Thumati Pavan Venkata Narendra Kumar

EMP-ID-289219

BASH-Assignment-1

1.Arithmetic Operators:

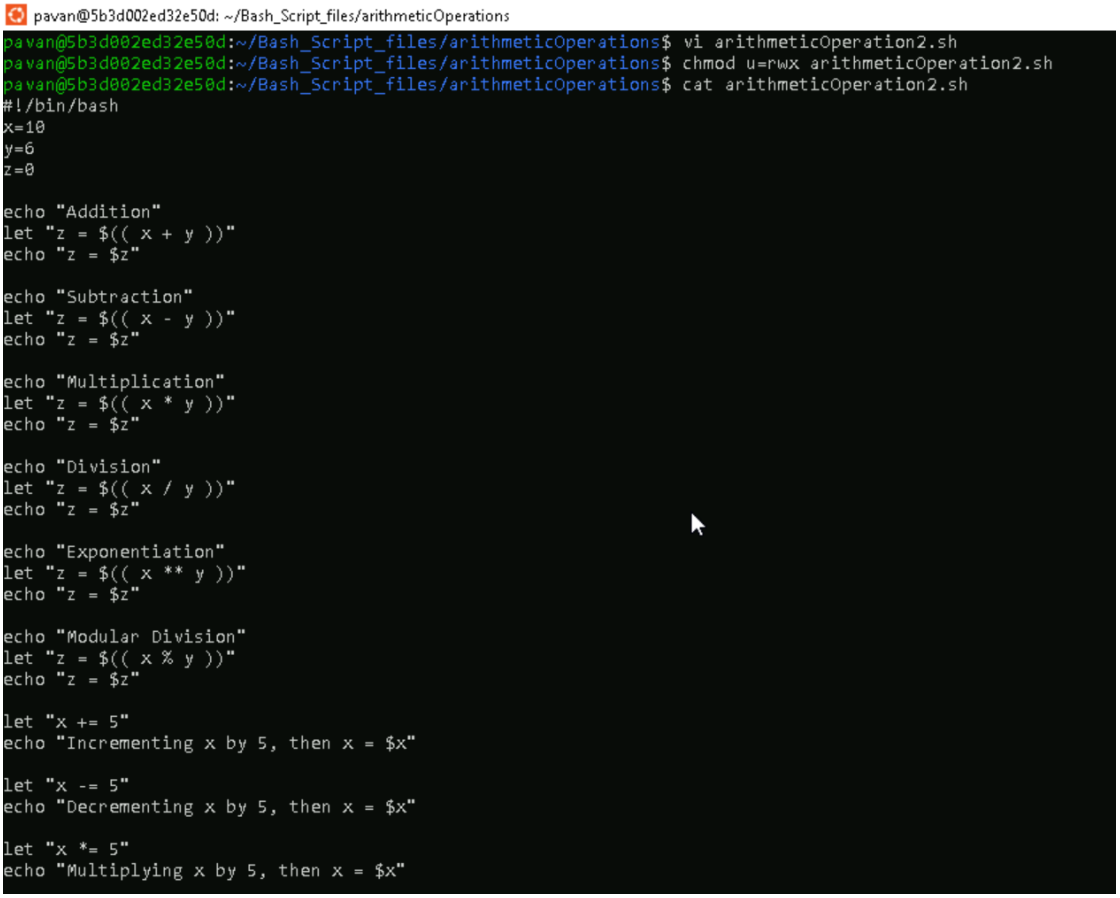
1. To use of double parentheses for arithmetic operations in a Bash shell script.

