Fibonacci series

```
N=6

a=0

b=1

echo "The Fibonacci series is : "

for (( i=0; i<N; i++ ))
do
   echo -n "$a "
   fn=$((a + b))
   a=$b
   b=$fn
done
```

Program to swap two numbers

```
first=5
second=10

temp=$first
first=$second
second=$temp
```

```
echo "After swapping, numbers are:"
echo "first = $first, second = $second"
```

1) Accessing variable

Variable data could be accessed by appending the variable name with '$' as follows:

```
#!/bin/bash

VAR_1="Devil"
VAR_2="OWL"

echo "$VAR_1$VAR_2"
```

Output:

2) Unsetting Variables

The unset command directs a shell to delete a variable and its stored data from list of variables. It can be used as follows:

```
#!/bin/bash

var1="Devil"
var2=23
echo $var1 $var2

unset var1

echo $var1 $var2
```

Output:

```
DEVIL 23
23
```

3) Read only Variables.

These variables are read only i.e., their values could not be modified later in the script. Following is an example:

```
#!/bin/bash

var1="Devil"

var2=23

readonly var1

echo $var1 $var2

var1=23

echo $var1 $var2
```

Output:

```
Devil 23

./bash1: line 8: var1: readonly variable

Devil 23
```

Conditional Statements | Shell Script

Conditional Statements: There are total 5 conditional statements which can be used in bash programming

1. if statement
2. if-else statement
3. if..elif..else..fi statement (Else If ladder)
4. if..then..else..if..then..fi..fi..(Nested if)
5. switch statement

Their description with syntax is as follows:

if statement

This block will process if specified condition is true.

Syntax:

```
if [ expression ]
then
   statement
fi
```

if-else statement

If specified condition is not true in if part then else part will be execute.

Syntax

```
if [ expression ]
then
   statement1
else
   statement2
fi
```

if..elif..else..fi statement (Else If ladder)

To use multiple conditions in one if-else block, then elif keyword is used in shell. If expression1 is true then it executes statement 1 and 2, and this process continues. If none of the condition is true then it processes else part.

Syntax

```
if [ expression1 ]
then
   statement1
   statement2
   .
   .
elif [ expression2 ]
then
   statement3
   statement4
   .
   .
else
   statement5
fi
```

if..then..else..if..then..fi..fi..(Nested if)

Nested if-else block can be used when, one condition is satisfies then it again checks another condition. In the syntax, if expression1 is false then it processes else part, and again expression2 will be check.

Syntax:

```
if [ expression1 ]
then
   statement1
   statement2
   .
else
   if [ expression2 ]
   then
      statement3
      .
   fi
fi
```

switch statement

case statement works as a switch statement if specified value match with the pattern then it will execute a block of that particular pattern

When a match is found all of the associated statements until the double semicolon (;;) is executed.

A case will be terminated when the last command is executed.

If there is no match, the exit status of the case is zero.

Syntax:

```
case  in
   Pattern 1) Statement 1;;
   Pattern n) Statement n;;
esac
```

Example Programs

Example 1:

Implementing `if` statement

```
#Initializing two variables

a=10
```

```
b=20


#Check whether they are equal
if [ $a == $b ]
then
    echo "a is equal to b"
fi


#Check whether they are not equal
if [ $a != $b ]
then
    echo "a is not equal to b"
fi
```

Output

```
$bash -f main.sh
a is not equal to b
```

Example 2:

Implementing `if.else` statement

```
#Initializing two variables
a=20
b=20
```

```
if [ $a == $b ]

then

    #If they are equal then print this

    echo "a is equal to b"

else

    #else print this

    echo "a is not equal to b"

fi
```

Output

```
$bash -f main.sh

a is equal to b
```

Example 3:

Implementing `switch` statement

```
CARS="bmw"


#Pass the variable in string

case "$CARS" in

    #case 1

    "mercedes") echo "Headquarters - Affalterbach, Germany" ;;


    #case 2

    "audi") echo "Headquarters - Ingolstadt, Germany" ;;
```

```
    #case 3
    "bmw") echo "Headquarters - Chennai, Tamil Nadu, India" ;;
esac
```

Output

```
$bash -f main.sh
Headquarters - Chennai, Tamil Nadu, India.
```