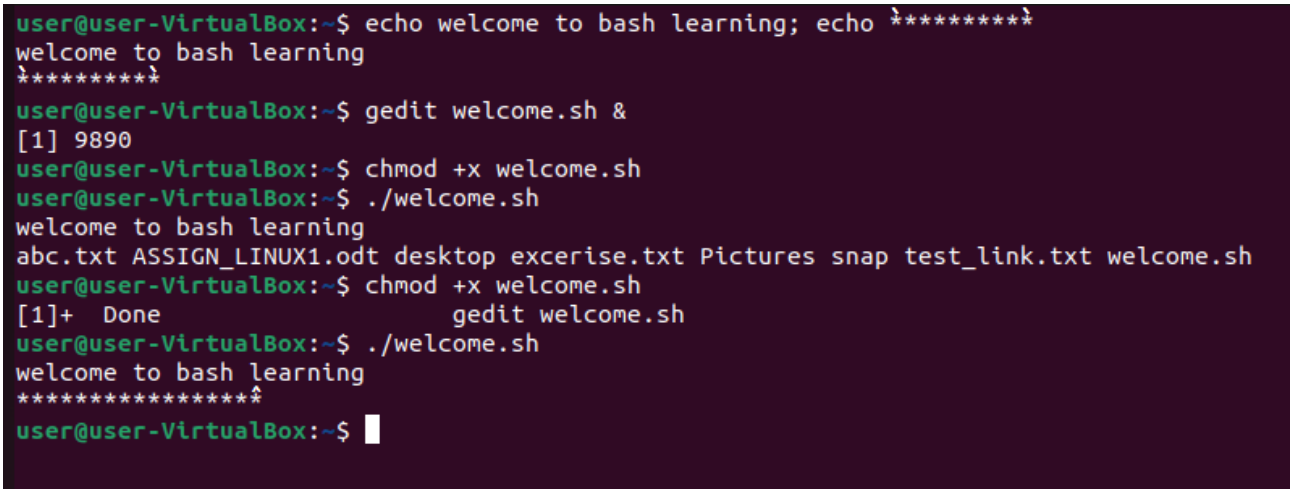
MODULE 3- BASH TRAINING

Introduction to Bash

1) Write a simple Bash shell script to display the message "Welcome to Bash learning" and "***************" on separate lines.



Basics of Shell Scripting

1) Write a simple Bash program to get the following system variables:
   a. pwd
   b. logname

```
user@user-VirtualBox:~$ gedit systemvariable.sh &
[1] 5293
user@user-VirtualBox:~$ chmod +x systemvariable.sh
[1]+  Done                    gedit systemvariable.sh
user@user-VirtualBox:~$ ./systemvariable.sh
pwd : /home/user
logname : user
user@user-VirtualBox:~$
```

Open | systemvariable.sh ~/ | Save

```
#!/bin/bash
pwd=$(pwd)
echo "pwd : $pwd"
log_name=$(logname)
echo "logname : $log_name"
```

2) Write a simple Bash program:
   a. To ask username from user
   b. Exit the program, if user does not enter anything within 10 seconds

Hint: read -t 10 -p

```
user@user-VirtualBox:~$ gedit usernmae.sh &
[1] 5493
user@user-VirtualBox:~$ chmod +x usernmae.sh
user@user-VirtualBox:~$ ./usernmae.sh
Enter the username :ben
./usernmae.sh: line 3: [-z: command not found

 Hello ben
```

```
#!/bin/bash
read -t 10 -p "Enter the username :" username
if [-z "$username" ]; then
echo "Exiting the program"
exit 1
fi
echo " Hello $username"
```

Command Line arguments and Quoting

1) Write a bash program for addition using command line arguments.

```
user@user-VirtualBox:~$ ./addition.sh 2 3
./addition.sh: line 2: [2: command not found
The sum is : 5
user@user-VirtualBox:~$
```

*add.sh | addition.sh

```
1 #!/bin/bash
2 if ["$#" -ne 2]; then
3  echo "Usage:$0 <num1><num2>"
4  exit 1
5  fi
6
7  sum=$(($1+$2))
8  echo "The sum is : $sum "
9
```

