University of Windsor
Faculty of Science
School of Computer Science

COMP-2650
Computer Architecture I: Digital Design
Fall 2024

Lab#	Date	Title	Due	Grade Release
Lab02	Week 02	**Programming Environment Setup**	Sept. 24, Tuesday Midnight	Sept. 30

The objectives of the first lab will be for you to set up a programming environment, specifically to have an integrated development environment (IDE) installed and functioning on your local drive in the lab or on your own desktop or laptop computer. Also, you will be developing on the program specification of the lab assignments that you will be gradually completing throughout this term.

Step 1. Environment Setup

In computer engineering, there are hardware description languages (HDL) such as VHDL and Verilog that are specialized computer languages used to describe the structure and behavior of electronic circuits, and most commonly, digital logic circuits. However, in computer science and this course mainly, we rely on schematic-based design methods that create a representation of functionality. We literally use pen and paper to draw circuits and gates. Therefore, we do not cover any HDL. We instead simulate the circuit's functionality via writing programs using a general-purpose programing language to make sure that our designed circuit is working correctly.

We will use C as the programming language. C++ is an extension to the C and has more features. We are not using those features in this course and, therefore, C++ is similar to C for us. Although we write C++ program, we only use C language. This way, you will see some limitations of C language and will be motivated to move on to C++ language in future.

For IDE, we use none, that is using a text editor like vi and C/C++ compiler gcc. You are however free to adopt any programming environment such as NetBeans, CodeBlocks or Eclipse. Make sure you can start up a Hello World program in C/C++ and run it.

1.1.C/C++ on Unix-based/like System (macOS, Linux)

1.1.1. Your Laptop or PC (Local)

If you have a mac laptop or a pc with linux-base/like operating system, move on to next section (Step 2.)

1.1.2. School of Computer Science (Remote)

You can connect to the computer systems (Debian GNU/Linux 11) available in the school of computer science to work remotely with vi and gcc. Here are the steps:

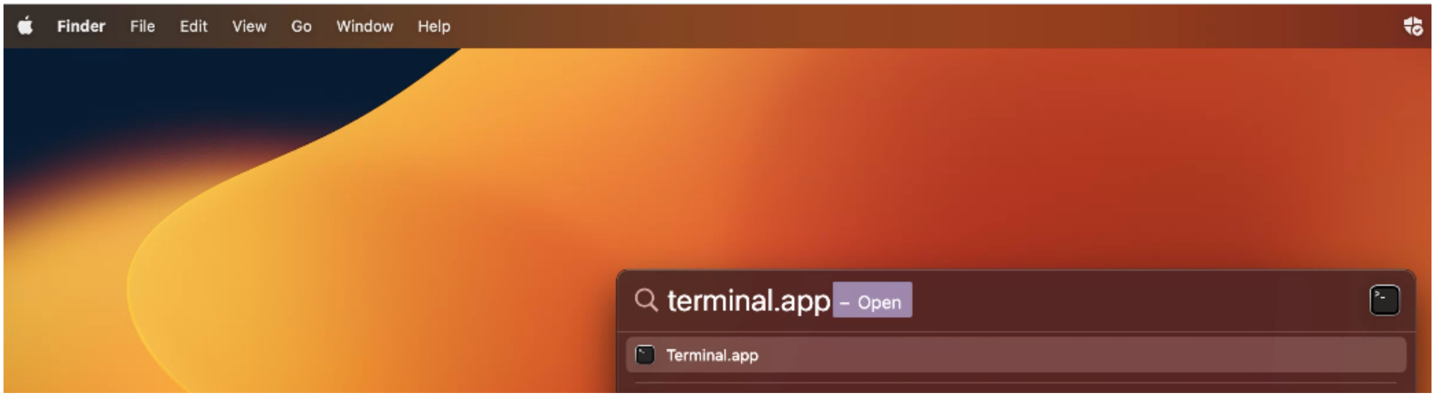
1.1.2.1. Connect to VPN

If you're connecting from home, anywhere outside or inside the campus, you must connect via the UWindsor's VPN server. You can find help from the school's user guide at https://help.cs.uwindsor.ca/books/cs-help/page/vpn.

