

DoP :21/02/2024

Aim: Implement shell script to demonstrate conditional statements

Learning Outcomes: 1. To understand the conditional statements

2. To Demonstrate the shell script to demonstrate conditional statements using Linux command.

Hardware/Software:

Theory:

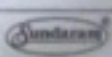
`else`
 # code block to execute if condition false
`fi`

- `if... else if... else fi` statement (else if ladder):
The 'if... else if... else' statement allows testing multiple conditions sequentially. It starts with an 'if' statement, followed by zero or more 'elif' statements, and ends with an optional 'else' statement. Each condition is evaluated in order, and the corresponding code block is executed for the first true condition encountered.

syntax: `if [condition1]; then`
 # code block to executed if condition1 is true
`elif [condition2]; then`
 # code block to execute if condition2 is true
`else`
 # code block to execute if all conditions are false
`fi`

- `if... then... else if... then` (nested): It allows conditional structures within other conditional structures. The inner 'if' statement is evaluated only if the outer 'if' statement's condition is true. This enables more complex decision-making by evaluating conditions within conditions.

syntax: `if [condition1]; then`
 `if [condition2]; then`
 # code block to execute if both conditions are true
 `fi`
`else`
 # code to be executed if condition 1 is false
`fi`

 FOR EDUCATIONAL USE

- a) Shell script to print whether the number entered by the user is even or odd
- b) Shell script to print the largest of three numbers

```
1 #odd or even
2 echo "Enter a number"
3 read n
4 if [ $n -eq 0 ]
5 then
6 echo "neither odd nor even"
7 elif [ `expr $n % 2` -eq 0 ]
8 then
9 echo "even"
10 else
11 echo "odd"
12 fi
13 echo " "
14
15 #largest of 3 numbers
16 echo "Enter 3 numbers"
17 read a
18 read b
19 read c
20 if [ $a -ge $b ]
21 then
22 if [ $a -ge $c ]
23 then
24 echo "$a is the greatest number"
25 else
26 echo "$c is the greatest number"
27 fi
28 else
29 if [ $b -ge $c ]
30 then
31 echo "$b is the greatest number"
32 else
33 echo "$c is the greatest number"
34 fi
35 fi
36 echo " "
37
```

```

1 #odd or even
2 echo "Enter a number"
3 read n
4 if [ $n -eq 0 ]
5 then
6 echo "neither odd nor even"
7 elif [ `expr $n % 2` -eq 0 ]
8 then
9 echo "even"
10 else
11 echo "odd"
12 fi
13 echo " "
14
15 #largest of 3 numbers
16 echo "Enter 3 numbers"
17 read a
18 read b
19 read c
20 if [ $a -ge $b ]
21 then
22 if [ $a -ge $c ]
23 then
24 echo "$a is the greatest number"

```

```

38 #leap year or not
39 echo "Enter a year:"
40 read y
41
42 if [ $((y % 4)) -eq 0 ] && [ $((y % 100)) -ne 0 ]; then
43     echo "$y is a leap year."
44 elif [ $((y % 400)) -eq 0 ]; then
45     echo "$y is a leap year."
46 else
47     echo "$y is not a leap year."
48 fi
49
50 #tax
51 echo "Enter balance: "
52 read bal

```

not

```

53 38 #leap year or not
54 e39 echo "Enter a year:"
55 r40 read y
56 41
57 42 if [ $((y % 4)) -eq 0 ] && [ $((y % 100)) -ne 0 ]; then
58 43     echo "$y is a leap year."
59 e44 elif [ $((y % 400)) -eq 0 ]; then
60 45     echo "$y is a leap year."
61 46 else
62 47     echo "$y is not a leap year."
63 48 fi
64 49
65 50 #tax
66 51 echo "Enter balance: "
67 52 read bal
68 53
69 54 echo "Enter withdrawal: "
70 55 read wd
71 56
72 57 if (( wd > bal )); then
73 58     echo "Insufficient balance"
74 59 else
75 60     tax=0

```

d) Shell
script to

Steps to execute the program:

- Create a shell script file using text editor like 'gedit'
- Write the code with proper spacing in it and save it with .sh
- Open a terminal window on your system.
- Use 'cd' command to navigate to the directory where the shell script file is located.
- Use 'bash' command to execute your file.
eg. bash abc.sh.

Conclusion:

Conditional statements are essential in shell scripting for making decisions based on conditions. These constructs, ranging from simple 'if' statements to nested structures, enable precise control over script execution. Understanding their syntax allows developers to create efficient and adaptable scripts to meet various requirements effectively.

Output:

nditions:

hdrawal

ax as 4% of

drawal

ent balance

Steps to execute the program:

- Create a shell script file using text editor like 'gedit'
- Write the code with proper spacing in it and save it with .sh
- Open a terminal window on your system.
- Use 'cd' command to navigate to the directory where the shell script file is located.
- Use 'bash' command to execute your file.
eg. bash abc.sh.

Conclusion:

Conditional statements are essential in shell scripting for making decisions based on conditions. These constructs, ranging from simple 'if' statements to nested structures, enable precise control over script execution. Understanding their syntax allows developers to create efficient and adaptable scripts to meet various requirements effectively.

```
user@satviki:~/Documents$ bash exp5.sh
Enter a number
4
even

Enter 3 numbers
4
5
5
5 is the greatest number

Enter a year:
2004
2004 is a leap year.
Enter balance:
5000
Enter withdrawal:
1000
Amount withdrawn: 1000
Tax deducted: 30
Amount withdrawn after tax: 970
```

Conclusion:
assessment schemes.

Attendance	Discipline	Short oral	Correctness of Lab Report	Timely completion of Lab Report	Total marks (10)	Signature of Teacher with Date