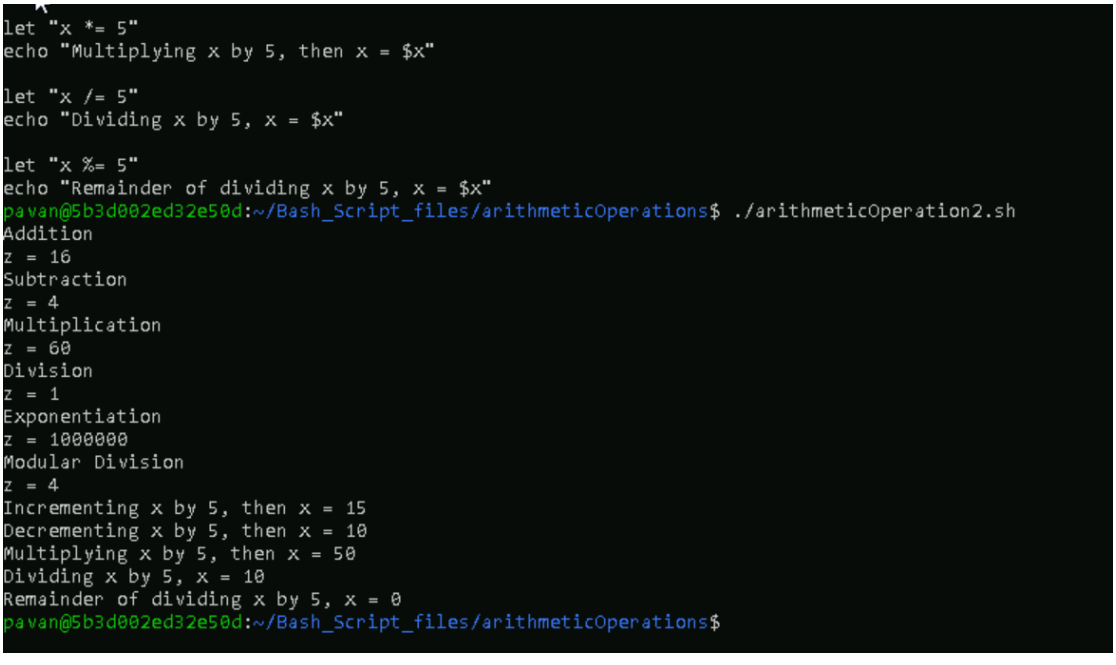
```
pavan@5b3d002ed32e50d:~/Bash_Script_files/arithmeticOperations$ vi arithmeticOperation1.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/arithmeticOperations$ cat arithmeticOperation1.sh
#!/bin/bash

x=8
y=2
echo "x=8, y=2"
echo "Addition of x & y"
echo $(( $x + $y ))
echo "Subtraction of x & y"
echo $(( $x - $y ))
echo "Multiplication of x & y"
echo $(( $x * $y ))
echo "Division of x by y"
echo $(( $x / $y ))
echo "Exponentiation of x,y"
echo $(( $x ** $y ))
echo "Modular Division of x,y"
echo $(( $x % $y ))
echo "Incrementing x by 5, then x= "
(( x += 5 ))
echo $x
echo "Decrementing x by 5, then x= "
(( x -= 5 ))
echo $x
echo "Multiply of x by 5, then x="
(( x *= 5 ))
echo $x
echo "Dividing x by 5, x= "
(( x /= 5 ))
echo $x
echo "Remainder of Dividing x by 5, x="
(( x %= 5 ))
echo $x
```

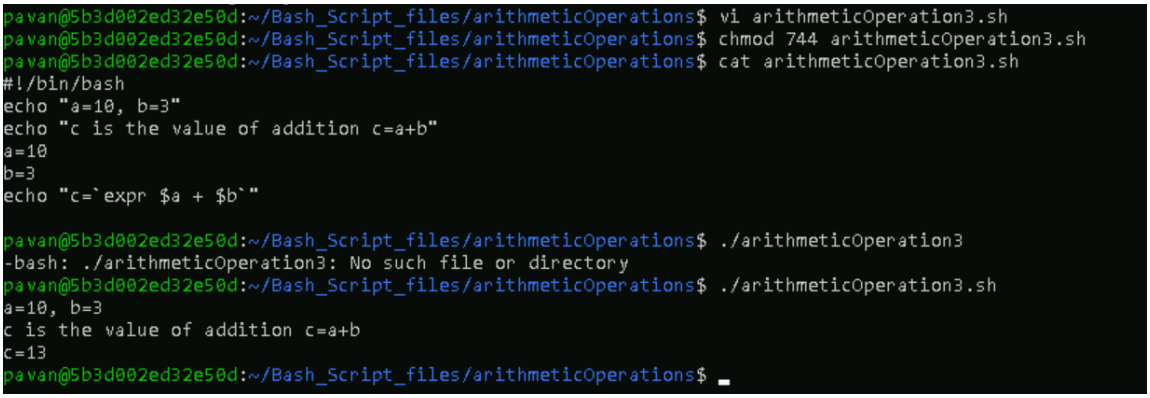
```
pavan@5b3d002ed32e50d:~/Bash_Script_files/arithmeticOperations$ ./arithmeticOperation1.sh
x=8, y=2
Addition of x & y
10
Subtraction of x & y
6
Multiplication of x & y
16
Division of x by y
4
Exponentiation of x,y
64
Modular Division of x,y
0
Incrementing x by 5, then x=
13
Decrementing x by 5, then x=
8
Multiply of x by 5, then x=
40
Dividing x by 5, x=
8
Remainder of Dividing x by 5, x=
3
```

