Project 1.

3. There are two stdouts representing the two samples we created and one stderr representing a sample ls can't find

4. There is only a stderr stating it cannot find sample3, the stdout went to the file because redirect uses stdout by default

5. The contents of the file are the stdout of the previous command because that's where we told it to go via the redirect

6. There is only stdout on the screen because we redirected the stderr to the file

7. The contents of the file are the stderr of the previous command. The contents were not retained because a single arrow symbol represents that we want to overwrite the previous file completely

8. There is not stdout or stderr because we redirected both into their own files

9. The contents are the stdout of the previous command because we redirected it to this file

10. The contents are the stderr of the previous command because we redirected it to file2

11. Both the stdout and stderr were redirected to file

12. The contents are both the stdout and stderr of the previous command because we redirected both to this file

13. Yes there is stdout on the screen and no stderr because we redirected it to file2

14. The contents of file2 are stderr because we didn't declare 2 before we tried to redirect stdout to it

16. The contents of the file are the current time and date because we ran the date command and redirected it into the file

18. The contents of the file are the stdout of the former date command and the latter because we appended it using two redirects instead of overwriting it with one

19. The error was 'extra operand' which means tr is being given too many things to operate on

20. The command replaced all the o's with capital O's from the hosts file

Project 2.

3. The list is smaller because we don't need to export all the programming needed for the current bash environment, just the basics

4. It changed the prompt to 'Hello There:' instead of my user, machine name, and pwd because PS1 controls what the prompt says

5. The prompt reset back to defualt. We can ensure the prompt persists by editing the user's bash\_profile file

6. The script will first tell the user to provide a y or n if they want a hello prompt, then it will wait for an input, and if that input is equal to a y then it will change the prompt to 'Hello There: '

7. My prompt changed to 'Hello There: ' because my 'y' satisfied the if condition and changed the PS1 variable

10. The variable is listed because set contains all the variables made in the current shell session

11. The variable is not listed because it's not set to be available to other programs, which requires modifying the environment variables

12. The variable is listed now because we exported it to the environment variables

14. Nothing is listed under the variable because environments do not persist between sessions

17. The variable is set to 'My sample variable' because set it to export everytime we start a session

18. Aliases for types of 'grep' and for 'ls' are already present.

19. This command creates and alias called 'asample' for displaying the contents of the hosts file and the files in the users home directory

20. It displayed the contents of the /etc/hosts file and the files in the users home directory. I could add the alias command to the .bash\_profile file to ensure it persists between sessions

Project 3.

4. The myscript file has rw permissions for owner and read permissions for everyone else. The script executed. The \t created a tab space in the text. \a rings a bell.

5. I got a permission denied when trying to run the script because the script doesn't have execute permissions

6. The script executed because we gave it permission to execute

Project 4.

5. Yes the entry was present because the script read the variables we input and appended it to the database file

8. Yes the name is there

9. Yes the name is there because grep found the name in the file

11. We recieved an error that said to enter the letter a or s because we did not enter the letter a or s

12. It did not matter whether we entered a or A because the script accounts for both cases

13. (see 12)

15. The menu appears again because the script is contained in a while loop and won't escape until an exit command is given such as via the q

18. The stuff directory was created successfully because there was nothing preventing it from doing so. It was successfully copied because there was nothing preventing it from doing so.

19. The stuff directory wasn't created because it was alreay created before. The file was not successfully copied because the file doesn't exist to be copied.

22. The permissions were changed for all 3 scripts correctly