Conditional Flow Statements

Arithmetic Operators

expr command

In shell script all variables hold string value even if they are numbers. So, to perform arithmetic operations we use the `expr` command.

The `expr` command can only work with integer values. For floating point numbers we use the `bc` command.

To compute the result we enclose the expression in backticks `` ` ` ``.

Addition

```
#!/bin/bash

# take two integers from the user
echo "Enter two integers: "
read a b

# perform addition
result=`expr $a + $b`

# show result
echo "Result: $result"
```

Multiplication

```
#!/bin/bash

# take two numbers from user
echo "Enter two numbers: "
read a b

# compute multiplication result
result=`expr "$a * $b" | bc`

# print output
echo "Result: $result"
```

Relational Operators

- Equal to -eq
- Not equal to -ne
- Greater than -gt
- Less than -lt
- Greater than or equal to -ge
- Less than or equal to -le

Equal to -eq

```
#!/bin/bash

# take two numbers from the user
echo "Enter two numbers: "
read a b

# check
if [ $a -eq $b ]
then
  echo "Numbers are equal."
else
  echo "Not equals."
fi
```

Note:give space after [and before] like [$a -eq $b].

Logical Operators

We use the logical operators to test more than one condition.

-a	Logical AND
-o	Logical OR

Logical AND

```
#!/bin/bash

# take a number from the user
echo "Enter a number: "
read a

# check
if [ `expr $a % 2` == 0 -a $a -gt 10 ]
then
  echo "$a is even and greater than 10."
else
  echo "$a failed the test."
fi
```

String Operators

We use the `=` equal operator to check if two strings are equal.

```
#!/bin/bash

# take two strings from user
echo "Enter first string:"
read str1

echo "Enter second string:"
read str2
# check
if [ "$str1" = "$str2" ]
then
  echo "Strings are equal."
else
  echo "Strings are not equal."
fi
```

-z to check size zero

```
#!/bin/bash

# take a string from user
echo "Enter string:"
read str

# check
if [ -z "$str" ]
then
  echo "String size equal to 0."
else
  echo "String size not equal to 0."
fi
```

${#str} to find length of the string

```
#!/bin/sh

# take a string from user
echo "Enter string:"
read str

echo "Length of the entered string = ${#str}"
```

If Else statement

Following is the syntax of the if statement.

```
if [ condition ]
then
  # if block code
fi
```

to check if two numbers are equal

```
#!/bin/bash

# take two numbers from the user
echo "Enter two numbers: "
read a b

# check
if [ $a == $b ]
then
  echo "Numbers are equal."
fi

echo "End of script."
```

If else

```
#!/bin/sh

# take two numbers from the user
echo "Enter two numbers: "
read a b

# check
if [ $a == $b ]
then
  echo "Numbers are equal."
else
  echo "Numbers are not equal."
fi

echo "End of script."
```

If elif else

```
if [ condition ]
then
  # if block code
elif [ condition2 ]
then
  # elif block code
else
  # else block code
fi
```

Write a Shell Script to check if a number is odd, even or zero

Case statement

Similar to the if statement we use the case statement to make decisions and execute a block of code based on some match.

```
case word in
  pattern1)
     # block of code for pattern1
     ;;

  pattern2)
     # block of code for pattern2
     ;;

  *)
     # default block
     ;;
esac
```

Example

```
#!/bin/bash
# take a number from user
echo "Enter number:"
read num
case $num in
  1)
     echo "It's one!"
     ;;

  2)
     echo "It's two!"
     ;;

  *)
     echo "It's something else!"
     ;;
esac
```

Write a Shell Script to display greetings

```
take user name
take time of the day
```