2. To use let command in a Bash script.



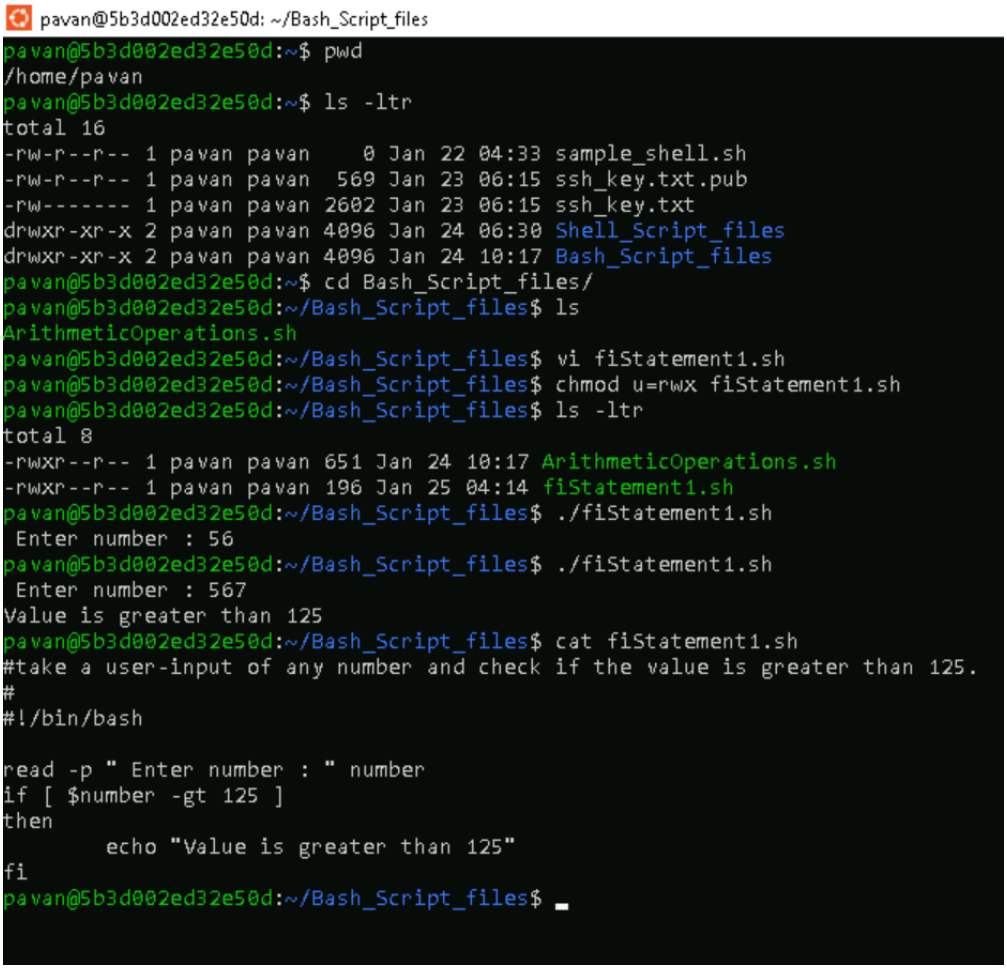


3. To use backticks and expr in a Bash script.



2.IF Statements

1. To take a user-input of any number and check if the value is greater than 125.



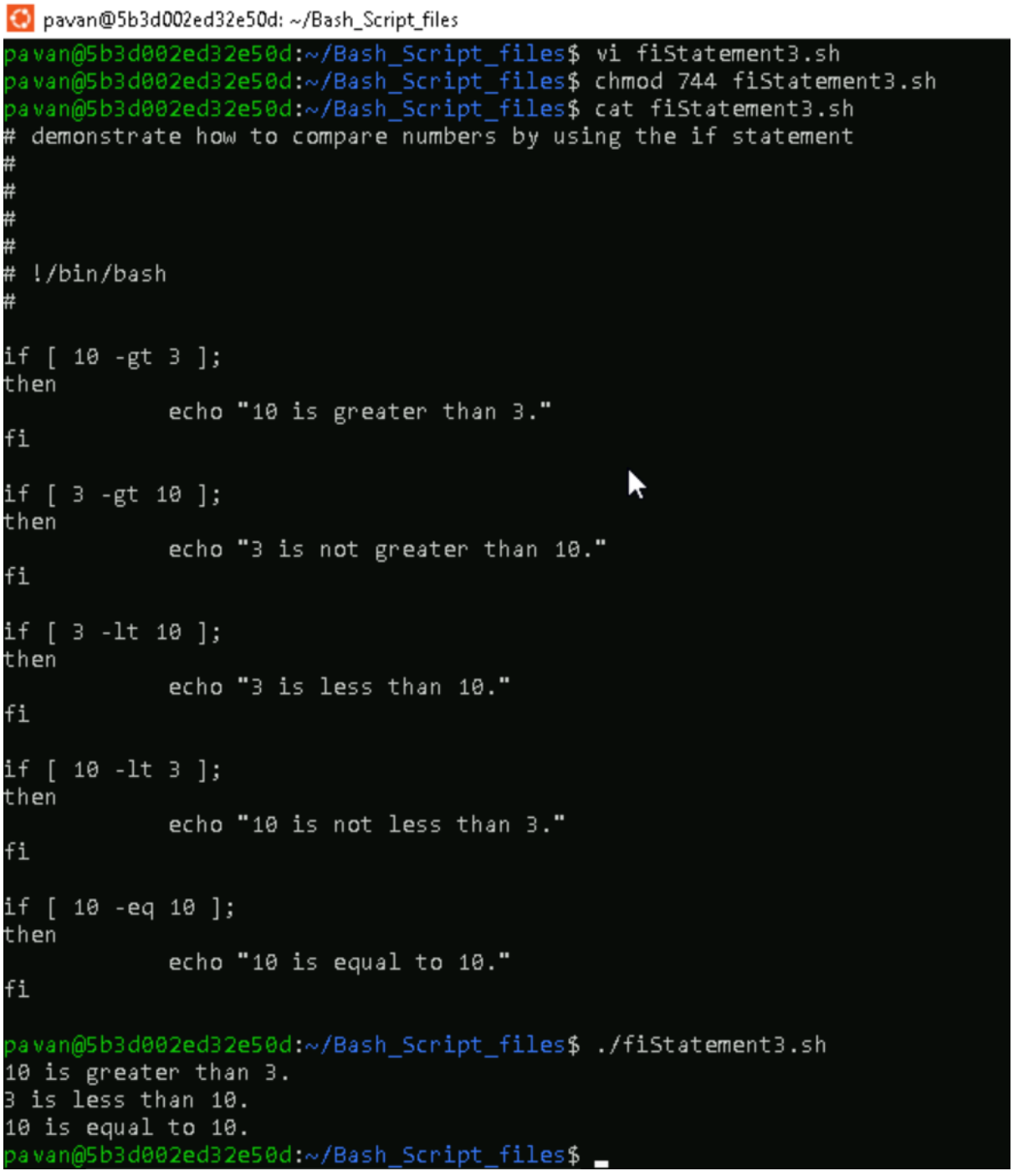
2. To use of if statement with a simple scenario of comparing two strings.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files$ vi fiStatement2.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ chmod u=rwx fiStatement2.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ ls -ltr
total 12
-rwxr--r-- 1 pavan pavan 651 Jan 24 10:17 ArithmeticOperations.sh
-rwxr--r-- 1 pavan pavan 196 Jan 25 04:14 fiStatement1.sh
-rwxr--r-- 1 pavan pavan 247 Jan 25 04:18 fiStatement2.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ cat fiStatement2.sh
#the usage of if statement with a simple scenario of comparing two strings
#
#!/bin/bash