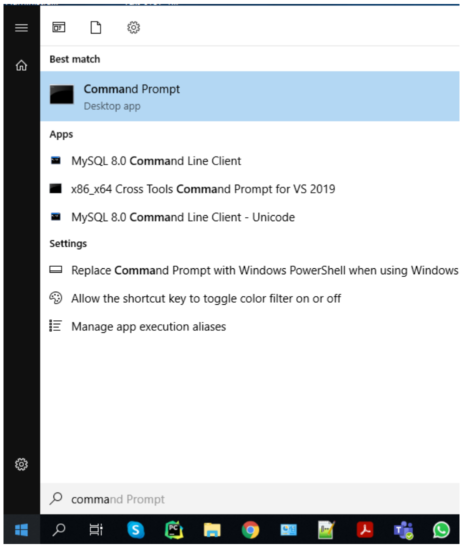
1.1.2.2.Connect to the school's systems

You need an application that can create a Secure Shell (SSH) to NoMachine and download/upload your files.

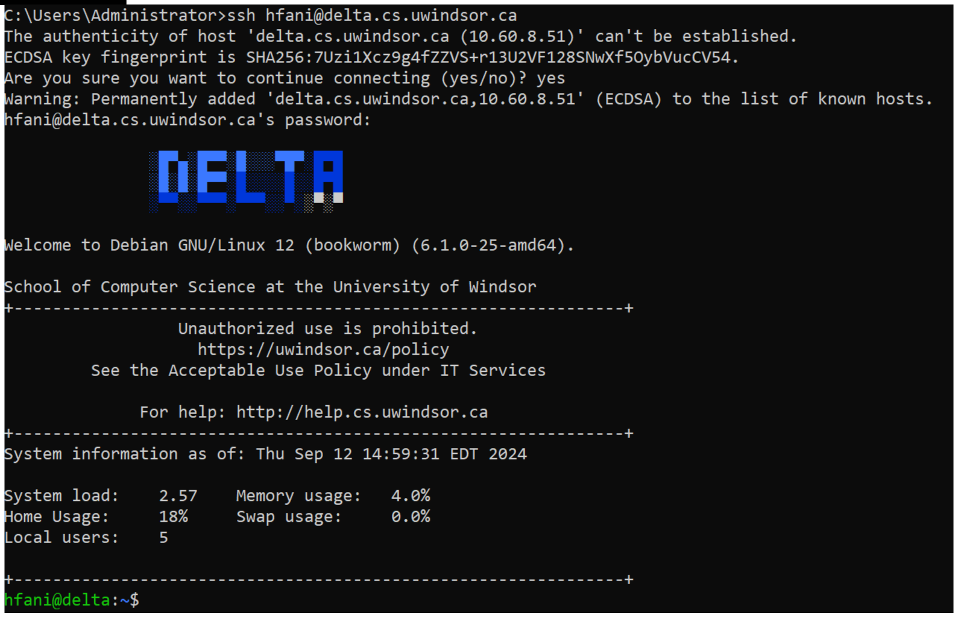
From macOS, open Terminal application using the Finder:



From Windows, enable SSH by following the steps here https://www.howtogeek.com/336775/how-to-enable-and-use-windows-10s-built-in-ssh-commands/. When enabled, open Command Prompt by clicking the Start button and typing in Command Prompt, and then select it from the list when it appears:


Best match
Command Prompt
Desktop app
Apps
MySQL 8.0 Command Line Client
x86_x64 Cross Tools Command Prompt for VS 2019
MySQL 8.0 Command Line Client - Unicode
Settings
Replace Command Prompt with Windows PowerShell when using Windows
Allow the shortcut key to toggle color filter on or off
Manage app execution aliases
command Prompt

