Bash Commands:

Command	Flags	Description
pwd		Gives **p**resent **w**orking **d**irectory
ls		**Lis**ts regular files in current directory
ls	-a	**Lis**ts **a**ll files in current directory (including hidden files)
ls	-l	**Lis**ts regular files along with properties (**l**ong form)
mkdir		**M**a**k**es a new **dir**ectory of given name (ex. mkdir cs137)
cd		**C**hanges **d**irectory to directory of given name (ex. cd cs137), given directory name must be either in the current directory or one level above (ex. cd ..)
rm		**R**e**m**oves file(s) of given name, can be given a list (ex. rm hi.c hey.c yo.c)
rm	-i	Same as rm but prompts before removing each file
rmdir		**R**e**m**oves given **dir**ectory, must be completely empty (ex. ca137)
cp		**C**o**p**ies contents from first given file into second given file, overwrites contents of second file if it already exists (ex. cp hey.c hello.c)
mv		**M**o**v**es contents from first given file into second given file, overwrite contents of second file if it already exists, deletes first file (ex. mv hey.c hello.c). Or moves given file into given directory (ex. mv file.c directory)
cat		Prints contents of given file to screen (ex. cat hello.c)
man		Prints out **man**ual of given command (ex. man ls)

Other things to know:

- . represents the current directory
- .. represent the directory one level above your current directory
- Pressing tab part way through a command or file name has the computer guess what you are trying to type (useful for long file names)
- Pressing the up arrow scrolls through previous commands (useful for long commands like “ssh [username]@linux.student.cs.uwaterloo.ca”)
- To create a new file type the name of your text editor then the name of the new file (ex. vim hello.c , emacs hello.c , nano hello.c)

Note: once you delete or overwrite a file you **cannot** get it back; it is **not** like a “trash can” on your desktop