuhome    Videos
```

- **cd** : To navigate through the Linux directories and files. if you write **cd** and press tab it will show you possible directories you can enter using **cd**, you can write part of file name to make the search space for that smaller.
- use **cd ..** to go backward.

```
sepidh@csil-cpu8:~$ ls
Android  Documents  Music      Public     Templates  'VirtualBox VMs'
Desktop Downloads  Pictures   sfuhome    Videos

sepidh@csil-cpu8:~$ cd
Android/      Documents/    Music/        Templates/
.android/     Downloads/    Pictures/     .vagrant.d/
.cache/       .gnupg/      Public/       Videos/
.CLion/       .GoLand/     .PyCharm/    VirtualBox VMs/
.conda/       .gradle/     .RubyMine/   .vscode-server/
.config/     .IntelliJIdea/ sfuhome/
Desktop/     .local/      .ssh/

sepidh@csil-cpu8:~$ cd D
Desktop/ Documents/ Downloads/

sepidh@csil-cpu8:~$ cd Desktop
sepidh@csil-cpu8:~/Desktop$ ls
sepidh@csil-cpu8:~/Desktop$ cd ..
sepidh@csil-cpu8:~$ ls
Android  Documents  Music      Public     Templates  'VirtualBox VMs'
Desktop Downloads  Pictures   sfuhome    Videos
```

- **mkdir** : use this to create new directory
- **rm** : use these to remove files, and directories. For directories you need to use **rm -r**

```
sepidh@csil-cpu8:~$ ls
Android  Documents  Music      Public    Templates  'VirtualBox VMs'
Desktop  Downloads  Pictures   sfuhome   Videos
sepidh@csil-cpu8:~$ mkdir sepid
sepidh@csil-cpu8:~$ ls
Android  Documents  Music      Public    sfuhome     Videos
Desktop  Downloads  Pictures   sepid     Templates   'VirtualBox VMs'
sepidh@csil-cpu8:~$ rm -r sepid
sepidh@csil-cpu8:~$ ls
Android  Documents  Music      Public    Templates  'VirtualBox VMs'
Desktop  Downloads  Pictures   sfuhome   Videos
sepidh@csil-cpu8:~$
```

- **mkdir** : use this to create new directory
- **rm** : use these to remove files, and directories. For directories you

need to use **rm -r**

```
sepidh@csil-cpu8:~$ ls
Android  Documents  Music      Public    Templates  'VirtualBox VMs'
Desktop  Downloads  Pictures   sfuhome   Videos
sepidh@csil-cpu8:~$ mkdir sepid
sepidh@csil-cpu8:~$ ls
Android  Documents  Music      Public    sfuhome    Videos
Desktop  Downloads  Pictures   sepid     Templates  'VirtualBox VMs'
sepidh@csil-cpu8:~$ rm -r sepid
sepidh@csil-cpu8:~$ ls
Android  Documents  Music      Public    Templates  'VirtualBox VMs'
Desktop  Downloads  Pictures   sfuhome   Videos
sepidh@csil-cpu8:~$
```

- *: means anything. for example if you say `rm file*.txt` you mean remove all file“something”.txt

```
sepidh@csil-cpu8:~$ ls
Android  Downloads  file.txt  Public  Videos
Desktop  file2.txt  Music     sfuhome 'VirtualBox VMs'
Documents file3.txt  Pictures  Templates
sepidh@csil-cpu8:~$ rm file*.txt
sepidh@csil-cpu8:~$ ls
Android  Documents  Music     Public  Templates 'VirtualBox VMs'
Desktop  Downloads  Pictures  sfuhome  Videos
sepidh@csil-cpu8:~$
```


- **cp:** use **cp** for copying file and directories, for directories again you need to do **cp -r**. Same as other command tab can give you possible options for being copied.

```
sepidh@csil-cpu8:~$ ls
Android  Downloads  file.txt  Pictures  sfuhome  Videos
Desktop  Downloads  Music     Public    Templates 'VirtualBox VMs'
sepidh@csil-cpu8:~$ cp file.txt file2.txt
sepidh@csil-cpu8:~$ ls
Android  Downloads  Music     sfuhome  'VirtualBox VMs'
Desktop  file2.txt  Pictures  Templates
Documents file.txt   Public    Videos
sepidh@csil-cpu8:~$ cp -r Do
Documents/ Downloads/
sepidh@csil-cpu8:~$ cp -r Downloads Downloads_p
sepidh@csil-cpu8:~$ ls
Android  Downloads  file.txt  Public  Videos
Desktop  Downloads_p Music     sfuhome 'VirtualBox VMs'
Documents file2.txt  Pictures  Templates
sepidh@csil-cpu8:~$
```

- **mv:** use mv to move files and directories.

