GETTING STARTED
WITH
SHELL VARIABLES

➢ SHELL VARIABLES:

- Variable is a data name or Memory location name.
- It is a temporary storage location.
- It values can change during execution of the program.
- No data types in shell scripting.
- Each and every variable, it treated as string variable.
- Shell variables are in integral part of shell programming. They provide the ability to store and manipulate information within a shell program.
- The variables you use are completely under your control.
- You can create and destroy any number of variables as needed to solve the problem at hand.

RULES OF BUILDING SHELL VARIABLES:

- A variable name is any combination of alphabets, digits and underscore ('_').
- No commas or blanks are allowed within a variable name.
- The first character of a variable name must either be an alphabet or an underscore.
- variable names should be of any responsible length.
- Variable names are case-sensitive.
- Here are a few sample variable names:

si_int
m_hra
pop_e_89
name

NOTE: While assigning values to variables using the assignment operator '=' , there should be no space on either side of =. If you leave a space the shell will try to interpret the values being assigned as a command to be executed.

SHELL KEYWORDS:

- Keywords are the words whose meaning has already been explained to the shell.
- The keywords can't be used as variable names because if we done so we are trying to assign a new meaning to the keyword, which is not allowed by the shell.
- The keywords are also be called as "Reserved words".

- Following is the list of keywords available in Bourne shell.

echo	if	until	trap
read	else	case	wait
set	fi	esac	eval
unset	while	break	exec
readonly	do	continue	ulimit
shift	done	exit	umask
export for return			

➢ SHELL VARIABLE TYPES:

- Shell Variables are classified into two types.
  - Unix Defined variables (or) System variables.
  - User defined variables.

SYSTEM VARIABLES:

- These are standard variables which are always accessible. The shell provides the values for these variables. These variables are usually used by the system itself.
- If we so desire, we can change the values of these variables as per our preferences and customize the system environment.

  → To list all system variables: **$set**

  → To change the system prompt: **$PS1="What next"**

  PATH: System defined variables are used to set application software paths like java,oracle..etc.

➢ USER DEFINED VARIABLES:

- These are defined by us and are used most extensively in shell programming.
- These are three types.
  - Local Variables
  - Global Variables.
  - Constant Variables.

LOCAL VARIABLES:

- All the variables we have used thus far have been local variables.
- A local variable is a variable which is either a variable declared within the function or is an argument passed to a function. As you may have encountered in your programming, if we declare variables in a function then we can only use them within that function
- Examples of local variables.

```
a=10
b=20
course=linux
```

→ To get a variable: **$echo $a**
$echo $course

GLOBAL VARIABLES:

- A global variable (DEF) is a variable which is accessible in multiple scopes. It is important to note that global variables are only accessible after they have been declared.
- **"export"** is the keyword, to create global variables.

$export <variable name>
$export a

CONSTANT VARIABLES:

- The constant variable in the equation was a fixed cost so we could do nothing about it unless we made a drastic change.
- "readonly" is the keyword to create constant variables or unchanging variables.

$x=100
$readonly x

WIPING OUT VARIABLES:

- Variables can be made to cease existing. If we want the shell to forget about a variable altogether, we use the **unset command.**
  → To erase a variable value: **$unset a**
  → Unset system variable: **$unset PS1**

NOTE: It is not allowed, since if PS1 is unset then there would be no system prompt left for you.

➢ A FEW TIPS & TRAPS OF SHELL VARIABLES:

- All shell variables are string variables. In the statement a=20, the '20' stored in 'a' is treated not as a number, but as a string of character 2 and 0.
- A variable may contain more than one word.

  $c="Two words"

- We can carry out more than one assignment in a line.

  $name=India age=69
  $echo $name $age

- All variables are defined inside a shell script die the moment the execution of the script is over.
- A variable which has been defined but has not been given any value is known as a null variable. A null variable can be created in any of the following ways.

  $d=""
  $d="
  $d=

- On echoing a null variable, only a blank line appears on the screen.
- If a null variable is used anywhere in a command, the shell manages to ignore it.
- Not only the system variables but also the user defined variables defined at the $ prompt or in a shell script can be displayed using the set command.