Basics of Linux and the vi editor

This handout will cover some of the basics of the linux operating system and the vi editor. We will only cover the essentials, basically only what is needed for you to start writing code. I encourage you all to learn more on your own time. Most linux books also include a section that covers the vi editor. The text that I prefer most is:

Linux in a Nutshell
Author: Ellen Siever
Publisher: O'Reilly
ISBN: 1-56592-585-8

This is a great reference text to have around.

Spend some time on the following webpage:

http://www.ee.surrey.ac.uk/Teaching/Unix/

Some basic linux commands

Note: most linux commands have flags. A flag is an additional instruction (with a dash in front of it).

ls	This lists the contents of the directory that you are currently in.
ls -F	This lists the contents of the directory in such a way that differentiates files from directories
cd *path*	This "changes directory" to the one described by *path*
cd	The cd command alone simply takes you to your home directory
cd ..	This takes you up one directory
cd -	This takes you to the directory that you were just at
pwd	This tells you which directory you are currently in
clear	This clears your screen

rm *filename*	This will remove a file. Be careful. Once you delete a file, it is lost forever.
rm –r *directory*	This removes a directory. It can also be used to remove a file.
rmdir *directory*	This removes a directory that does not contain any files. This is a safe way to remove a directory that you "think" is empty. If its not empty, it will not be deleted.
cp *file1 file2*	This will copy *file1* to a new file named *file2*. Beware, if *file2* already exists, it will be overwritten. If you omit *file2*, the new file will have the same name as the original. You can also used this to copy to different locations. For example, to copy your homework to another directory that you have created (called backup): cp *homework1 backup/*
cp –r *directory path*	This will copy a directory to another place, defined by *path.*
mv *file_or_directory1 file_or_directory2*	This will either move or rename a file or a directory. If *file_or_directory2* is in the same directory as *file_or_directory1*, then it will be renamed. If you designate a path in front of *file_or_directory2* then it will be moved. Examples: Rename a file: mv homework1 homework1_copy2 Move a file to a directory called backup: mv homework1 backup/homework1
mkdir *directory_name*	This will make a directory
man *command*	This is a useful command to learn about other commands. It will provide an

	onscreen manual page for the command of interest. For example: "man ls " will provide you an overview of the ls command, and it will list all of the flags that can be used with it.
Control-C	If you press the control button and the c button at the same time, you will exit whatever program you are currently in.
ssh *username***@***computername*	This will log you into a remote computer. We will use this to log into our server.
scp *filename username@computername:path*	This will copy a file from your current directory to a directory on a remote computer. You will likely use this to copy your homeworks over to the server.

The vi editor

The vi editor is a simple, text editing program that we will use to write source code. When writing code, it is important to use a "text-editor", and not a "word-processor" because the latter inserts all kinds of nasty formatting characters that the compiler will not understand. Vi is tricky to use at first, but once you get the hang of it, you'll find it to be one of the most efficient text editing programs around. The first thing you should realize about vi is that it doesn't work with your mouse, so don't even try.

Here, only the basic vi editor commands are listed. Please see the web for more; there are numerous resources on the internet. Some nice tutorials:

http://www.wikihow.com/Learn-vi

https://wiki.gentoo.org/wiki/Vim/Guide

http://ryanstutorials.net/linuxtutorial/vi.php

[An interactive tutorial (note, I haven't tried this, so I don't know if it is any good)]
http://www.openvim.com/

To create a new text file with vi, type the following:

vi *filename*
To open an existing text file with vi, the command is the same:

vi *preexisting_filename*

Both of these commands will open the text file with the vi editor.

In vi, there are 2 modes, the command mode and the append mode. The command mode is used to move around your document, save it, quit it, etc. The append mode is used to insert, delete, copy and paste, or modify text. You will switch modes quite frequently, so try to mentally keep track of which mode you are in. Some versions of vi will include the words "INSERT" at the bottom of your screen when you are in append mode.

The "escape" key will always move you into command mode. To go back into edit mode (from command mode), either type *i* or *a*.

Navigating in command mode

Commands to Move Around	
h	Move one space to the left
j	Move one line down
k	Move one line up
l	Move one space to the right
You can move across multiple lines too. For example:	
10**j**	This moves you 10 lines down
23**l**	This moves you 23 columns to the right
0 (zero, not the letter)	This moves you to the beginning of the line
$	This moves you to the end of the line
Shift-**G**	This moves you to the end of the document
:1 (this is number 1, not the letter)	This moves you to the front of the document
:*some_number*	This will take you to the *some_number*^{ith} line of the file
Commands to Put you into Append Mode	
a	This puts you into append mode, one space to the right of the cursor.
i	This puts you into append mode, one space to the left of the cursor.
o	This puts you into append mode, on a newly created line just below the cursor.
O (the letter, not the number)	This puts you into append mode, on a newly created line just above the cursor

Copy and Paste Commands	
yy	This will copy the current line
dd	This will cut the current line
Shift-**P**	This will paste the copied or cut line to a new line above the cursor position.
You can copy or cut multiple lines too. For example:	
some_number **yy**	This will copy *some_number* lines.
Editing Text	
x	This will delete the character under the cursor
r *new_character*	This will replace the character beneath the cursor with a new character.
u	This will undo your last edit
Saving and Quitting	
:w	Saves the work, but keeps you in the program
:w!	Sometimes you will get a warning that says you cannot save the file. This often happens if the file is read only (but you have full control over it). If this happens, the exclamation point forces the save.
:q!	Quits without saving and returns to the shell
:wq	Saves the work, then quits.
:wq!	Sometimes you will get a warning that says you cannot save the file. This often happens if the file is read only (but you have full control over it). If this happens, the exclamation point forces the save.
Searching for Text	
/ *some_text_string*	This will search for a text string, starting at the current position and moving downward. Once the first match is found, you can simply type **/** again to find the next one.
? *some_text_string*	Same as **/** but this will search in the upward direction