CSCI 1300/1310 Introduction to Computer Programming
Recitation 1
Instructor: Knox and Montero-Quesada
Due: at the end of your recitation section (or by Sun Jan 22, 5pm)

Objectives:

1. Setup your Moodle account and test that it works
2. Setup the VM image on your laptop
3. Introduction to Linux

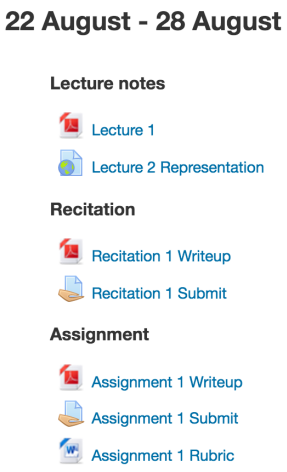
Moodle Accounts:
All students in CSCI 1300 and CSCI 1310 this semester will be accessing course materials through the Computer Science Moodle: http://moodle.cs.colorado.edu. To use Moodle, you log in with your identikey and password, and then enroll in your class.

Once you've logged in, click on the link to your class:

"CSCI 1300 - Knox - CS 1: Starting Computing Spring 2017"
The enrollment key for CSCI 1300 is:
KNOX-CS1300
or
"CSCI 1310 - Montero-Quesada -CS1:Starting Computing (Experienced)"
for CSCI1310 is:
csci1310.

Once you're enrolled in the class, you should see the course materials that have been uploaded so far, organized by week.

Submit to Moodle
To test that your Moodle account is working, download the Profile.txt file found on this week's Moodle section. Edit the file to insert your information. Submit a file to Moodle using the Recitation 1 Submit link.



Virtual Machine Image:

In recitation today, you will be installing the Computer Science Virtual Machine on your laptop, which is designed to provide a consistent development environment for all students. In addition to the VM Image, you need to install VirtualBox to run the VM Image. Your TA has USB drives with both VirtualBox and the VM. The steps to install are:

1. Plug in USB drive
2. Navigate to USB Drive > cu-cs-vm-spring17-x64-v1.0 directory
3. Run (e.g. double click + follow prompts) Virtualbox installer (included in above directory: .dmg for OSX, .exe for WIndows)
4. Import VM image (either double click on .ova file or open Virtualbox -> File -> import -> navigate to flash drive -> .ova)
5. When import completes, eject flash drive and pass on.
6. Launch VM to make sure it works.

If you do not install the VM in recitation, there is information about the VM, and how to install it on your computer can be found on the CU CS Foundation Website: https://foundation.cs.colorado.edu/vm/#intro. The link "Obtaining the VM Image" explains how to download the Image using either BitTorrent (recommended), or through direct download. To download by BitTorrent, you first need to download a BitTorrent client, such as Transmission. Downloading the VM takes a long time and it is recommended that you do this when you will have a reliable Internet connection for several minutes.

In addition to the VM Image, you need to download VirtualBox to run with the VM Image. Instructions for how to do this are found under the link "Installing the VM". Install VirtualBox and then import the VM Image following the instructions on "Installing the VM".

If you encounter issues getting the VM setup on your machine, there are help sessions available. The times and days are listed on the Foundation website: https://foundation.cs.colorado.edu/vm/#intro.

If you don't have a laptop

If you do not have a laptop, the lab machines in ECCS 112 have the same environment as the one installed on the VM. To use these machines, log in with your CU identikey and password. The main difference between the lab machines and your laptop is that the lab doesn't have Dropbox, you will need to store your work on the USB drive.

If you need assistance with the lab machines, go to office hours with your TA or course instructor and they can help you get started.
Note: If you don't get your VM working in recitation, go to VM help hours. The schedule for help sessions can be found here:
https://foundation.cs.colorado.edu/vm/#intro.

What is a VM and how do I use it?