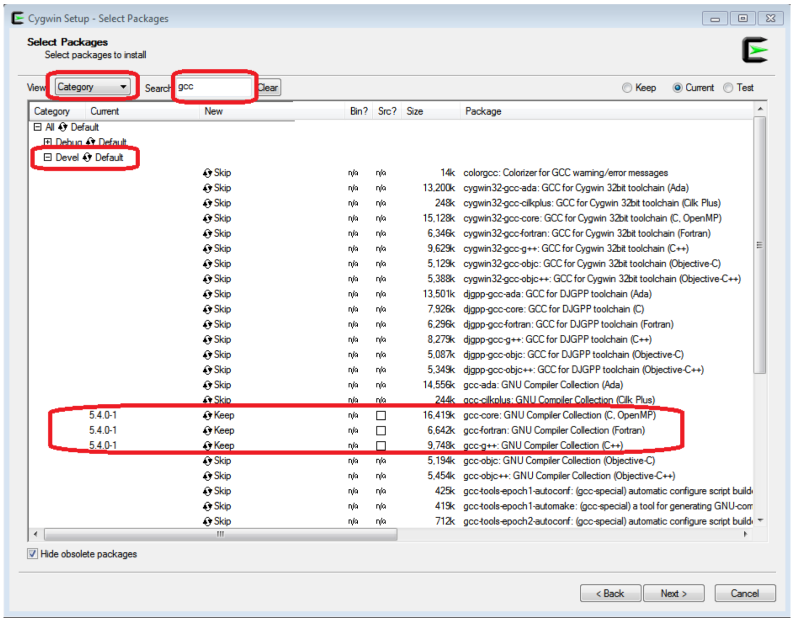
Whether on macOS or Windows, enter the SSH command `ssh uwinid@delta.cs.uwindsor.ca`. Replace `uwinid` with your own. When asked for a password, put your uwin password. This will connect to the server and the connection will look like the following. For instance, my uwinid is `hfani`, so I entered `hfani@delta.cs.uwindsor.ca`



```
C:\Users\Administrator>ssh hfani@delta.cs.uwindsor.ca
The authenticity of host 'delta.cs.uwindsor.ca (10.60.8.51)' can't be established.
ECDSA key fingerprint is SHA256:7Uzi1Xcz9g4fZZVS+r13U2VF128SNwXf5OybVucCV54.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'delta.cs.uwindsor.ca,10.60.8.51' (ECDSA) to the list of known hosts.
hfani@delta.cs.uwindsor.ca's password:

               DELTA

Welcome to Debian GNU/Linux 12 (bookworm) (6.1.0-25-amd64).

School of Computer Science at the University of Windsor
+---------------------------------------------------------------+
                  Unauthorized use is prohibited.
                    https://uwindsor.ca/policy
         See the Acceptable Use Policy under IT Services

              For help: http://help.cs.uwindsor.ca
+---------------------------------------------------------------+
System information as of: Thu Sep 12 14:59:31 EDT 2024

System load:    2.57    Memory usage:   4.0%
Home Usage:     18%     Swap usage:     0.0%
Local users:    5

+---------------------------------------------------------------+
hfani@delta:~$
```

1.2.C/C++ on Windows

Unfortunately, Windows operating system does not have C/C++ compiler; you need to install. There are many options including MinGW[1] and CygWin[2]. During the installation, you need to select gcc packages:



Cygwin Setup - Select Packages

Select Packages
Select packages to install

View: Category | Search: gcc | Clear | ○ Keep ● Current ○ Test

Category	Current	New	Bin?	Src?	Size	Package
⊟ All Default						
⊞ Debug Default						
⊟ Devel Default						
		Skip	n/a	n/a	14k	colorgcc: Colorizer for GCC warning/error messages
		Skip	n/a	n/a	13,200k	cygwin32-gcc-ada: GCC for Cygwin 32bit toolchain (Ada)
		Skip	n/a	n/a	248k	cygwin32-gcc-cilkplus: GCC for Cygwin 32bit toolchain (Cilk Plus)
		Skip	n/a	n/a	15,128k	cygwin32-gcc-core: GCC for Cygwin 32bit toolchain (C, OpenMP)
		Skip	n/a	n/a	6,346k	cygwin32-gcc-fortran: GCC for Cygwin 32bit toolchain (Fortran)
		Skip	n/a	n/a	9,629k	cygwin32-gcc-g++: GCC for Cygwin 32bit toolchain (C++)
		Skip	n/a	n/a	5,129k	cygwin32-gcc-objc: GCC for Cygwin 32bit toolchain (Objective-C)
		Skip	n/a	n/a	5,388k	cygwin32-gcc-objc++: GCC for Cygwin 32bit toolchain (Objective-C++)
		Skip	n/a	n/a	13,501k	djgpp-gcc-ada: GCC for DJGPP toolchain (Ada)
		Skip	n/a	n/a	7,926k	djgpp-gcc-core: GCC for DJGPP toolchain (C)
		Skip	n/a	n/a	6,296k	djgpp-gcc-fortran: GCC for DJGPP toolchain (Fortran)
		Skip	n/a	n/a	8,279k	djgpp-gcc-g++: GCC for DJGPP toolchain (C++)
		Skip	n/a	n/a	5,087k	djgpp-gcc-objc: GCC for DJGPP toolchain (Objective-C)
		Skip	n/a	n/a	5,349k	djgpp-gcc-objc++: GCC for DJGPP toolchain (Objective-C++)
		Skip	n/a	n/a	14,556k	gcc-ada: GNU Compiler Collection (Ada)
		Skip	n/a	n/a	244k	gcc-cilkplus: GNU Compiler Collection (Cilk Plus)
	5.4.0-1	Keep	n/a	□	16,419k	gcc-core: GNU Compiler Collection (C, OpenMP)
	5.4.0-1	Keep	n/a	□	6,642k	gcc-fortran: GNU Compiler Collection (Fortran)
	5.4.0-1	Keep	n/a	□	9,748k	gcc-g++: GNU Compiler Collection (C++)
		Skip	n/a	n/a	5,194k	gcc-objc: GNU Compiler Collection (Objective-C)
		Skip	n/a	n/a	5,454k	gcc-objc++: GNU Compiler Collection (Objective-C++)
		Skip	n/a	n/a	425k	gcc-tools-epoch1-autoconf: (gcc-special) automatic configure script build
		Skip	n/a	n/a	419k	gcc-tools-epoch1-automake: (gcc-special) a tool for generating GNU-com
		Skip	n/a	n/a	712k	gcc-tools-epoch2-autoconf: (gcc-special) automatic configure script build

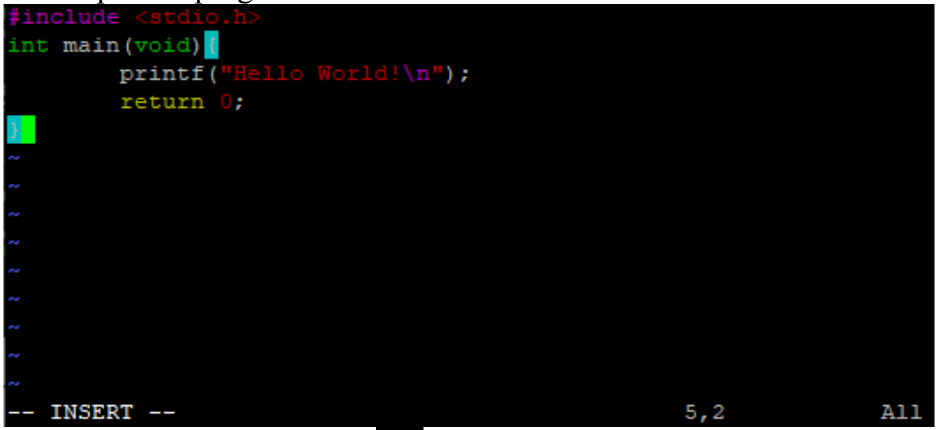
☑ Hide obsolete packages

Step 2. Hello World!

To write our first program in C/C++ programing language, we need a text editor to write our C/C++ code and a compiler for C language. vi (/ˌviːˈaɪ/ pronounce v eye) is the most common text editor which is available on almost all UNIX-based/like OS. On Windows, you can use Notepad. You can create an empty text file by typing vi followed by the filename (hello.c):

```
hfani@alpha:~$ vi hello.c
```

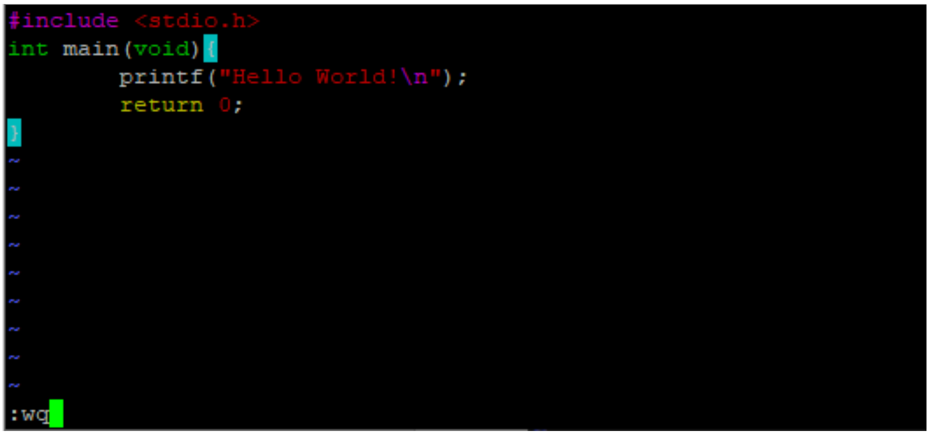
To start inserting new characters, you should put vi in the --INSERT-- mode by pressing SHIFT I. Then start typing the simplest C program as follow:



```
#include <stdio.h>
int main(void){
        printf("Hello World!\n");
        return 0;
}
~
~
~
~
~
~
~
~
~
-- INSERT --                                        5,2           All
```