if [ "myfile" == "myfile" ];
then
        echo "true condition"
fi
# if condition is false
if [ "myfile" == "yourfile" ];
then
        echo "false condition"
fi
pavan@5b3d002ed32e50d:~/Bash_Script_files$ ./fiStatement2
-bash: ./fiStatement2: No such file or directory
pavan@5b3d002ed32e50d:~/Bash_Script_files$ ./fiStatement2.sh
true condition
pavan@5b3d002ed32e50d:~/Bash_Script_files$
```

3. To compare numbers by using the if statement.



4. To use AND operator to include multiple conditions in the if expression.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files$ vi fiStatement4.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ chmod 744 fiStatement4.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ cat fiStatement4.sh
#In this program, we will define how to use AND operator to include multiple conditions in the if expression
#!/bin/bash


if [ 8 -gt 6 ] && [ 10 -eq 10 ]; then
        echo "Conditions are true"
fi

if [ "mylife" == "mylife" ] && [ 3 -gt 10 ]; then
        echo "Conditions are false"
fi

pavan@5b3d002ed32e50d:~/Bash_Script_files$ ./fiStatement4.sh
Conditions are true
pavan@5b3d002ed32e50d:~/Bash_Script_files$
```

5. To use OR operator to include multiple conditions in the if expression.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files$ vi fiStatement5.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ chmod 744 fiStatement5.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ cat fiStatement5.sh
#In this program, we will define how to use OR operator to include multiple conditions in the if expression
#!/bin/bash

if [ 8 -gt 7 ] || [ 10 -eq 3 ]; then
        echo "Condition is true."
fi

if [ "mylife" == "yourlife" ] || [ 3 -gt 10 ]; then
        echo "Condition is false."
fi

pavan@5b3d002ed32e50d:~/Bash_Script_files$ ./fiStatement5.sh
Condition is true.
pavan@5b3d002ed32e50d:~/Bash_Script_files$
```

6. To use AND and OR to include multiple conditions in the if expression.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files$ vi fiStatement6.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ chmod 744 fiStatement6.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ cat fiStatement6.sh
#In this program, we will define how to use AND and OR to include multiple conditions in the if expression
#!/bin/bash
#

if [[ 10 -eq 10 && 5 -gt 4 || 3 -eq 4 || 3 -lt 6 ]]; then
        echo "Condition is true."
fi

if [[ 8 -eq 8 && 8 -gt 10 || 9 -lt 5 ]]; then
        echo "Condition is false"
fi

pavan@5b3d002ed32e50d:~/Bash_Script_files$ ./fiStatement6.sh
Condition is true.
pavan@5b3d002ed32e50d:~/Bash_Script_files$
```

3.IF ELSE Statements

1. To use IF-ELSE statement.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files$ ls
ArithmeticOperations.sh  fiStatement1.sh  fiStatement2.sh  fiStatement3.sh  fiStatement4.sh  fiStatement5.sh  fiStatement6.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files$ mkdir ifElseStatements
pavan@5b3d002ed32e50d:~/Bash_Script_files$ cd ifElseStatements/
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ls
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ vi ifElseStatement1.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ chmod u=rwx ifElseStatement1.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ls -l
total 4
-rwxr--r-- 1 pavan pavan 246 Jan 25 04:51 ifElseStatement1.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ cat ifElseStatement1.sh
#!/bin/bash
if [ 10 -gt 3 ];
then
            echo "10 is greater than 3."
    else
                echo "10 is not greater than 3."
fi

if [ 3 -gt 10 ];
then
            echo "3 is greater than 10."
    else
                echo "3 is not greater than 10."
fi

pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ./ifElseStatement1.sh
10 is greater than 3.
3 is not greater than 10.
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$
```

2. To use IF-ELSE statement and bash logical operators to join multiple conditions.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ vi ifElseStatement2.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ chmod u=rwx ifElseStatement2.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ cat ifElseStatement2.sh
#!/bin/bash

if [[ 10 -gt 9 && 10 == 9 || 2 -lt 1 || 25 -gt 20 ]]; then
            echo "Given condition is true."
    else
                echo "Given condition is false."
fi

if [[ 10 -gt 9 && 10 == 8 || 3 -gt 4 || 8 -gt 8 ]]; then
            echo "Given condition is true."
    else
                echo "Given condition is not true."
fi
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ./ifElseStatement2.sh
Given condition is true.
Given condition is not true.
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$
```