VM stands for Virtual Machine. Essentially, it is a computer running inside of your laptop (or desktop or whatever). The VirtualBox application is what lets you use the Virtual Machine (i.e., it 'hosts' the Virtual Machine Image). Thus, if you wanted to, you could host multiple images using just one VirtualBox application.

A web browser allows you to access what someone else (e.g. Google) has made available. VirtualBox works the same way, allowing you to access the virtual computer (VM) you have. If the idea of a VM is still fuzzy, think of it this way. The VM is like a separate computer running inside of your computer. Thus, your computer and the VM cannot talk to each other. Therefore, you cannot save the work you do in the VM to a folder in *your* personal laptop, as they're treated like two separate computers, just like if you had two desktop computers sitting right next to each other. If you do work on a desktop computer and save it to your My Documents folder, you won't be able to access it from the neighboring computer. This is why we strongly encourage you to use **Dropbox**. It allows you to save your work from the VM that you can later access from your personal (or any) computer. Dropbox is already installed on the VM. If you create a Dropbox account and install Dropbox on your local machine, then you can access your files from both computers.

Lesson: How do I use a VM/Linux?

The VM has an operating system (OS), just like any other computer. The OS is called Linux, and while it might look unfamiliar to you, it is just like Windows or OSX or Android. Also, just like any other computer, there are features of the VM that will be useful in learning programming. The features that we will use primarily this semester are:

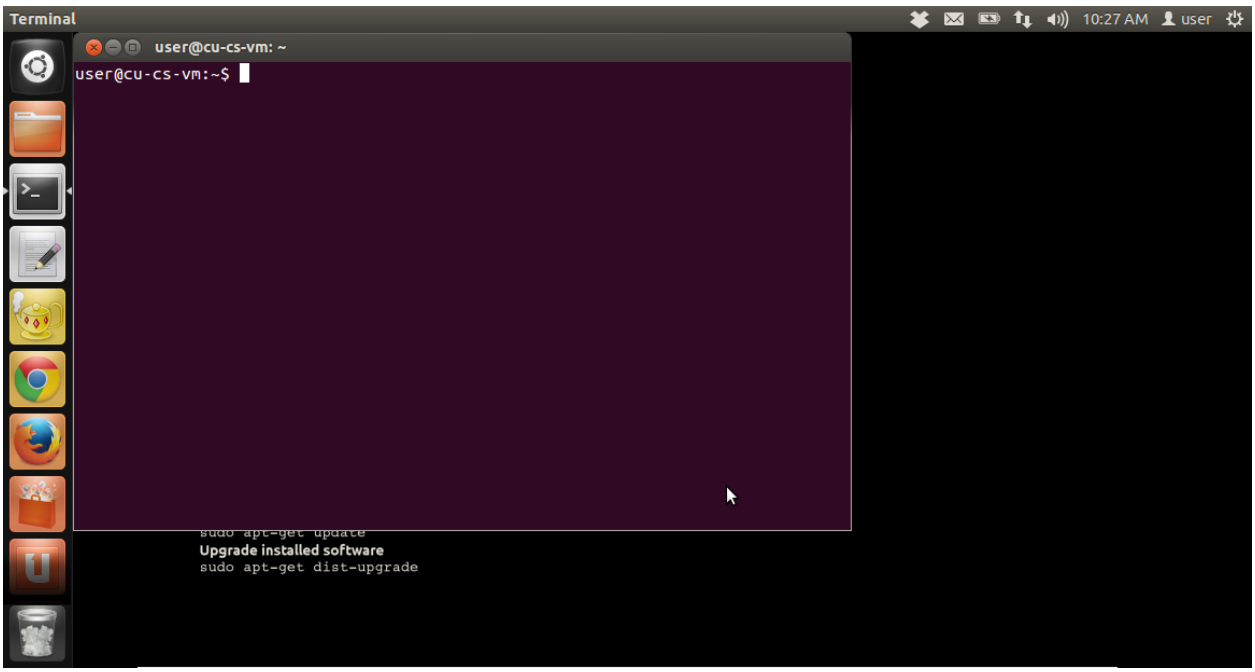
- File System: Directories and files are found here. Directories are shown as orange/manilla folders. Your Dropbox folder can be found here.
- Terminal (black square with some white text): This is the command line interface that allows you to navigate the VM without using the graphical interface. The terminal provides you the ability to do everything on your computer using words (i.e. commands), not clicks.
- gedit (notepad with a pencil): A plain-text editor (e.g. notepad or textedit).
- CodeBlocks An Integrated Development Environment (IDE) for developing source code.
- Geany (lamp with jewels): Another IDE for developing source code.
- Firefox (globe with a fox): A web browser (e.g. internet explorer or safari).