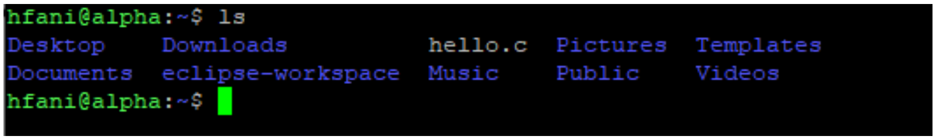
To exit the edit mode, press ESC key. In order to save the file, you should press SHIFT : after which vi needs a command. Enter wq , which means write and quit.

[1] https://www.mingw-w64.org/
[2] http://cygwin.com/



```
#include <stdio.h>
int main(void){
        printf("Hello World!\n");
        return 0;
}
~
~
~
~
~
~
~
~
~
:wq
```

To ensure that your file has been saved properly, use the ls command to list the files.

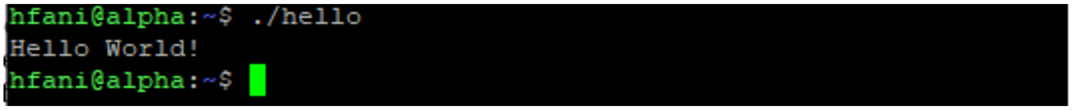


```
hfani@alpha:~$ ls
Desktop    Downloads          hello.c  Pictures  Templates
Documents  eclipse-workspace  Music    Public    Videos
hfani@alpha:~$ 
```

To compile our program, we use the built-in C compiler application, named cc, as follows:

```
hfani@alpha:~$ cc hello.c -o hello
```

And now, the executable file is ready to be launched and run.



```
hfani@alpha:~$ ./hello
Hello World!
hfani@alpha:~$ 
```

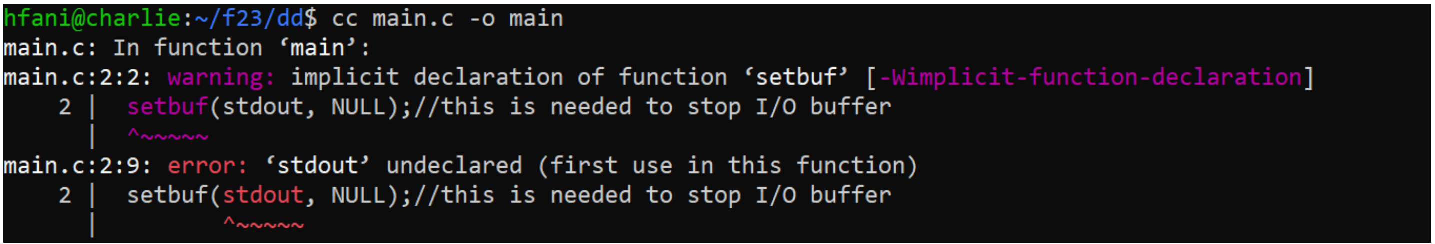
Step3. Writing the First Program

Now, let’s write a quick program that accepts two Boolean values and returns the AND of them. Recall from math, the truth table for AND operation, also shown as ‘∧’ or ‘&’ is as follows:

x	y	x AND y	x ∧ y	x & y
false	false	false	false	false
false	true	false	false	false
true	false	false	false	false
true	true	true	true	true

