LW: Setup (Hello World)

[Setup](#_tuwjuqxb6rj8)

[Windows](#_ue06oej53eg5)

[Installing a Linux environment in Windows](#_3oqhvn1ofors)

[Installing a C++ compiler in Windows](#_4yp63p18j1hr)

[Files on your Windows computer](#_j0b3f18nln7w)

[Text Editor Installation in Windows](#_phgih4w3ar7)

[Text editor options for Windows:](#_t8xnigx3g95o)

[Mac](#_u26piu6v8uzd)

[Installing a C++ Compiler in Mac](#_c4swnsymvsec)

[Text Editor Installation in Mac](#_3tikrrn91yzb)

[Text editor options for Mac](#_krqnfk5f67hg)

[Linux](#_fixfp4tgjklm)

[Installing a C++ Compiler in Linux](#_4bmbcli8z511)

[Text Editor Installation in Linux](#_chhnw0uie5eb)

[Text editor options for Linux](#_prw43tbeomkd)

[UNIX Tutorial](#_q0f13spyd7g6)

[Write, Compile and Run a C++ Program](#_217sxilaocu6)

[Submission](#_an8aiajaw3yb)

# 

# Setup

Jump to instructions for your Operating System:

* [Windows](#_ue06oej53eg5)
* [Mac](#_u26piu6v8uzd)
* [Linux](#_fixfp4tgjklm)

## Windows

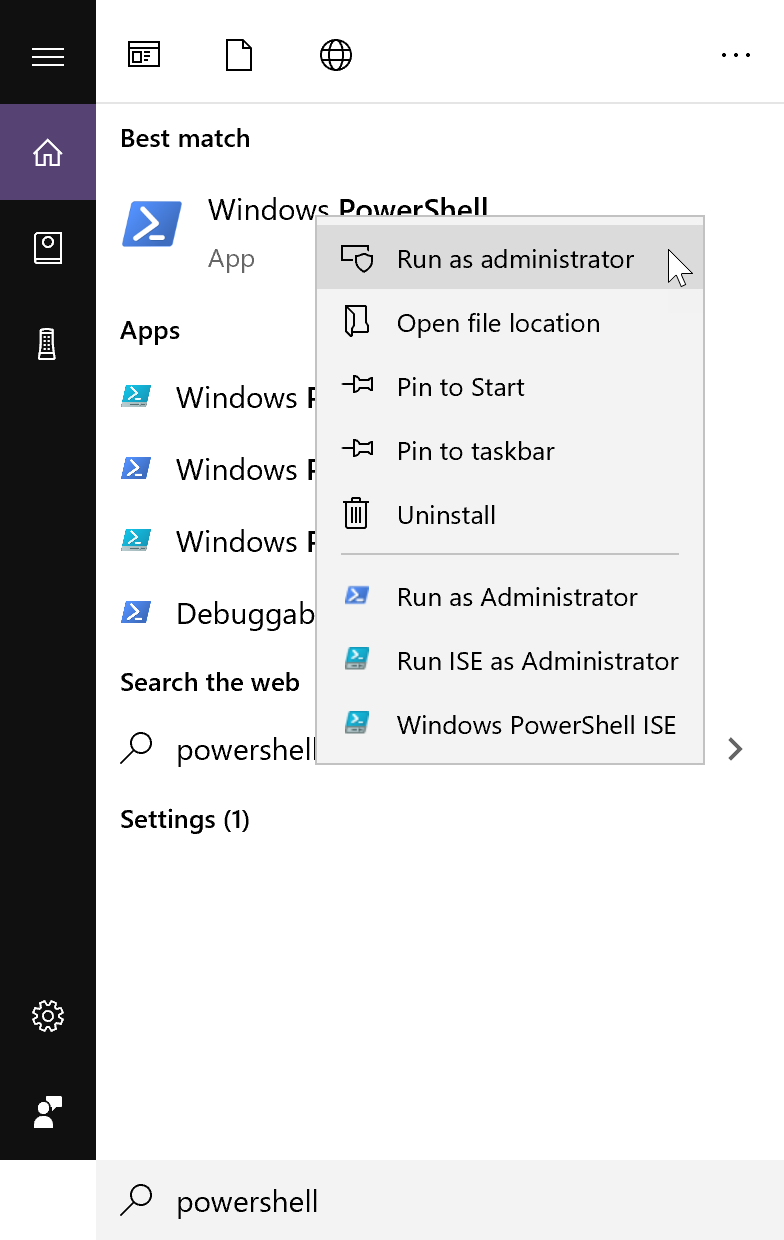
For the first week only, you can complete this labwork on your own and do not have to attend a lab session to get credit.