```
sepidh@csil-cpu8:~$ ls
Android  Downloads  file.txt  Public  Videos
Desktop  Downloads_p Music     sfuhome 'VirtualBox VMs'
Documents file2.txt  Pictures  Templates
sepidh@csil-cpu8:~$ mv file.txt Downloads/
sepidh@csil-cpu8:~$ ls
Android  Downloads  Music     sfuhome  'VirtualBox VMs'
Desktop  Downloads_p Pictures    Templates
Documents file2.txt  Public     Videos
sepidh@csil-cpu8:~$ cd Downloads
sepidh@csil-cpu8:~/Downloads$ ls
file.txt
sepidh@csil-cpu8:~/Downloads$ cd ..
sepidh@csil-cpu8:~$ mv Downloads_p Do
Documents/ Downloads/ Downloads_p/
sepidh@csil-cpu8:~$ mv Downloads_p Downloads
sepidh@csil-cpu8:~$ ls
Android  Documents  file2.txt  Pictures  sfuhome  Videos
Desktop  Downloads  Music     Public    Templates 'VirtualBox VMs'
sepidh@csil-cpu8:~$ cd Downloads/
sepidh@csil-cpu8:~/Downloads$ ls
Downloads_p  file.txt
sepidh@csil-cpu8:~/Downloads$
```

- **find:** use to find files and directories.

```
sepidh@csil-cpu8:~$ ls
Android    Downloads  Pictures   sfuhome   'VirtualBox VMs'
Desktop    file2.txt  Public     Templates
Documents  Music      sepid.txt  Videos
sepidh@csil-cpu8:~$ find sepid.txt
sepid.txt
sepidh@csil-cpu8:~$ find *.txt
file2.txt
sepid.txt
sepidh@csil-cpu8:~$
```

- **find:** use to find files and directories.

```
sepidh@csil-cpu8:~$ ls
Android    Downloads  Pictures   sfuhome    'VirtualBox VMs'
Desktop    file2.txt  Public     Templates
Documents  Music      sepid.txt  Videos
sepidh@csil-cpu8:~$ find sepid.txt
sepid.txt
sepidh@csil-cpu8:~$ find *.txt
file2.txt
sepid.txt
sepidh@csil-cpu8:~$
```


- **tar: use tar for zip and unzip or append a file to a zip file.**

to practice tar check : <https://www.linuxtechi.com/17-tar-command-examples-in-linux/>

For more on commands check : <https://www.hostinger.com/tutorials/linux-commands>

VSCode Configuration, and run “Hello Word”

Open VS code, go to extensions and install c/c++ extension

The screenshot displays the Visual Studio Code interface. On the left, the 'EXTENSIONS' sidebar is open, showing a search for 'C++'. The 'C/C++' extension by Microsoft is highlighted with a red box. Below it, other extensions like 'C/C++ IntelliSense, debugging, and code browsing' are listed. A red box also highlights the 'Install' button for the 'C/C++' extension. On the right, the 'Extension: C/C++ - Visual Studio Code' page is shown. It features the 'C/C++' logo, the version 'v1.6.0', and the publisher 'Microsoft'. The 'Install' button is highlighted with a red box. Below the 'Install' button, there are tabs for 'Details', 'Feature Contributions', 'Changelog', and 'Runtime Status'. The 'Details' tab is selected, showing the extension's description: 'C/C++ IntelliSense, debugging, and code browsing.' and a list of links: 'Repository', 'Issues', 'Documentation', 'Code Samples', and 'Offline Installers'. The 'Live Share' button is also visible, with 'enabled' next to it. The bottom section of the page is titled 'Overview and tutorials' and includes a link to 'C/C++ extension overview'.

Extension: C/C++ - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXTENSION: C/C++

C/C++ v1.6.0
Microsoft | 23,039,341 | ★★★★★ (439)
C/C++ IntelliSense, debugging, and code browsing.
Uninstalled Install

Details Feature Contributions Changelog Runtime Status

C/C++ for Visual Studio Code

Repository | Issues | Documentation | Code Samples | Offline Installers

Live Share enabled

The C/C++ extension adds language support for C/C++ to Visual Studio Code, including features such as IntelliSense, debugging, and code browsing.

Overview and tutorials

- C/C++ extension overview

C/C++ extension tutorials per compiler and platform

- Microsoft C++ compiler (MSVC) on Windows

VSCode Configuration, and run “Hello Word”

go to File, create new file. paste “**hello world**” program

below in the file and save is as hello.c on your machine.

hello world:

```
#include <stdio.h>

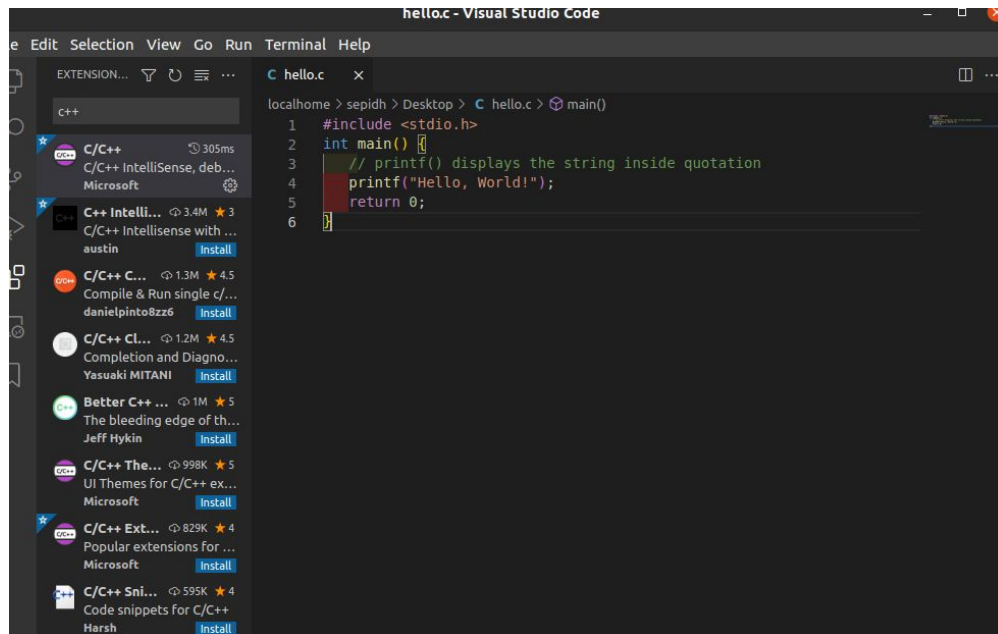
int main() {

    // printf() displays the string inside
    quotation

    printf("Hello, World!\n");

    return 0;

}
```



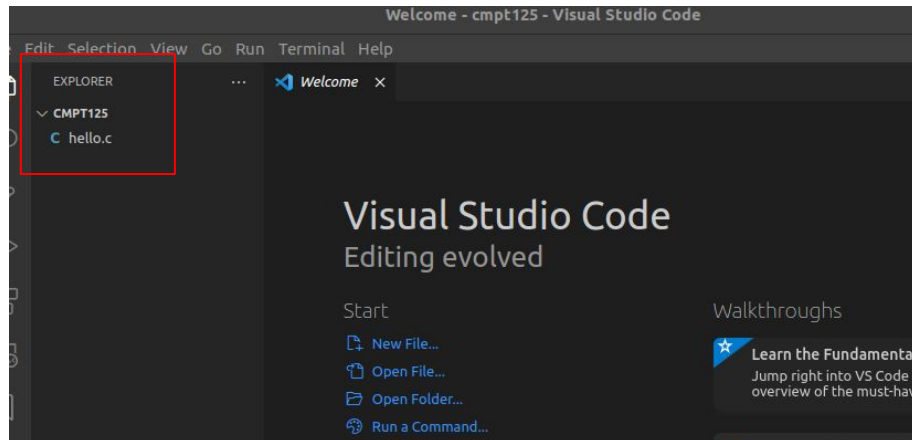
VSCode Configuration, and run “Hello Word”

Then go to the directory that file is saved there. right click
and select open terminal. in the terminal write code . and
enter.