Globbing and Export statement

1) Write a Bash script to do all operations discussed under Globbing

```
user@user-VirtualBox:~/desktop$ ls *.sh
sample1.sh  sample3.sh  sample5.sh  test2.sh
sample2.sh  sample4.sh  test1.sh    test3.sh
user@user-VirtualBox:~/desktop$ ls s*.sh
sample1.sh  sample2.sh  sample3.sh  sample4.sh  sample5.sh
```

```
user@user-VirtualBox:~/desktop$ ls [s-u]*.sh
sample1.sh  sample3.sh  sample5.sh  test2.sh
sample2.sh  sample4.sh  test1.sh    test3.sh
user@user-VirtualBox:~/desktop$ ls [^s-u]*.sh
ls: cannot access '[^s-u]*.sh': No such file or directory
user@user-VirtualBox:~/desktop$ ls [^q-s]*.sh
test1.sh  test2.sh  test3.sh
user@user-VirtualBox:~/desktop$ ls [Tt]*.sh
test1.sh  test2.sh  test3.sh
user@user-VirtualBox:~/desktop$
```

Array Operations in BASH

1) Declare an Array names of length 7 and find
   a. The total number of elements
   b. Print all the elements
   c. Print the 5th element

```
user@user-VirtualBox:~/desktop$ ./array.sh
Total number of element in the array: 5

Total element in the array: A B C D E

The fifth element in the array: E

user@user-VirtualBox:~/desktop$
```

```
1 #!/bin/bash
2 declare -a students=('A' 'B' 'C' 'D' 'E')
3 echo -e "Total number of element in the array: ${#students[@]} \n"
4 echo -e "Total element in the array: ${students[@]}\n"
5 echo -e "The fifth element in the array: ${students[4]} \n"
6
```

More on Arrays

1) Declare an Array names2 of length 7 and perform following operations-
   a. Extract three elements starting from index two.
   b. Replace third element with 'Debian' and display.
   c. Append any new name at the end of Array.

```
user@user-VirtualBox:~/desktop$ ./name2.sh
Original elements in the array is A B C D E F G

extracted elements starting from index 2 is C D E

Replacing thrid element with the string debian A B Debian D E F G

Appending new name at th end A B Debian D E F G H

user@user-VirtualBox:~/desktop$
```

```
1 declare -a name2=('A' 'B' 'C' 'D' 'E' 'F' 'G')
2 echo -e "Original elements in the array is ${name2[@]} \n"
3 echo -e "extracted elements starting from index 2 is ${name2[@]:2:3} \n"
4 name2[2]='Debian'
5 echo -e "Replacing thrid element with the string debian ${name2[@]} \n"
6 name2=("${name2[@]}" "H")
7 echo -e "Appending new name at th end ${name2[@]} \n"
```

Conditional execution

1) Write a script which will take your name as an input.
2) It should check this name with your system's username.
3) If the username matches, it should greet you by displaying "Hello".
4) Else, it should display "Try again"

HINT: Your system's username is stored in a variable $USER

```
user@user-VirtualBox:~/desktop$ ./ifstatement.sh
Enter User name : Qwerty

 Try Again
user@user-VirtualBox:~/desktop$ ./ifstatement.sh
Enter User name : user123

 Hello
user@user-VirtualBox:~/desktop$
```

```
#!/bin/bash

USER="user123"
read -p "Enter User name : " myuser
if [ "$myuser" == "$USER" ] ;
then
        echo -e "\n Hello"
else
        echo -e "\n Try Again"
fi
```

Nested and multilevel if elseif statements

1) Write a program to output different messages when number is:
   a. Greater than 3
   b. Lesser than 3
   c. Or equal to 3
   d. Or when the user input is empty

```
user@user-VirtualBox:~/desktop$ gedit three.sh &
[2] 16870
user@user-VirtualBox:~/desktop$ chmod +x three.sh
[2]+  Done                    gedit three.sh
user@user-VirtualBox:~/desktop$ ./three.sh
Enter a number :6
Greater than 3
user@user-VirtualBox:~/desktop$ ./three.sh
Enter a number :3
        Equal to 3
user@user-VirtualBox:~/desktop$ ./three.sh
Enter a number :1
Lesser than 3
user@user-VirtualBox:~/desktop$
```

Open

three.sh
~/desktop

```
1 #!/bin/bash
2 read -p "Enter a number :" mynum
3 if [ -z "$mynum" ]; then
4         echo "User input is empty"
5 elif [ "$mynum" -gt 3 ] ; then
6         echo "Greater than 3"
7 elif [ "$mynum" -lt 3 ] ; then
8         echo "Lesser than 3"
9 else
10        echo "  Equal to 3"
11 fi
```