We will be using a common compiling environment for all students in the class, regardless of whether you use macOS, Windows, or a Linux distribution. The way we have decided to do this is to have all students compile in a command-line environment. While there are other environments we could use (e.g. Cygwin), using Ubuntu will allow access to tools to better debug your code.

This guide allows you to edit your source code in Windows and then compile the code with the installed tools in Ubuntu. Tutorials about how to use a command-line environment to navigate the file system and compile your source code can be found in [Part 2 (UNIX Tutorial)](#_q0f13spyd7g6) and [Part 3 (Write, Compile and Run a C++ Program)](#_217sxilaocu6)Part 3.

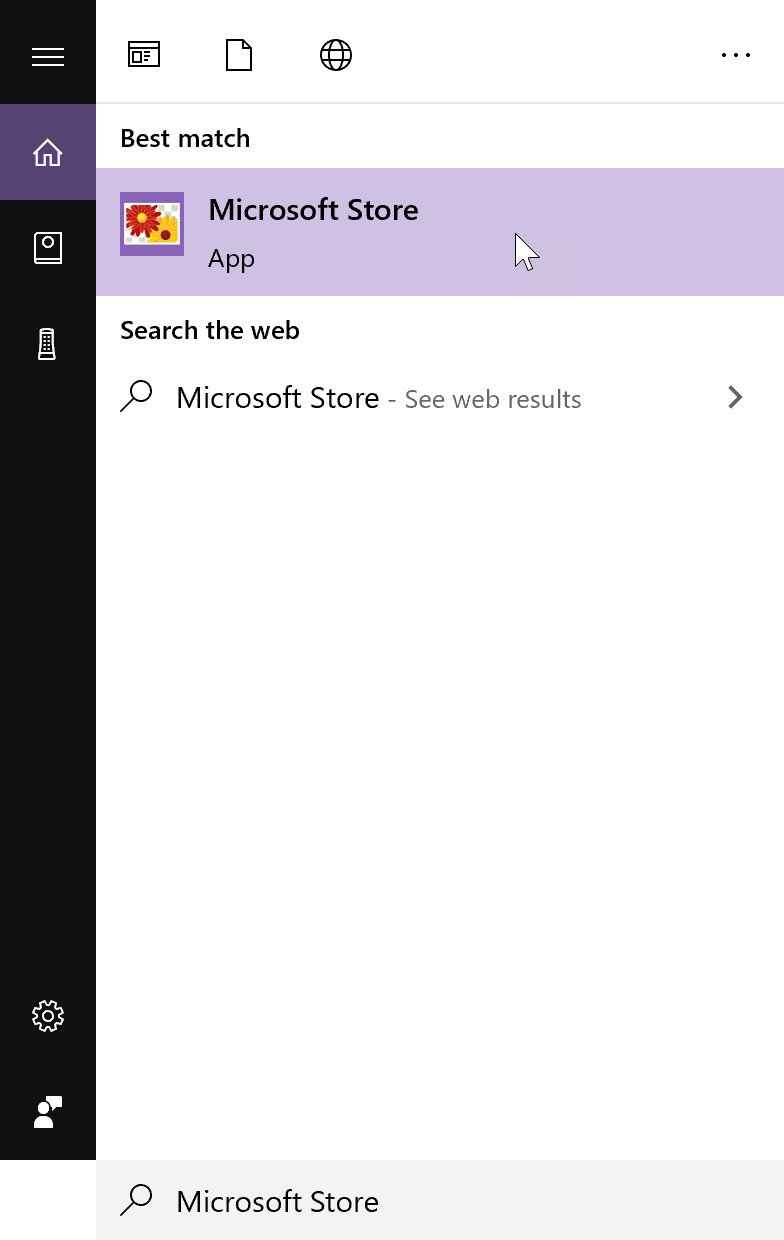
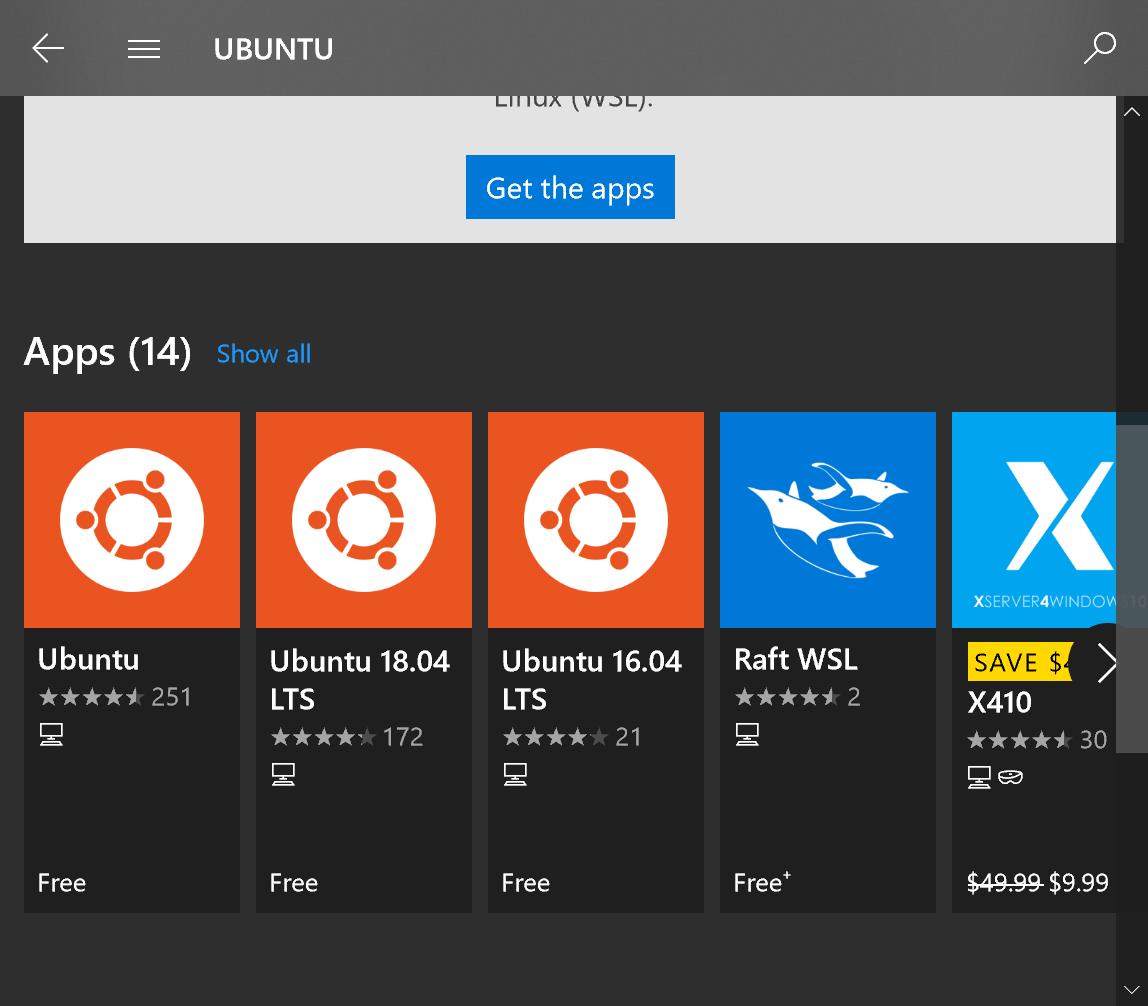
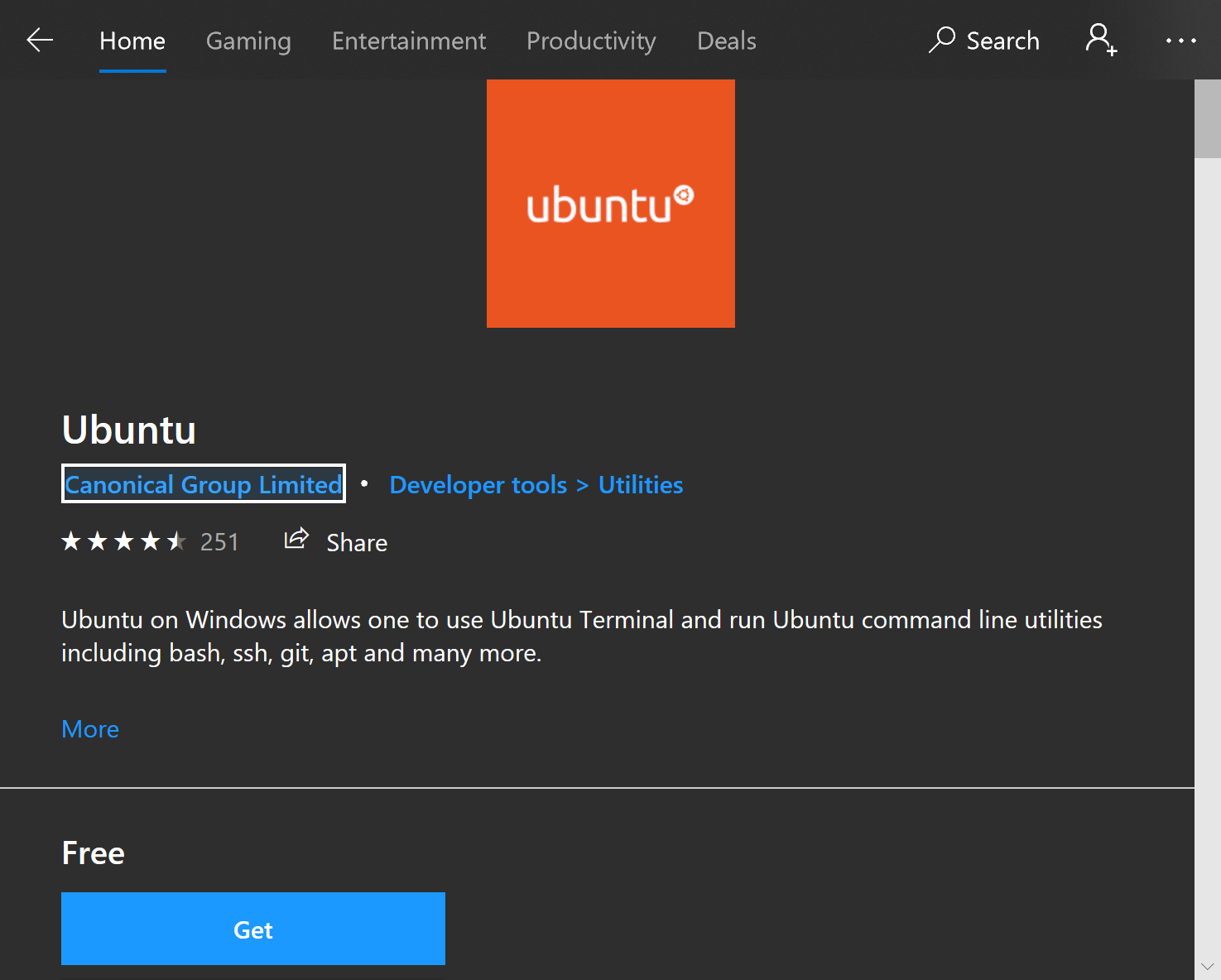
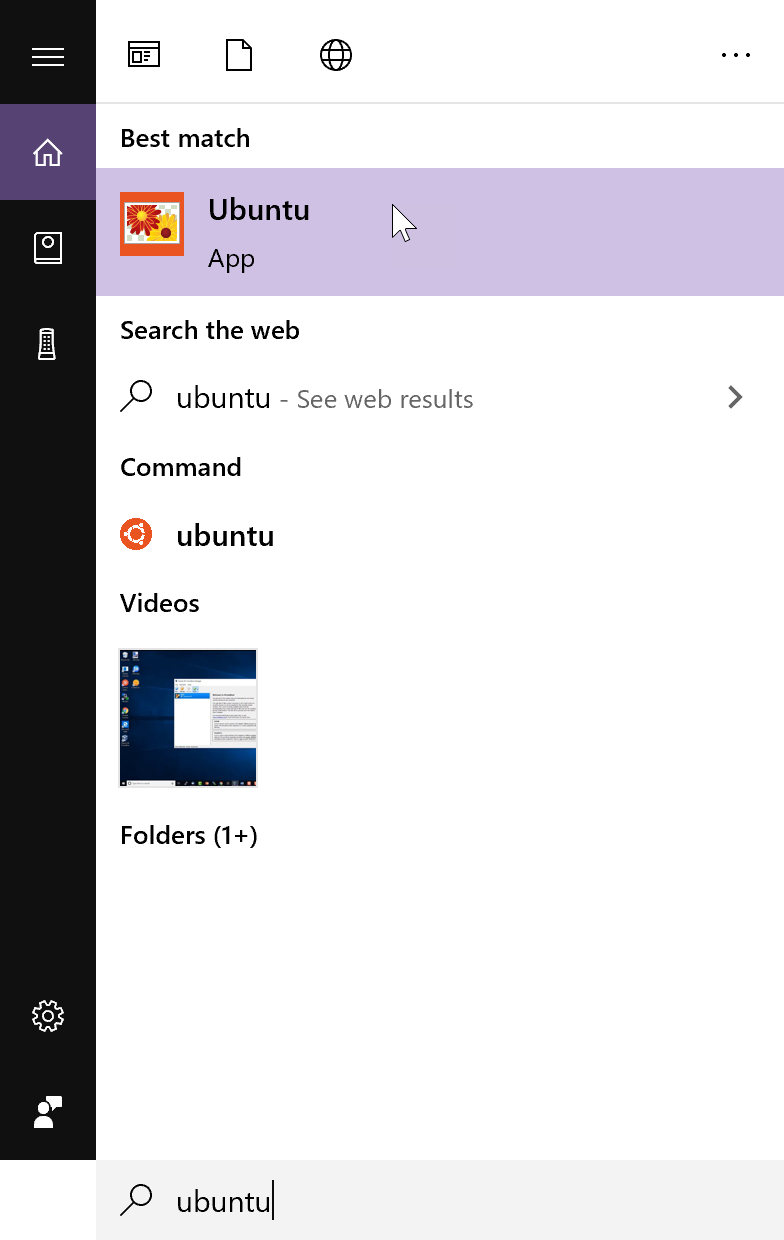
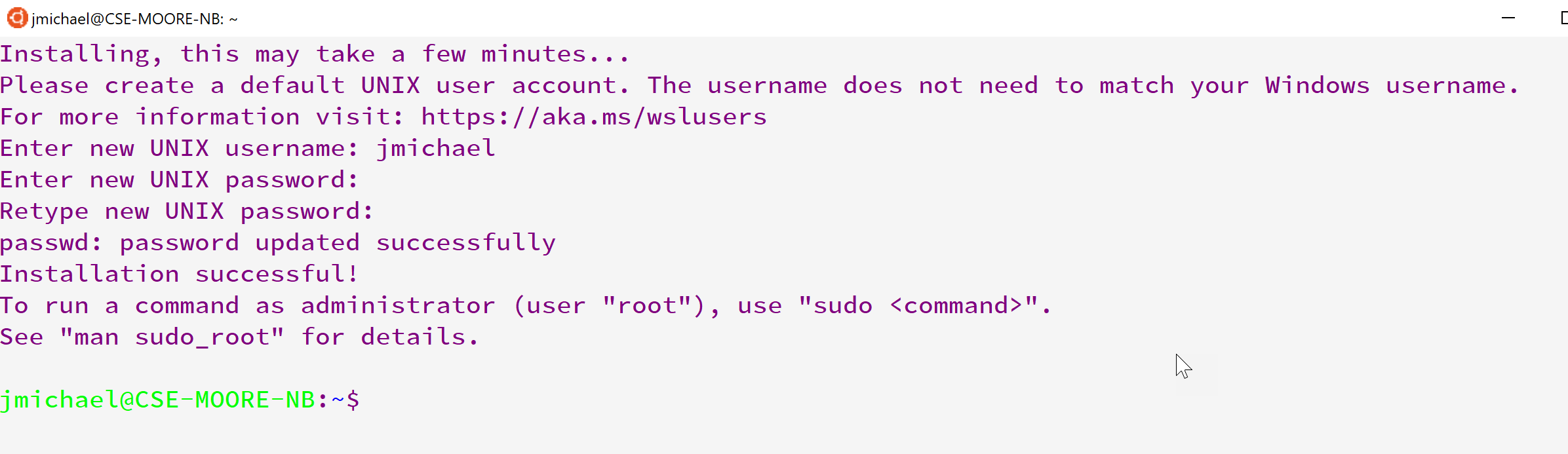
### Installing a Linux environment in Windows

