

Good Afternoon

CEG2350

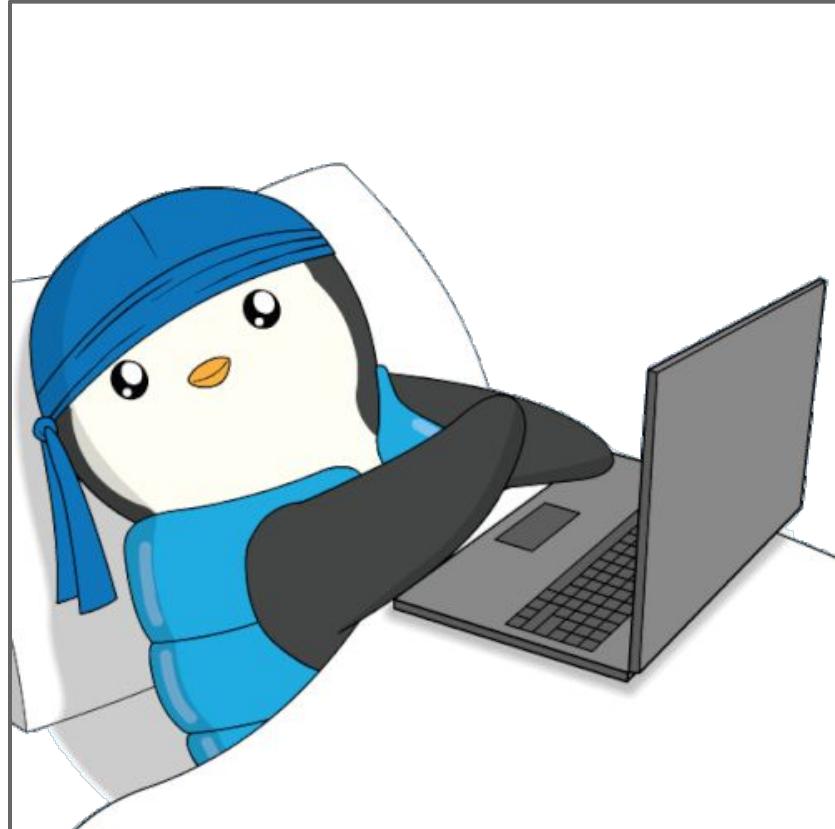
Aiden Cox - CEG 2350 Lab Lead
Austin Kellough - CEG 2350 Lab Assistant
Quote of the week:
“better late than never, but never late is better”

How was Lab05?



Grep, sed, awk?

Midterm review??



Beginning Lab06

Lab Instructions: <https://pattonsgirl.github.io/CEG2350/Labs/Lab06/Instructions.html>

Lab Template:

<https://raw.githubusercontent.com/pattonsgirl/CEG2350/refs/heads/main/docs/Labs/Lab06/LabTemplate.md>

How's life?

Other classes? Calculus?

.bash_aliases - Part 1

Part 1 - bash aliases

Craft an `alias` with a name of your choice that pipes a `fortune` to a `cowsay` cow of your choice (`-f` flag)

Craft an `alias` with a name of your choice to something “fun”. Checkout these as suggestions. You may need to install these programs first with `apt`.

- curl wttr.in
- have cat use lolcat instead
- asciiaquarium

Add the aliases to a file named `.bash_aliases` in your user's home directory.

Read through the `.bashrc` file in your user's home directory for a section that is relevant to “enabling” usage of the `.bash_aliases` file. Copy and paste the snippet into your lab template and describe what it does.

Confirm that your aliases continuously work as long as the conditions are met.

Copy your `.bash_aliases` file to your GitHub repository in your `Lab06` folder.

You may delete the `.bash_aliases` file in your user's home directory as needed through this lab.

```
41
42 # Code preview
43 alias pre="bat --color=always --style=numbers --line-range=:500 {}"
44 █
45 # Better cd zoxide with fzf
46 alias cd="z"
47 alias fd="zi"
48
49 # Git shortcuts
50 alias g="git"
51 alias gaa="git add ."
52 alias gb="git branch"
53 alias gcb="git checkout -b"
54 alias gm="git commit -m"
55 alias gp="git push"
56
57 neofetch
```



Some of my aliases in my personal machine!

Dotfile Installer Script - Part 2

Part 2 - Dotfile Installer Script

Your tasking is to write a script that sets up your user account to use your `.bash_aliases`. To accomplish this, you will create an “installation” script that accomplishes these taskings.

WARNING

You must have a **minimum of 3 (THREE) COMMITS** made for this script. Your commit messages must describe what's currently working / tested. When (after which steps / testing) you make a commit is your choice. Your commits must demonstrate an incremental building process - a bulk dump and then making a spelling change or two will not get you out of the penalty in the rubric.

END WARNING

1. Create a script named `dotinstall` in your `Lab06` folder.
2. Create a function called `printHelp`. `printHelp` should output the following:

```
Usage: dotinstall [-OPTION] [ARG]
-s setup - attempts to create a symbolic link .bash_aliases file to user's
-d disconnect - removes symbolic link
-a append - adds a new alias to .bash_aliases file
-r remove - removes an alias from .bash_aliases file
```

1. Use `getopts` to read in options and save arguments that correlate with options. `getopts` should support the following options in the `OPTSTRING`

- `-h`
 - call `printHelp` function and exit script
- `-s`
 - calls function to attempt to create a symbolic link between the `.bash_aliases` in the `Lab06` folder to a `.bash_aliases` in the user's home directory
 - if `.bash_aliases` already exists, prompt the user to confirm the most appropriate action before overwriting the existing file
- `-d`
 - calls function to remove symbolic link from user's home directory
- `-a`
 - is followed by an argument - a string containing an alias
 - calls a function to append the given alias to `.bash_aliases`
- `-r`
 - is followed by an argument - a string containing an alias name
 - calls a function to search for and remove the alias of the name from `.bash_aliases`
 - user should be notified if no matching alias was found
- `\?`
 - use as case when option is not part of `OPTSTRING`
 - call `printHelp` and exit script

Getopts examples!

```
1 #!/bin/bash
2
3 while getopts "ab:c" opt; do
4   case $opt in
5     a)
6       echo "Option -a detected"
7       ;;
8     b)
9       echo "Option -b detected with argument: $OPTARG"
10    ;;
11    c)
12      echo "Option -c detected"
13      ;;
14    \?)
15      echo "Invalid option: -$OPTARG"
16      ;;
17    esac
18 done
```

```
./example.sh -a
```

```
./example.sh -b world
```

```
./example.sh -c
```

Option -a detected

Option -b detected with argument: world

Option -c detected



Retrospective and Usage Guide - Part 3 & 4

Part 3 - Retrospective

1. How would you explain getopt to a friend?
2. What did you get stuck on while working on this script? How did you overcome it?
3. What feature would you add to this script?

Part 4 - Usage Guide

Fill out the Usage Guide section in your lab template. It should contain a minimum of the following:

1. Information on your script's purpose
2. Information on what is in `.bash_aliases`
3. Examples of script usage and output

Have a Good Weekend!

Don't hesitate to reach out and ask questions!

Quote of the week:

“Better late than never, but never late is better”