

Assignment 3

Tasks:

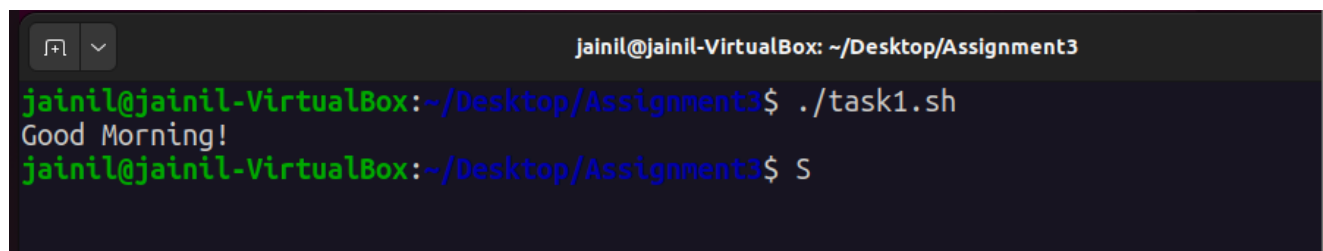
1: Basic Functions

Objective: Write a shell script that uses a function to print a greeting.

```
#!/bin/bash

greet() {
    echo "Good Morning!"
}

greet
```

A terminal window titled 'jainil@jainil-VirtualBox: ~/Desktop/Assignment3'. The prompt is 'jainil@jainil-VirtualBox:~/Desktop/Assignment3\$'. The user enters './task1.sh', and the output is 'Good Morning!'. The prompt then changes to 'jainil@jainil-VirtualBox:~/Desktop/Assignment3\$' with a space before the '\$' symbol.

```
jainil@jainil-VirtualBox: ~/Desktop/Assignment3
jainil@jainil-VirtualBox:~/Desktop/Assignment3$ ./task1.sh
Good Morning!
jainil@jainil-VirtualBox:~/Desktop/Assignment3$ S
```

2: Debugging Scripts

Objective: Write a buggy shell script and use debugging to identify and fix the issue.

```
#!/bin/bash

add() {
    local a=$1
    echo $(expr 1 + $a)
}

echo $(add 'a')
```

```
bash -x ./task2.sh
```

```
jainil@jainil-VirtualBox: ~/Desktop/Assignment3
jainil@jainil-VirtualBox:~/Desktop/Assignment3$ bash -x ./task2.sh
++ add a
++ local a=a
+++ expr 1 + a
expr: non-integer argument
++ echo
+ echo

jainil@jainil-VirtualBox:~/Desktop/Assignment3$
```

fixed script:

```
#!/bin/bash

add() {
    local a=$1
    echo $(expr 1 + $a)
}

echo $(add 15)
```

3: Using Functions for Repetitive Tasks

Objective: Write a shell script that uses a function to create multiple directories.

```
#!/bin/bash

createDir() {
    for i in "$@"
    do
        mkdir -p "$i"
        echo "directory $i created"
    done
}

read dirs
createDir $dirs
```

```
jainil@jainil-VirtualBox:~/Desktop/Assignment3$ ./task3.sh
dir1 dir2 dir3 dir4 dir5
directory dir1 created
directory dir2 created
directory dir3 created
directory dir4 created
directory dir5 created
jainil@jainil-VirtualBox:~/Desktop/Assignment3$
```

```
jainil@jainil-VirtualBox:~/Desktop/Assignment3$ ls
dir1 dir2 dir3 dir4 dir5 dirs task1.sh task2.sh task3.sh
jainil@jainil-VirtualBox:~/Desktop/Assignment3$
```

4: Script with All Learned Concepts

Objective: Write a shell script that uses variables, conditionals, loops, and functions to interact with the file system.

Script:

```
#!/bin/bash

createDirs() {
    for i in "$@"
    do
        mkdir -p "$i"
        echo "directory $i created"
    done
}

createFiles() {
    for i in "$@"
    do
        touch "$i"
        echo "file $i created"
    done
}

removeDirs() {
    for i in "$@"
    do
        rm -r "$i"
        echo "$i removed"
    done
}

readFile() {
    cat $1
}

while true
do
    echo "Select one of the following:"
    echo "1. Create files"
    echo "2. Create Dirs"
    echo "3. Read file"
    echo "4. Remove files/dirs"
    echo "5. Quit"

    read -p "Enter your choice: " choice
    case $choice in
        1)
            read -p "Enter file names to create: " fileNames
            createFiles $fileNames
            ;;
        2)
    
```

```

        read -p "Enter dir names to create: " dirNames
        createDirs $dirNames
        ;;
3)
        read -p "Enter a file name to read: " fileNameToRead
        readFile $fileNameToRead
        ;;
4)
        read -p "Enter a files/dirs to remove: " fileDirNames
        removeFiles fileDirNames
        ;;
5)
        echo "Quitting..."
        break
        ;;
*)
        echo "Invalid choice"
        ;;
esac
done

```

```

jainil@jainil-VirtualBox:~/Desktop/Assignment3$ ./task4.sh
Select one of the following:
1. Create files
2. Create Dirs
3. Read file
4. Remove files/dirs
5. Quit
Enter your choice:

```