3. To use IF-ELSE statement along with “-gt” operator.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ vi ifElseStatement3.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ chmod u=rwx ifElseStatement3.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ cat ifElseStatement3.sh
#!/bin/bash


read -p "Enter a value:" value
if [ $value -gt 9 ]; then echo "The value you typed is greater than 9."; else echo "The value you typed is not greater than 9.";fi

pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ./ifElseStatement3.sh
Enter a value:4
The value you typed is not greater than 9.
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ./ifElseStatement3.sh
Enter a value:-79
The value you typed is not greater than 9.
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ./ifElseStatement3.sh
Enter a value:247
The value you typed is greater than 9.
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$
```

4. To use NESTED IF-ELSE statements in bash.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ vi ifElseStatement4.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ chmod u=rwx ifElseStatement4.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ls -l
total 16
-rwxr--r-- 1 pavan pavan 246 Jan 25 04:51 ifElseStatement1.sh
-rwxr--r-- 1 pavan pavan 315 Jan 25 04:54 ifElseStatement2.sh
-rwxr--r-- 1 pavan pavan 194 Jan 25 04:57 ifElseStatement3.sh
-rwxr--r-- 1 pavan pavan 261 Jan 25 05:03 ifElseStatement4.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ cat ifElseStatement4.sh
#!/bin/bash


read -p "Enter a value:" value
if [ $value -gt 9 ];
then
        if [ $value -lt 11 ];
        then
                echo "$value>9, $value<11"
        else
                echo "The value you typed is greater than 9."
        fi
else
        echo "The value you typed is not greater than 9."
fi

pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ./ifElseStatement4.sh
Enter a value:10
10>9, 10<11
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ./ifElseStatement4.sh
Enter a value:-3483
The value you typed is not greater than 9.
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$ ./ifElseStatement4.sh
Enter a value:67
The value you typed is greater than 9.
pavan@5b3d002ed32e50d:~/Bash_Script_files/ifElseStatements$
```

4. ELSE-IF Statements

1. To use ELSE-IF statement.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files$ mkdir elseIfStatements
pavan@5b3d002ed32e50d:~/Bash_Script_files$ cd elseIfStatements/
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ vi elseIfStatement1.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ls -l
total 4
-rw-r--r-- 1 pavan pavan 254 Jan 25 05:07 elseIfStatement1.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ chmod u=rxw elseIfStatement1.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ls -l
total 4
-rwxr--r-- 1 pavan pavan 254 Jan 25 05:07 elseIfStatement1.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ cat elseIfStatement1.sh
#!/bin/bash

read -p "Enter a number of quantity:" num

if [ $num -gt 100 ];
then
        echo "Eligible for 10% discount"
elif [ $num -lt 100 ]; then
        echo "Eligible for 5% discount"
else
        echo "Lucky Draw Winner"
        echo "Eligible to get the item for free"
fi

pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ./elseIfStatement1.sh
Enter a number of quantity:101
Eligible for 10% discount
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ./elseIfStatement1.sh
Enter a number of quantity:99
Eligible for 5% discount
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ./elseIfStatement1.sh
Enter a number of quantity:100
Lucky Draw Winner
Eligible to get the item for free
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$
```

2. To use ELSE-IF statement along with bash logical operators.

```
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ vi elseIfStatement2.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ chmod u=rxw elseIfStatement2.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ls -l
total 8
-rwxr--r-- 1 pavan pavan 254 Jan 25 05:07 elseIfStatement1.sh
-rwxr--r-- 1 pavan pavan 377 Jan 25 05:11 elseIfStatement2.sh
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ cat elseIfStatement2.sh
#!/bin/bash

read -p "Enter a number of quantity:" num

if [ $num -gt 200 ];
then
        echo "Eligible for 20% discount"
elif [[ $num == 200 || $num == 100 ]]; then
        echo "Lucky Draw Winner"
        echo "Eligible to get the item for free"
elif [[ $num -gt 100 && $num -lt 200 ]]; then
        echo "Eligible for 10% discount"
elif [ $num -lt 100 ]; then
        echo "No discount"
fi
#Else is optional
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ./elseIfStatement2.sh
Enter a number of quantity:420
Eligible for 20% discount
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ./elseIfStatement2.sh
Enter a number of quantity:121
Eligible for 10% discount
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ./elseIfStatement2.sh
Enter a number of quantity:25
No discount
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$ ./elseIfStatement2.sh
Enter a number of quantity:100
Lucky Draw Winner
Eligible to get the item for free
pavan@5b3d002ed32e50d:~/Bash_Script_files/elseIfStatements$
```