

echo “Hello CEG2350”

Aiden Cox - CEG 2350 Lab Lead
Austin Kellough - CEG 2350 Lab Assistant
Quote of the week:
“It worked on my machine”

Check in! How was Lab03?

Questions over anything up until now?

Scripting related, IO Redirection, Git
commands?



Beginning Lab04

Lab Instructions: <https://github.com/pattonsgirl/CEG2350/tree/main/Labs/Lab04>

Lab Template:

<https://raw.githubusercontent.com/pattonsgirl/CEG2350/refs/heads/main/Labs/Lab04/LabTemplate.md>

Questions from things covered in class?



Finance Tracker Script - Part 1

Deleting Strings

Switch Case

```
case EXPRESSION in
    PATTERN_1)
        STATEMENTS
    ;;
    PATTERN_2)
        STATEMENTS
    ;;
    PATTERN_N)
        STATEMENTS
    ;;
    *)
        STATEMENTS
    ;;
esac
```

The command below will delete all the lines that contain the word 'string':

```
sed -i '/string/d' filename
```

Sample Using Global Variables and function call

Functions in Bash

```
#!/bin/bash

hello_world () {
    echo 'hello, world'
}

hello_world
```

```
#!/bin/bash

string1=$1
string2=$2
taskLocation="$HOME/.tasks"
add_task () {
    if [[ -z "$string2" ]]; then
        echo "Please input what task you would like to add"
        read task
        echo $task >> $taskLocation
        echo "The task has been added"
    else
        echo $string2 >> $taskLocation
        echo "The task has been added"
    fi
}
```

Sample Runs Part 2

Part 2 - User Guide & Sample Runs

User Guide

Fill in with your user guide here - make sure you are focusing on good *markdown* formatting

Sample runs

Example of using `add` record

```
show blocks  
of copy / pasted runs  
of your script
```



Example of using `remove` record

```
show blocks  
of copy / pasted runs  
of your script
```



Example of using `view` records

```
show blocks  
of copy / pasted runs  
of your script
```



Example of using `clear` records

```
show blocks  
of copy / pasted runs  
of your script
```



PATH for all - Part 3

```
~ > echo $PATH                                01:54:28 PM  
/Library/Frameworks/Python.framework/Versions/3.13/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Crypt  
exes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin  
:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/b  
ootstrap/usr/appleinternal/bin:/Library/Apple/usr/bin:/Applications/VMware Fusion.app/Contents/Public:/Users/joshu/.loca  
l/bin
```

Talked about PATH before in class?

```
~/2350Demos > ll  
.rwxr-x--x@ 119 aidencox 25 Jun 14:27 hardLink  
.rwxr-x--x@ 119 aidencox 25 Jun 14:27 linkDemo  
lrwxr-xr-x@   - aidencox 25 Jun 14:26 softLink -> linkDemo
```

Links! Chart on next slide for differences

Link Differences

Feature	Soft Link (Symbolic Link)	Hard Link
Type	Pointer to the file path	Pointer to the file's data (inode)
Broken if original is deleted	Yes, link breaks	No, data remains accessible
Identifies file by	File name/path	Inode (data on disk)
Can link to directory	Usually (with <code>-s</code>)	Not allowed (normally)
Acts like a shortcut	Yes	No, it's another name for the same file

echo “Have a Good Weekend!”

Don’t hesitate to reach out and ask questions!
Quote of the week:
“It worked on my machine”