These instructions are based on this help page for Windows 10: <https://docs.microsoft.com/en-us/windows/wsl/install-win10>

1. Install the Windows Subsystem for Linux
   1. Open Powershell as Administrator
      1. Type “powershell” into the search bar
      2. Right-click on “Windows PowerShell”
      3. Select “Run as administrator”  
         
      4. On the command line run:

|  |
| --- |
| dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart |

* + 1. Restart your computer

1. Install your Linux Distribution of Choice (Ubuntu, if you don’t have one yet)  
   Most of you will get it from the windows store. If that doesn’t work for you, then follow the alternate directions provided and/or talk to a TA or instructor.
   1. Open the Microsoft Store
      1. Type “Microsoft Store” in the search bar.  
         
      2. In the store, search for “Ubuntu”  
         
      3. Select “Ubuntu”
      4. From the distro page for the “Canonical Group Unlimited”, select “Get”  
           
         It should download the app after you select how you want to install it.
2. Initialize  
   After installing it, launch it.
   1. You can launch by selecting “Launch” from the Microsoft Store.
   2. You can type Ubuntu into the search bar and select Ubuntu from there.  
      
3. Set up a new Linux user account. You can use any username you want.  
   Make sure the password is something that you can remember since you’ll need it when you do anything which requires super-user privileges, like installing and updating software.  
   Note: You will not see characters being typed when you enter the password. Just know that it is taking whatever you type, so be careful.  
   
4. Update & upgrade your packages  
   In your Ubuntu shell type the following:

|  |
| --- |
| sudo apt update && sudo apt upgrade |

sudo is name of the program which lets you *do* something as the *s*uper-*u*ser. apt is Ubuntu’s package manager, a program that lets you install, update, and remove software. update tells apt to check for updates upgrade tells apt to actually perform the update

Provide the password you used to set up.  
Note: If you have something on your clipboard, you can paste it in a command line by right-clicking.  
Eventually, it will ask if you want to continue. Type ‘Y’ or ‘y’.  
The updates can take some time.  
When it asks about restarting services, choose ‘Yes’.

### Installing a C++ compiler in Windows

These instructions are based on:

<https://linuxize.com/post/how-to-install-gcc-compiler-on-ubuntu-18-04/>

1. In your Ubuntu command line execute

|  |
| --- |
| sudo apt update |

This should not be necessary, but better safe than sorry.

1. Then execute

|  |
| --- |
| sudo apt install build-essential |

Type ‘y’ when it asks if you want to continue.