Lesson: What is an IDE?

An Integrated Development Environment (IDE) is a program that allows you to type, edit, and run code (e.g., gedit, CodeBlocks, Geany), much like Microsoft Word allows you to edit text documents. This is where you will be doing your programming for the class.

Lesson: How do I use the Terminal?

Now open the terminal (black square icon with some white text). If you don't have a laptop, partner up with a neighbor so you can step through this with them. Here is what you should see:



- You see <somethingA>@<somethingB> /~ $
  - **SomethingA** = what your name is (the account name/your login name). VMs have a default one.
  - **SomethingB** = what your computer's name is. The VMs should have the same one.
  - **~/** (i.e. everything else before the $) = your current directory (**think of a directory as the folder you're in**).
  - **~** is shorthand for the current user's home directory.
  - **$** = end of prompt for entering a command.

File browsing using the terminal is like using Windows explorer or like clicking on

folders and navigating to different folders on your laptop. In the terminal, instead of clicking on folders we use commands to tell the computer what we want. If we want to go to a folder where we saved our last homework, we can type the commands to navigate to that folder and display its contents.

Try these commands:

- ls = (that's a lowercase "L" not an uppercase "i") stands for *list* and is used to 'list' or show you everything in the current directory.
- cd = stands for *change directory* is just like changing folders. It means, take me to place X. Commonly used as cd <name_of_directory>. **Note that there will always be a space between cd and the name of the directory that you want to navigate to and the name of the directory will **not** include the carrots (side arrows) displayed above.**
    - **Note:** cd takes you places in reference to your current location. It's like going into a folder, and then clicking on a folder within that folder, and then clicking on another folder within that folder. You will always navigate deeper within that folder. To back up, we use "cd .." (explained in a little bit).
    - **Note:** cd (and other commands) is case sensitive. 'dropbox' does not equal Dropbox. Make sure you type in directory names exactly as they are spelled.

Go to Dropbox ($ cd Dropbox). Now you are in your Dropbox directory. If you type 'ls', it will now display all of the files that you have in your dropbox. **Notice how the command line in your terminal now says <Dropbox> to reflect which directory you are in.**

- the "." = shorthand for current location

You can think of a directory structure as being like a family tree. From your current directory, there can be a parent directory, and a parent to that parent, and so on. There can also be other directories that branch off from the current directory that can be considered as children in the tree.

- "cd .." = go to the parent of my current location. It's essentially backing up.

Want to learn more about linux commands? http://community.linuxmint.com/tutorial/view/244 has a list of categorized linux commands.

Running applications from the command line

Many of the applications on the VM can be run from the terminal. Running programs

in the terminal is also called running them from the command line. Typically, installed applications have the ability to be run from any location. You just type the name of the application, and anything else it needs. For example, if you type in gedit in terminal, a gedit window will open up. So, instead of double clicking on the gedit (in the lefthand pane) to open it, you can just type "gedit" into the command line of your terminal and it will open gedit for you!