We know that C++ language has a type for Boolean values, called bool, that accept false and true. However, C language does not have it. So, we can use the standard type int for our program and use 0 for false and 1 for true values.

```
int main(void) {
    setbuf(stdout, NULL);//This is needed to stop I/O buffer
     int x;
     int y;
     scanf("%d", &x);
     scanf("%d", &y);
     printf("%d AND %d is %d\n", x, y, x & y);
     return 0;
}
```

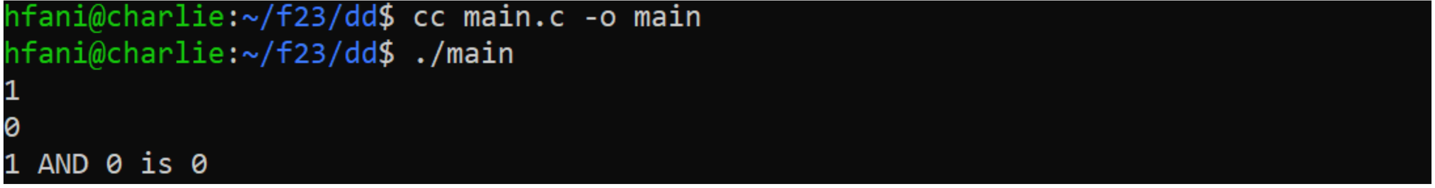


```
hfani@charlie:~/f23/dd$ cc main.c -o main
main.c: In function ‘main’:
main.c:2:2: warning: implicit declaration of function ‘setbuf’ [-Wimplicit-function-declaration]
    2 |  setbuf(stdout, NULL);//this is needed to stop I/O buffer
      |  ^~~~~~
main.c:2:9: error: ‘stdout’ undeclared (first use in this function)
    2 |  setbuf(stdout, NULL);//this is needed to stop I/O buffer
      |         ^~~~~~
```

As you see, the build failed for our program due to 2 errors in finding scanf and printf functions. Because the linker could not find the declaration for these functions. Recall from the C++ program language course, we know that these two functions are in the standard I/O library in stdio.h. Let’s include this library and built the program again:

```
01 #include <stdio.h>
02 int main(void) {
03     setbuf(stdout, NULL);//This is needed to stop I/O buffer
04      int x;
05      int y;
06      scanf("%d", &x);
07      scanf("%d", &y);
08      printf("%d AND %d is %d\n", x, y, x & y);
09      return 0;
10 }
```

As seen below, our program is compiled with no issue. It is running and waiting for the inputs, instructed by lines #06 and #07 of the program. When entering the Boolean values, the program outputs the AND result:



```
hfani@charlie:~/f23/dd$ cc main.c -o main
hfani@charlie:~/f23/dd$ ./main
1
0
1 AND 0 is 0
```

Do run the program and try different inputs, e.g., 1 AND 1, 0 AND 0, 0 AND 1, and check whether the program correctly calculates the AND operation.

Step 3. Lab Assignment

You should complete the above program that firstly outputs a menu of commands as follows:

Enter the command number:

0) Exit
1) AND
2) OR

Based on the chosen number of commands by the user, the program should then ask for the input(s). For instance, if a user selects (1), the program should accept two inputs as follows:

x =
y =

When the user enters the two Boolean values (0 or 1), the program should

1) apply the AND command on the input x and y
2) print the result
3) comes back to the main menu. If the user selects (0), the program ends.
4) restrict the user to enter inputs within the range {0,1}. For instance, if the user enters 2, -1, …, print out

an error message and come back to ask for new inputs.

Deliverables

Prepare and submit the program in one single zip file lab02_uwinid.zip containing the following items:

1. The code file (main.c or main.cpp) and executable file (main.exe in Windows or main in Unix/macOS)
2. The result of the four commands in the file results.pdf/png/jpg. Simply make a screenshot of the results.
3. *[Optional and if necessary]* A readme document in a txt file readme.txt. It explains how to build and run the program as well as any prerequisites that are needed. Please note that if your program cannot be built and run on our computer systems, you will lose marks.

(90%) lab02_hfani.zip

- (75%) main.c or main.cpp → Must be compiled and built with no error!
- (05%) main.exe or main
- (10%) results.pdf/jpg/png → Must match with the program output!
- (Optional) readme.txt

(10%) Files Naming and Formats

Please follow the naming convention as you lose marks otherwise. Instead of uwinid, use your own UWindsor account name, e.g., mine is hfani@uwindsor.ca, so, lab02_hfani.zip