1. (optional) Execute

|  |
| --- |
| sudo apt-get install manpages-dev |

1. Verify you installed gcc by executing

|  |
| --- |
| gcc --version |

The output should be similar to this:

|  |
| --- |
| gcc (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0 Copyright (C) 2017 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. |

### Files on your Windows computer

You can access your c drive (and probably other drives) through your ubuntu shell. You can list the drives by executing:

|  |
| --- |
| df |

You’ll see /mnt/c. This leads to the C:/ drive on your computer. This is useful for the Unix Tutorial.

### Text Editor Installation in Windows

You will want a text editor with syntax highlighting. Although it is not strictly necessary, it really makes reading code much easier.

You should avoid using the Notepad editor that is built into Windows.

#### Text editor options for Windows:

* [Notepad++](https://notepad-plus-plus.org/)
* [Sublime](https://www.sublimetext.com/)
* [Atom](https://atom.io/)
* [Visual Studio Code](https://code.visualstudio.com/)
  + <https://www.youtube.com/watch?v=DIw02CaEusY>

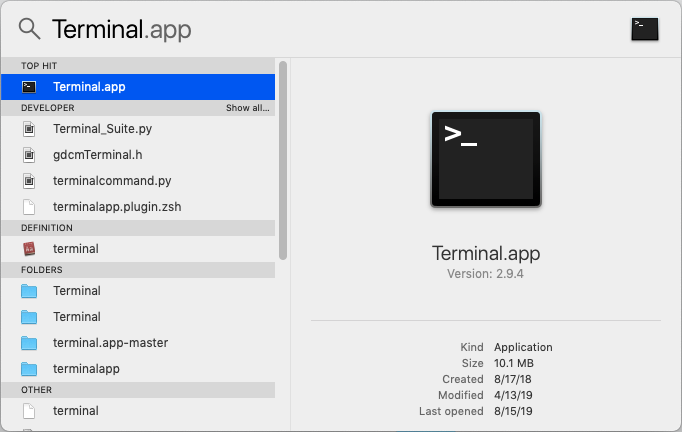
## 

## Mac

For the first week only, you can complete this labwork on your own and do not have to attend a lab session to get credit.

We will be using a common compiling environment for all students in the class, regardless of whether you use macOS, Windows, or a Linux distribution. The way we have decided to do this is to have all students compile in a command-line environment, i.e. the “Terminal” application.

### Installing a C++ Compiler in Mac

1. Launch the “Terminal” application. This can be done by the following sequence of steps:  
   1. Press cmd+space to launch Spotlight Search.   
      
   2. Type Terminal in the Spotlight Search field. A list of results should automatically populate. 
   3. Click on Terminal.app to launch Terminal.   
      
2. In the Terminal application, type xcode-select --install and press enter to install the C++ compiler.  
   

### Text Editor Installation in Mac

You will want a text editor with syntax highlighting. Although it is not strictly necessary, it really makes reading code much easier.

#### Text editor options for Mac

* [Sublime](https://www.sublimetext.com/)
* [Atom](https://atom.io/)
* [Visual Studio Code](https://code.visualstudio.com/)

## 

## Linux

For the first week only, you can complete this labwork on your own and do not have to attend a lab session to get credit.

We will be using a common compiling environment for all students in the class, regardless of whether you use macOS, Windows, or a Linux distribution. The way we have decided to do this is to have all students compile in a command-line environment. You will use your Linux terminal.

### Installing a C++ Compiler in Linux

1. Use your package manager to install gcc-g++
   1. Debian / Ubuntu:  
        
      sudo apt install build-essential
   2. RedHat / CentOS:  
        
      sudo yum install make automake gcc gcc-c++ kernel-devel
   3. Fedora:  
        
      sudo dnf @development-tools  
      sudo dnf group install "C Development Tools and Libraries"
   4. Arch:  
        
      sudo pacman -S base-devel
   5. Other: ask your instructor or TA
      1. Note: We are unlikely to know this outright. But, we will help you figure it out.

### Text Editor Installation in Linux

You will want a text editor with syntax highlighting. Although it is not strictly necessary, it really makes reading code much easier.

