Read User Input:

```
#!/bin/sh
echo "Please enter some input: "
read input_variable
echo "You entered: $input_variable"
```

```
#!/bin/bash
echo Please, enter your firstname and lastname
read FN LN
echo "Hi! $LN, $FN !"
```

```
#!/bin/bash
# read three numbers and assigned them to 3 variables
echo "Enter number one : "
read n1
echo "Enter number two : "
read n2
echo "Enter number three : "
read n3
```

```
#!/bin/bash
# Delete a directory

echo "Enter directory to delete : "
read dirname
rm -r $dirname
```

```
# install a package and show if it is installed

echo -e "Enter a package to install: "
read packagename
apt-get install $packagename
echo “”
dpkg -l $packagename
echo “The requested package $packagename was successfully installed”
```

```
#!/bin/bash
 # read a date from user and show calendar. Pay attention to echo command with \c and
without.

echo -e "Enter a month name (e.g. dec): \c"
read month
echo "Enter a year (e.g. 2012)"
read year

# display the calendar
cal –m $month $year
```

---

f Statement Syntax

```
if [ conditional expression ]
then
        statement1
        statement2
        .
Fi
```

If-elif-else Syntax

```
if [ expression 1 ]
then
   Statement(s) to be executed if expression 1 is true
elif [ expression 2 ]
then
   Statement(s) to be executed if expression 2 is true
elif [ expression 3 ]
then
   Statement(s) to be executed if expression 3 is true
else
   Statement(s) to be executed if no expression is true
fi
```

```
 #!/bin/sh

 a=10
 echo “Enter another number: ”
 read b

 if [ $a == $b ]
 then
   echo "a is equal to b"
 else
   echo "a is not equal to b"
 fi
```

```
 #!/bin/bash
 count=99
 if [ $count -eq 100 ]
 then
   echo "Count is 100"
 else
   echo "Count is not 100"
fi
```

```
#!/bin/sh
echo -e “Enter a number: \c”
read count
 if [ $count -eq 100 ]
 then
   echo "Count is 100"
 elif [ $count -gt 100 ]
 then
   echo "Count is greater than 100"
 else
   echo "Count is less than 100"
 fi
```

```
 #!/bin/bash
 # find if the number entered by the user is even or odd!
 number=0

 echo -n "Enter a number "
 read number
```

```
echo "The number you entered is: $number"
if [ $((number % 2)) -eq 0 ];
   then echo "Number is
   even"
else
  echo "Number is odd"
fi
```

Expressions used with if

Primary expressions

Primary	Meaning
[`-a` `FILE`]	True if `FILE` exists.
[`-b` `FILE`]	True if `FILE` exists and is a block-special file.
[`-c` `FILE`]	True if `FILE` exists and is a character-special file.
[`-d` `FILE`]	True if `FILE` exists and is a directory.
[`-e` `FILE`]	True if `FILE` exists.
[`-f` `FILE`]	True if `FILE` exists and is a regular file.
[`-g` `FILE`]	True if `FILE` exists and its SGID bit is set.
[`-h` `FILE`]	True if `FILE` exists and is a symbolic link.
[`-k` `FILE`]	True if `FILE` exists and its sticky bit is set.
[`-p` `FILE`]	True if `FILE` exists and is a named pipe (FIFO).
[`-r` `FILE`]	True if `FILE` exists and is readable.
[`-s` `FILE`]	True if `FILE` exists and has a size greater than zero.
[`-t` `FD`]	True if file descriptor `FD` is open and refers to a terminal.
[`-u` `FILE`]	True if `FILE` exists and its SUID (set user ID) bit is set.
[`-w` `FILE`]	True if `FILE` exists and is writable.
[`-x` `FILE`]	True if `FILE` exists and is executable.
[`-O` `FILE`]	True if `FILE` exists and is owned by the effective user ID.
[`-G` `FILE`]	True if `FILE` exists and is owned by the effective group ID.
[`-L` `FILE`]	True if `FILE` exists and is a symbolic link.
[`-N` `FILE`]	True if `FILE` exists and has been modified since it was last read.
[`-S` `FILE`]	True if `FILE` exists and is a socket.

[`FILE1` -`nt` `FILE2`]	True if `FILE1` has been changed more recently than `FILE2`, or if `FILE1` exists and `FILE2` does not.
[`FILE1` -`ot` `FILE2`]	True if `FILE1` is older than `FILE2`, or is `FILE2` exists and `FILE1` does not.
[`FILE1` -`ef` `FILE2`]	True if `FILE1` and `FILE2` refer to the same device and inode numbers.

```
#!/bin/bash
echo “enter an absolute path to a file name:”
read FILE

if [ -f "$FILE" ];
then
echo "File $FILE exist."
else
echo "File $FILE does not exist" > /tmp/ExistsOrNot.txt  #write the error message to a file!
fi
```

```
#!/bin/bash
echo “enter a directory path:”
read dirpath
dirpath=$dirpath/Test*
 if  test -s $dirpath
      then
         echo "found one"
    else
       echo "found none"
 fi
```

```
#!/bin/bash
echo “enter an absolute path to a file name:”
read FILE

if [ -g "$FILE" ];
then
   echo "File $FILE exist and its SGID is set"
else
#write the error message to a file!
   echo "File $FILE does not exist or SGID not set!" > /tmp/ExistsOrNot.txt
fi
```

For Statement

```
#!/bin/bash
for i in 1 2 3 4 5 do
echo "Welcome $i times"
done
```

```
#!/bin/bash
 i=1
 for day in "Mon Tue Wed Thu Fri"
 do
  echo "Weekday $((i++)) : $day"
 done
```

While Statement

```
 #!/bin/bash
 COUNTER=0
 while [  $COUNTER -lt 10 ];
 do
       echo The counter is $COUNTER
       COUNTER=COUNTER+1
 done
```

Case (Normal Case)

```
 #!/bin/bash

 echo "What is your preferred programming / scripting language"
 echo "1) bash"
 echo "2) perl"
 echo "3) phyton"
 echo "4) c++"
 echo "5) I do not know !"
 read selection;
 #simple case bash structure
 case $selection in
    1) echo "You selected bash";;
    2) echo "You selected perl";;
    3) echo "You selected phyton";;
    4) echo "You selected c++";;
 5) exit
esac
```

```
 #!/bin/bash

 echo "What is your preferred programming / scripting language"
 echo "a) bash"
 echo "b) perl"
 echo "c) phyton"
 echo "d) c++"
 echo "e) I do not know !"
 read selection;
 #simple case bash structure
 case $selection in
    a | A) echo "You selected bash";;
    b | B) echo "You selected perl";;
    c | C) echo "You selected phyton";;
    d | D) echo "You selected c++";;
 e | E) exit
esac
```

Case example (repeat until user exits (using until))

```
 #!/bin/bash

 Selection =
 until [ "$selection" = "0" ];
 do echo ""
    echo "PROGRAM MENU"
    echo "1 - display free disk space"
    echo "2 - display free memory"
    echo ""
    echo "0 - exit program"
    echo ""
    echo -n "Enter selection: "
    read selection
    echo ""
    case $selection in
       1 ) df ;;
       2 ) free ;;
       0 ) exit ;;
       * ) echo "Please enter 1, 2, or 0"
    esac
 done
```