Lab 0(c): Shell Scripting in Details

What is a Shell?

A shell is a program which reads and executes commands for the user. Shells also usually provide features such job control, input and output redirection and a command language for writing shell scripts.

Command for Compile Shell Program:

sh <file name>.sh (*.*sh* is extension name)

Arithmetic Operator:

Operator	**Description**	**Example**
+	Addition - Adds values on either side of the operator	`expr $a + $b` will give 30
-	Subtraction - Subtracts right hand operand from left hand operand	`expr $a - $b` will give -10
*	Multiplication - Multiplies values on either side of the operator	`expr $a * $b` will give 200
/	Division - Divides left hand operand by right hand operand	`expr $b / $a` will give 2
%	Modulus - Divides left hand operand by right hand operand and returns remainder	`expr $b % $a` will give 0
=	Assignment - Assign right operand in left operand	a=$b would assign value of b into a
==	Equality - Compares two numbers, if both are same then returns true.	[$a == $b]
!=	Not Equality - Compares two numbers, if both are different then returns true.	[$a != $b]

It is very important to note here that all the conditional expressions would be put inside square braces with one spaces around them, for example [$a == $b] is correct whereas *[$a==$b] is incorrect*.
All the arithmetical calculations are done using long integers.

Relational Operator:

Operator	Description	Example
-eq	Checks if the value of two operands are equal or not, if yes then condition becomes true.	[$a -eq $b] is not true.
-ne	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	[$a -ne $b] is true.
-gt	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	[$a -gt $b] is not true.
-lt	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	[$a -lt $b] is true.
-ge	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	[$a -ge $b] is not true.
-le	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	[$a -le $b] is true.

Boolean Operator:

Operator	Description	Example
!	This is logical negation. This inverts a true condition into false and vice versa.	[! false] is true.
-o	This is logical OR. If one of the operands is true then condition would be true.	[$a -lt 20 -o $b -gt 100] is true.
-a	This is logical AND. If both the operands are true then condition would be true otherwise it would be false.	[$a -lt 20 -a $b -gt 100] is false.

String Operators:

Operator	Description	Example
=	Checks if the value of two operands are equal or not, if yes then condition becomes true.	[$a = $b]
!=	Checks if the value of two operands are equal or not, if values are not equal then condition	[$a != $b]

	becomes true.	
-z	Checks if the given string operand size is zero. If it is zero length then it returns true.	[-z $a]
-n	Checks if the given string operand size is non-zero. If it is non-zero length then it returns true.	[-z $a]
str	Check if str is not the empty string. If it is empty then it returns false.	[$a]

Conditional Operations:

if

```
if [ condition ]
      then
            commands
fi
```

if - else

```
if [ condition ]
      then
            commands
      else
            commands
fi
```

nested if

```
if condition;
      then
            commands
elif condition;
      then
            commands
fi
```

The case...esac Statement

```
case word
      in
      pattern1)
            Statement(s) to be executed if pattern1 matches
            ;;
      pattern2)
```

```
            Statement(s) to be executed if pattern2 matches
            ;;
      pattern3)
            Statement(s) to be executed if pattern3 matches
            ;;
esac
```

for loop

```
for VARIABLE in 1 2 3 4 5 .. N
do
      command1
      command2
      commandN
done
```

For loop with expression

```
for (( EXP1; EXP2; EXP3 ))
do
      command1
      command2
      command3
done
```

While statement

```
while [ condition ]
do
      command1
      command2
      commandN
done
```

Lab Work:

1. Write a shell program check whether a given number is odd or even.
2. Write a shell program to see current date, time, username, and current directory.
3. Write a shell program convert Celsius to Fahrenheit.
4. Write a shell program to check whether the given year is a leap year or not.
5. Write a shell program to accept 2 arguments from user and perform all arithmetic operation (+ , - , * , /)

Assignments:

1. Write a shell program to find Fibonacci numbers within a given range.
2. Write a shell program to find sum of digits of a number.
3. Write a shell program count the number of characters, words and lines in a file.
4. Write a shell program to check whether given number is Armstrong number or not.
5. Write shell program to reverse a given number.
6. Write a shell program to find out factorial number from a given number.
7. Write a shell program given number is prime number or not.
8. Write a shell program to check whether a number is palindrome or not.
9. Write a shell program for calculating GCD.
10. Write a shell program for calculating LCM.