#### Text editor options for Linux

* vi / vim
* emacs
* gedit
* nano / pico
* kate
* [Brackets](http://brackets.io/)
* [Light Table](http://lighttable.com/)
* [Sublime](https://www.sublimetext.com/)
* [Geany](https://www.geany.org/)
* [Visual Studio Code](https://code.visualstudio.com/)

# 

# UNIX Tutorial

For the first week only, you can complete this labwork on your own and do not have to attend a lab session to get credit.

Sometimes we need to use terminals to interact with systems that use a UNIX based command line operating system. This lab will give you practice navigating through the UNIX environment on your computer.

**Only the following parts are needed.** You are encouraged to learn more, but do not spend too much time right now learning more than what we need.

* [Introduction to the UNIX Operating System: The Directory Structure](http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html)
* [Tutorial One: all](http://www.ee.surrey.ac.uk/Teaching/Unix/unix1.html)
* [Tutorial Two: 2.1-2.3](http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html)
  + Note: In 2.1 you won’t be able to do the example as written since you aren’t connected to the server that has that file. You can download the file they provide or try finding a file on your own computer and copying that.
* [Tutorial Four: 4.1-4.2](http://www.ee.surrey.ac.uk/Teaching/Unix/unix4.html)

[See the entire tutorial (http://www.ee.surrey.ac.uk/Teaching/Unix/)](http://www.ee.surrey.ac.uk/Teaching/Unix/)

# 

# Write, Compile and Run a C++ Program

You should have already completed the setup for your OS[[1]](#footnote-0) and the Unix Tutorial before doing this part of the labwork.

1. Now that you’ve got the compiler installed, let’s go through the steps of writing a program that simply prints “Hello, World!” to the standard output.
2. You should set up a directory to hold your C++ programs. Also, each program should be in a separate directory. Make a note of where you set this up so you can navigate to it from your unix command line.
3. Open your favorite text editor and create a new file. If you don’t have a text editor installed, go to the “Text Editor Installation (optional)” section at the end of the setup instructions for your OS.
4. Type the following into your new file:

|  |
| --- |
| #include <iostream> I SAID TYPE I DID NOT SAY COPY+PASTE int main() { YOU SHOULD NOT COPY CODE YOU FIND ONLINE  std::cout << "Hello, World!" << std::endl;AND DEFINITELY DON’T PASTE IT  return 0; THANK YOU FOR ATTENDING MY TED TALK } \*MIC DROP\* |

1. Save this source file as hello\_world.cpp in the directory you set up for labwork 1 in the directory where you are saving your C++ programs. I would suggest creating a directory “labwork” to hold labworks and creating a directory “setup\_hello\_world” in it to hold the code for this labwork.
2. Open your unix terminal. (i.e. Ubuntu on Windows, Terminal on Mac, Terminal on Linux)
3. Navigate to the directory holding hello\_world.cpp.

|  |
| --- |
| jmichael@CSE-MOORE-NB:~$ cd /mnt/c/CSCE121/2020c/labwork/setup\_hello\_world jmichael@CSE-MOORE-NB:/mnt/c/CSCE121/2020c/labwork/setup\_hello\_world$ |

Note that prompt -- e.g. jmichael@CSE-MOORE-NB:/mnt/c/CSCE121/2020c/labwork/setup\_hello\_world$ -- will vary depending on the system you are using. From now on, we’ll only show the $ symbol to denote the command prompt.

1. Once in that directory, verify that hello\_world.cpp exists by listing the directory’s contents (ls):

|  |
| --- |
| $ ls hello\_world.cpp $ |

1. Compile and run:
   1. Compile the code with g++:

|  |
| --- |
| $ g++ -std=c++17 -o hello\_world hello\_world.cpp $ |

* 1. Run the code by executing ./hello\_world:

|  |
| --- |
| $ ./hello\_world Hello, World! $ |

# 

# Submission

Complete the “[LW] 1. Setup (Hello World)” activity on Mimir to acknowledge the completion of this lab work.

**A grade of "complete" on this lab work requires a score of 100%.**

1. Operating System [↑](#footnote-ref-0)