Shell script Programming

What is SHELL

Computer program which is designed to be executed by the Unix/Linux shell. In the linux/unix system, the shell is the command line interface/interpreter.

Shell script is execute in the shell. The following are the different shell

1) **Bourne shell**
2) **C shell**
3) **Korn shell**

Shell is a real programming language which consists of variables, control structures and so on. It has a list of command that is executed sequentially.

Shell Prompt

$ Command prompt, given by the shell.

In the Bourne shell $ is the default prompt

In the C-shell % is the default prompt.

How to define the variables

Variable is character string to which we assign a value. The value assigned could be (number, text).

Variable is a pointer to the data. The variable can contain only letters from a-z and A-Z, number (0-9) and underscore character

SHELL VARIABLE

```
echo "What is your university name"

read university

echo "welcome" $university
```

SHELL ARRAYS

How to use the shell arrays in UNIX. Shell supports the type of the variable called array variable, it holds the multiple values at the same time.

Advantage: We can use only a single array variable to store all the variables

How to create an array variable

Array_name [index]=value

How to access array values = ${array_name)[index]}

e.g

NAME[0]="NITHIN"

NAME[1]="JADHAV"

NAME[2]="JEEVA"

echo "First Index: ${NAME[0]}"

echo "Second Index: ${NAME[1]}"

We can access all the items in an array using

```
${array_name[*]}
${array_name[@]}
```

NAME[0]="NITHIN"

NAME[1]="JADHAV"

NAME[2]="JEEVA"

echo "First Method: ${NAME[*]}"

echo "Second Method: ${NAME[@]}"

Shell Basic Operators

val=`expr 5 + 5`

echo "sum : $val"

Sample shell programming

Program one:

Demonstrates the use of test command

 # if basic is less than 1500, then HRA=10% DA=90%

 # if his salary is either equal to or above 1500 then HRA = 150 DA 1350

```
echo "enter basic salary"
read sal
if [ $sal -eq 1500 ]
then
hra= expr $sal  \* 10 / 100
da= expr $sal \* 90 / 100
echo $hra
echo $da
fi
```

Program two:

Prints the square of integers in succession

```
i=1
while [ $i -lt 5 ]
do
sq=$(($i * $i))
echo $sq
i=$(($i + 1))
done
```

```
echo  "job"
```

Program three:

Nested if Statements and the elif Construct

Syntax

if command
then
command
else
if command
then
command
else
if command
then
command
fi
fi
fi

It is useful to select an option from a given set of options

code:

```
echo Enter either 1 or 2
read i
if [ $i -eq 1 ]
then
echo you would go to heaven!
else
if [ $i -eq 2 ]
then
echo  hello was created with you in mind
else
```

```
echo How about mother earth!
fi
fi
```

Program four : prints the given digit in words.

```
case value in
pattern1)
command
command ;;
...
patternn)
command;
esac
```

if –elif statement to perform a multiply branch, but if all the branch depend on the value of a single variable then we can use case…..esac statement

```
echo "enter a number from 1 to 8"
read num
echo  "entered number is:"
case $num in
1)      echo one ;;
2)      echo two ;;
3)      echo three ;;
4)      echo four ;;
5)      echo five ;;
6)      echo six ;;
7)      echo seven ;;
8)      echo eight ;;
esac
```