```
sepidh@cs-vml-42:~/Desktop/cmpt125$ code .
```

new vs code should open and your hello.c should be there.

chose it.



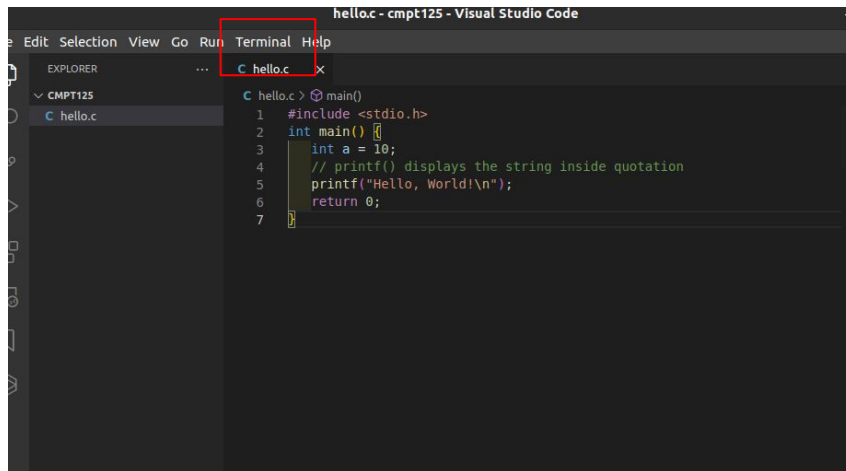
VSCode Configuration, and run “Hello Word”

from terminal choose “Run Build Task”.

then choose “C/C++: gcc build active file”

now your code is ready to run and debug using

“Run” in vs code.



The screenshot shows the Visual Studio Code interface with a C program named 'hello.c' open. The 'Run' menu is highlighted with a red box. The code in the editor is as follows:

```
hello.c - cmpt125 - Visual Studio Code
Edit Selection View Go Run Terminal Help
EXPLORER
  CMPT125
    hello.c
C hello.c
hello.c > main()
1 #include <stdio.h>
2 int main()
3 {
4     int a = 10;
5     // printf() displays the string inside quotation
6     printf("Hello, World!\n");
7     return 0;
}
```

VSCode Configuration

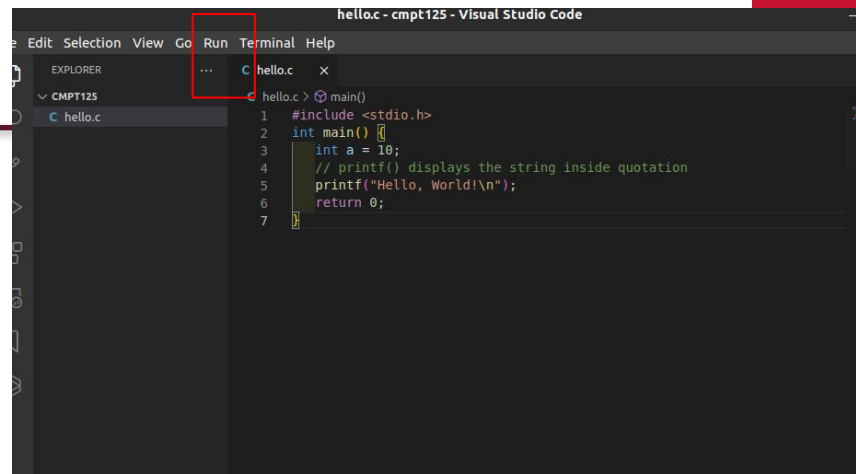
in “Run”, menu at top choose “Run Without Debugging”.

Then choose C++(GDB, LLDB), and

then choose “gcc build and debug active file”.

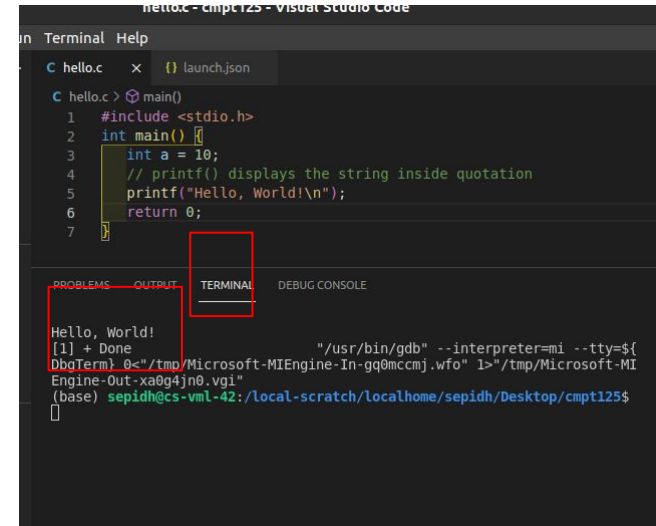
at terminal down you should be able to see output.

Change “Hello, World!\n” to any sentence you like and
see how printed output changes.



A screenshot of the Visual Studio Code interface. The top menu bar shows 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help'. The 'Run' menu is highlighted with a red box. Below the menu bar, the Explorer sidebar on the left shows a project named 'CMPT125' with a file 'hello.c' selected. The main editor area displays the contents of 'hello.c', which includes a C program that prints 'Hello, World!\n'.