Fun tips and tricks

- Tab Complete = if you're typing something in the command line that's very long, but unique, you can hit tab when you're partway through and it will try to fill in the rest (kind of like auto complete). If it doesn't, and you tap tab twice, it tells you everything it has as options.
- Command history browsing = if you have typed a command (e.g. gedit myFile.txt) and want to repeat it, just press the up arrow. It will bring up your last executed command. Pressing up again will go to the one before. Pressing down will go forward in time through the list.
- ASCII is a character encoding standard. It includes numerical characters: 0-9, letters A-Z and a-z, punctuation, and blank spaces. These character codes represent text in computers. For example, the ASCII representation of 'D' is 01000100, and the ASCII representation of 'd' is 01100100. Notice the difference? 'D' is different from 'd', therefore, it is important for you to always make sure that you have proper capitalization and spelling when you are typing from the command line.

More Linux Commands

Lesson: Creating Directories from the Command Line

Directories may be created using either the File Browser, or by creating them on the command line. Creating directories on the command line is rather simple. First, open a terminal, and cd to your Dropbox folder

```
$ cd ~/Dropbox
```

Remember, "~" is a shortcut to your home folder, and by default, Dropbox is stored in this location. If you installed Dropbox in a different location, you'll need to change the above path.

To create a new directory, use the command *mkdir*

```
$ mkdir my_dir
```

Spaces are troublesome on the command line, so we'll use underscore. If you open up Dropbox in the File Browser, you should be able to see the *my_dir* folder. Or, you can just use *ls*.

```
$ ls
Getting Started.pdf    my_dir
```

And, you can go ahead and cd to the new directory you created

```
$ cd my_dir
```

Lesson: Copying Files

Open gedit and create a real simple text file that includes your name.

Save this file as *test1.txt* in the *~/Dropbox/my_dir* folder. If you type *ls* in the terminal, you should now see this file listed. If you aren't in the correct folder, *cd* to it.

```
$ ls
test1.txt
```

How about making a copy of this file? To do this, we use the *cp* command (short for copy). To copy this file, use the following command

```
$ cp test1.txt   test2.txt
```

Now if you type ls, you should see two files

```
$ ls
test1.txt                test2.txt
```

The new file should contain the same content as the original file.

Note: You can use paths (absolute or relative) to indicate what file you want to

copy, and where you want to copy it to. So, you can copy the test1.txt file to the Desktop using the following command:

```
$ cp  ~/Dropbox/my_dir/test1.txt  ~/Desktop
```

Since Desktop is a folder, this command copies the file with the original name into Desktop.

Note: Be very careful not to overwrite a file when copying.

Lesson: Moving Files

It's possible to move files around using the command *mv*. Let's say we want *test2.txt* to be in Dropbox, rather than my_dir. We can move it up one directory in the following way

```
$ mv test2.txt ..
```

Remember, two dots indicates the *parent* directory (one level up). Now if we list the files in my_dir:

```
$ ls
test1.txt
```

we only see test1.txt. Moving one folder up and listing the directory shows us the other file we want.

```
$ cd ..
$ ls
Getting Started.pdf    my_dir          test2.txt
```

Let's move it back into the directory we want:

```
$ mv  test2.txt  my_dir
$ cd  my_dir
$ ls
test1.txt         test2.txt
```

Lesson: Zipping and Unzipping Files

An important thing to know for this class is how to zip your assignment submissions. To do this, there's a convenient command called zip. Let's zip up the two test files into a single zip file.

```
$ zip test_files.zip  test1.txt  test2.txt
```

The first argument after zip is the name of the zip file we want to produce. The files listed next are the files that go into this zip file. If you list your directory now, you should see three files. Notice that the original files are still present--zip and unzip don't destroy any files.

```
$ ls
test1.txt        test2.txt                test_files.zip
```

Let's try unzipping this. First, we'll create a second directory in Dropbox and move the zipped file into it.

```
$ cd  ~/Dropbox
$ mkdir  my_dir2
$ mv  my_dir/test_files.zip  my_dir2
$ cd  my_dir2
$ ls
test_files.zip
```

To unzip, just type the command

```
$ unzip  test_files.zip
```

You should see the two test programs in the directory. Note that the original zip file didn't get destroyed.

```
$ ls
test1.txt                test2.txt                test_files.zip
```