Augusta Manchester
HW 1 - Bash

1. Bash is a type of shell; it's an improved way of using a shell. A shell is a way to interact with a computer through using command lines.

2. Your home directory is the workspace where files and information is stored that the computer will use as a base.

```
(base) Augustas-MacBook-Air:~ augustamanchester$ echo $HOME
/Users/augustamanchester
(base) Augustas-MacBook-Air:~ augustamanchester$ ls
Applications            Public                  classwork5.Rhtml
Desktop                 Temp                    data412classwork2.R
Documents               Untitled.R              hw4.R
Downloads               Untitled5.R             hw4.html
Library                 anaconda3               stat.302.2
Movies                  catfish.R               testdir
Music                   classwork2.R            unt
Pictures                classwork2.R.html
(base) Augustas-MacBook-Air:~ augustamanchester$
```

3. The command cd../../ brings you up two directory levels, but since I was already in my home directory, with no other directories to move towards, nothing happened. If I were in a working project directory it would be different. The command pwd means print working directory, which has just been set to my home directory. When you just do cd, it brings you to your home directory and pwd again displays this.

```
(base) Augustas-MacBook-Air:~ augustamanchester$ `cd../../`
bash: cd../../: No such file or directory
(base) Augustas-MacBook-Air:~ augustamanchester$ pwd
/Users/augustamanchester
(base) Augustas-MacBook-Air:~ augustamanchester$ cd
(base) Augustas-MacBook-Air:~ augustamanchester$ pwd
/Users/augustamanchester
(base) Augustas-MacBook-Air:~ augustamanchester$
```

4. To see the manual of ls, I did man ls. It says the a flag stands to include directory entries starting with a '.', and the l flag for following links to targets and then it lists the file or

directory rather than the link itself.

```
  -A      Include directory entries whose names begin with a dot (‘.’) except
          for . and ..  Automatically set for the super-user unless -I is
          specified.
```

```
  -L      Follow all symbolic links to final target and list the file or
          directory the link references rather than the link itself.  This
          option cancels the -P option.
```

5. a/b:

```
(base) Augustas-MacBook-Air:~ augustamanchester$ mkdir temp_bash
(base) Augustas-MacBook-Air:~ augustamanchester$ ls
Applications            Public                  classwork5.Rhtml
Desktop                 Temp                    data412classwork2.R
Documents               Untitled.R              hw4.R
Downloads               Untitled5.R             hw4.html
Library                 anaconda3               stat.302.2
Movies                  catfish.R               temp_bash
Music                   classwork2.R            testdir
Pictures                classwork2.R.html       unt
(base) Augustas-MacBook-Air:~ augustamanchester$ cd ~/temp_bash
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ touch myfile.txt
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ ls
myfile.txt
```

C:

```
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ stat -x myfile.txt
  File: "myfile.txt"
  Size: 0            FileType: Regular File
  Mode: (0644/-rw-r--r--)         Uid: (  501/augustamanchester)  Gid: (   20/   s
taff)
Device: 1,7   Inode: 28400075    Links: 1
Access: Sun Sep 15 14:19:38 2024
Modify: Sun Sep 15 14:19:38 2024
Change: Sun Sep 15 14:19:38 2024
 Birth: Sun Sep 15 14:19:38 2024
(base) Augustas-MacBook-Air:temp_bash augustamanchester$
```

The size is the size of the file, file type the type, uid the username, gid the group ID, access when it was accessed, modify when the file was last modified, change when the larger data about the file was changed, and birth when it was created. On mac, the block and io block isn’t shown by default, but those denote the number of physical block discs given to the file and the size of the blocks.

6.

a.
```
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ echo "This is my first li
ne." >> myfile.txt
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ echo "This is my second l
ine." >> myfile.txt
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ cat myfile.txt
This is my first line.
This is my second line.
(base) Augustas-MacBook-Air:temp_bash augustamanchester$
```

b.
```
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ cp myfile.txt copy_myfile
.txt
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ ls
copy_myfile.txt myfile.txt
```

c.
```
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ echo "This is a new line"
> copy_myfile.txt
(base) Augustas-MacBook-Air:temp_bash augustamanchester$ cat copy_myfile.txt
This is a new line
```

d. When you use > instead of >>, it replaces the entire contents of the file, rather than adding to it. This is why the previous lines disappeared.