```
hello.c - cmpt125 - Visual Studio Code
Edit Selection View Go Run Terminal Help
EXPLORER
  CMPT125
    hello.c
EDITOR
  hello.c
1 #include <stdio.h>
2 int main() {
3     int a = 10;
4     // printf() displays the string inside quotation
5     printf("Hello, World!\n");
6     return 0;
7 }
```



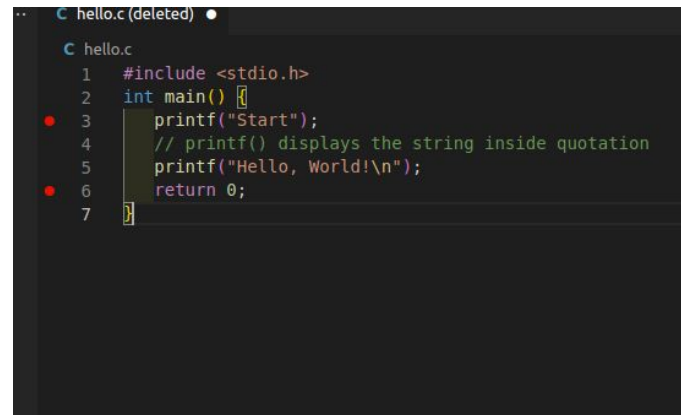
A screenshot of the Visual Studio Code interface showing the output of the program. The 'TERMINAL' tab is selected, and the output shows 'Hello, World!' followed by '[1] + Done'. The 'PROBLEMS' and 'OUTPUT' tabs are also visible. The 'Run' menu is highlighted with a red box. The terminal output shows the command 'gcc build and debug active file' being executed.

```
hello.c - cmpt125 - Visual Studio Code
Run Terminal Help
C hello.c x {} launch.json
C hello.c
1 #include <stdio.h>
2 int main() {
3     int a = 10;
4     // printf() displays the string inside quotation
5     printf("Hello, World!\n");
6     return 0;
7 }
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Hello, World!
[1] + Done
"/usr/bin/gdb" --interpreter=mi -tty=${
DbgTerm} 0<" /tmp/Microsoft-MIEngine-In-gg0mccmj.wfo" l>" /tmp/Microsoft-MI
Engine-Out-xa0g4jn0.vgi"
(base) sepidh@cs-vml-42: /local-scratch/localhome/sepidh/Desktop/cmpt125$
```

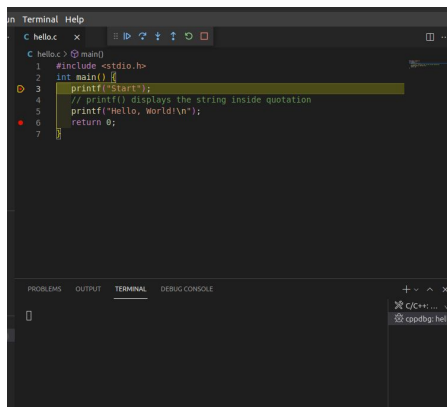
VSCode Configuration, and run “Hello Word”

To debug the code first click on the lines that you want your code stop on. you should see small red circles there after clicking.

Then from Run menu choose Start Debugging.



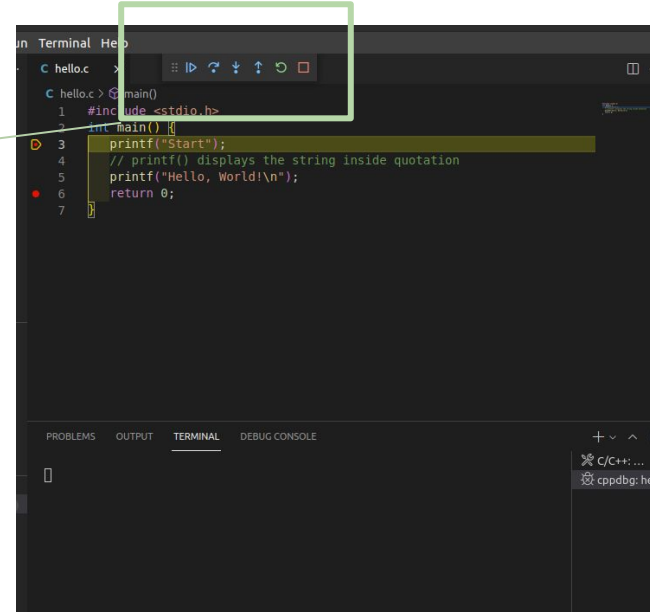
```
hello.c (deleted)
C hello.c
1 #include <stdio.h>
2 int main() {
3     printf("Start");
4     // printf() displays the string inside quotation
5     printf("Hello, World!\n");
6     return 0;
7 }
```



VSCode Configuration, and run "Hello Word"

It should stop in the selected lines.

Then by using options at top you can navigate through your code and move inside your code.



Run "Hello Word" using terminal.

Other way to run your code is using terminal and gcc there.

open a terminal and go the the directory that your file is in.

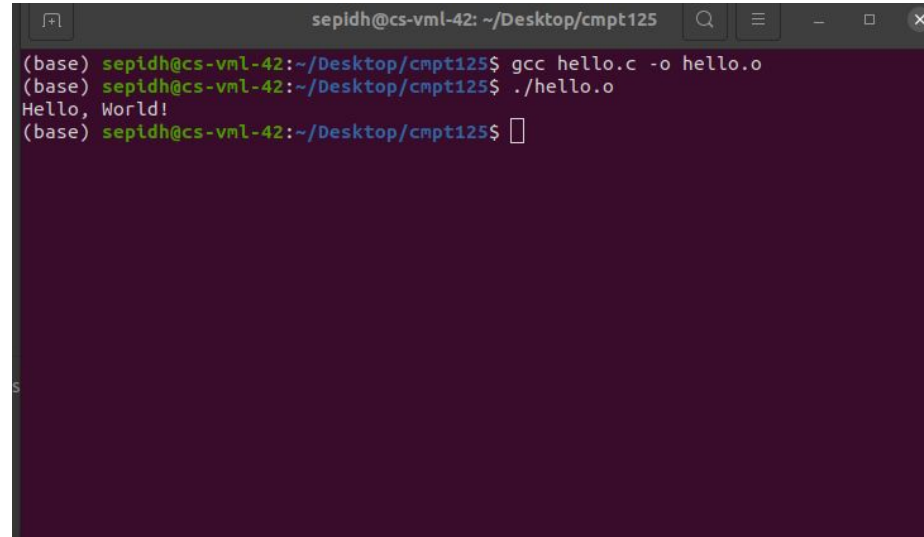
then :

write this command in the terminal:

```
gcc hello.c -o hello.o
```

with above command you created hello.o file. then you can

run it using command `./hello.o` and see the output.

A terminal window with a dark purple background. The title bar shows 'sepidh@cs-vml-42: ~/Desktop/cmpt125'. The terminal content shows three lines of interaction: a prompt '(base) sepidh@cs-vml-42:~/Desktop/cmpt125\$' followed by the command 'gcc hello.c -o hello.o'; a second prompt followed by the command './hello.o'; and the output 'Hello, World!'. The third line shows a prompt followed by a cursor.

```
sepidh@cs-vml-42: ~/Desktop/cmpt125
(base) sepidh@cs-vml-42:~/Desktop/cmpt125$ gcc hello.c -o hello.o
(base) sepidh@cs-vml-42:~/Desktop/cmpt125$ ./hello.o
Hello, World!
(base) sepidh@cs-vml-42:~/Desktop/cmpt125$
```

Some More Examples:

Do These Examples on your own and see what will happen.

```
1  #include <stdio.h>
2  int main() {
3
4      int number1, number2, sum;
5
6      printf("Enter two integers: ");
7      number1 =10;
8      number2=20;
9
10
11
12     // calculating sum
13     sum = number1 + number2;
14
15     printf("%d + %d = %d", number1, number2, sum);
16     return 0;
17 }
```

Some More Examples:

Do These Examples on your own and see what will happen.

```
1  #include <stdio.h>
2  int main() {
3      int number;
4
5      printf("Enter an integer: ");
6
7      // reads and stores input
8      scanf("%d", &number);
9
10     // displays output
11     printf("You entered: %d", number);
12
13     return 0;
14 }
```