

Department Of Cybersecurity

Linux Programming

Assignment 7

Name: Mohd Sahil Ansari USN: ENG24CY0134

Roll No:22 Class: 3A

1. What is a bash shell script? Give one example. (CO4)

Ans:

A bash shell script is a file which consists of a set of commands written according to task assigned , and are written in the Bash i.e. Bourne Shell Again.

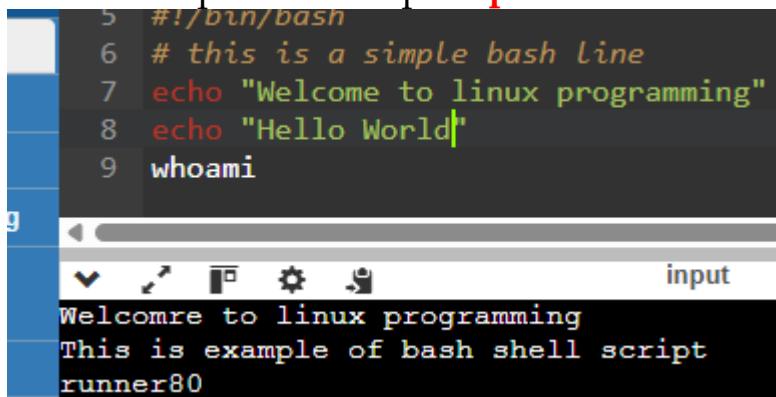
It allows the user to automate the tasks by executing multiple commands in sequence.

These are interpreted line by line by bash shell interpreter.

```
#!/bin/bash
# This is a simple bash script
echo "Welcomre to linux programming"
echo "This is example of bash shell script"
whoami|
```

Example:

2. Write a simple shell script to print "Hello World". (CO4)



```
5 #!/bin/bash
6 # this is a simple bash line
7 echo "Welcome to linux programming"
8 echo "Hello World"
9 whoami
```

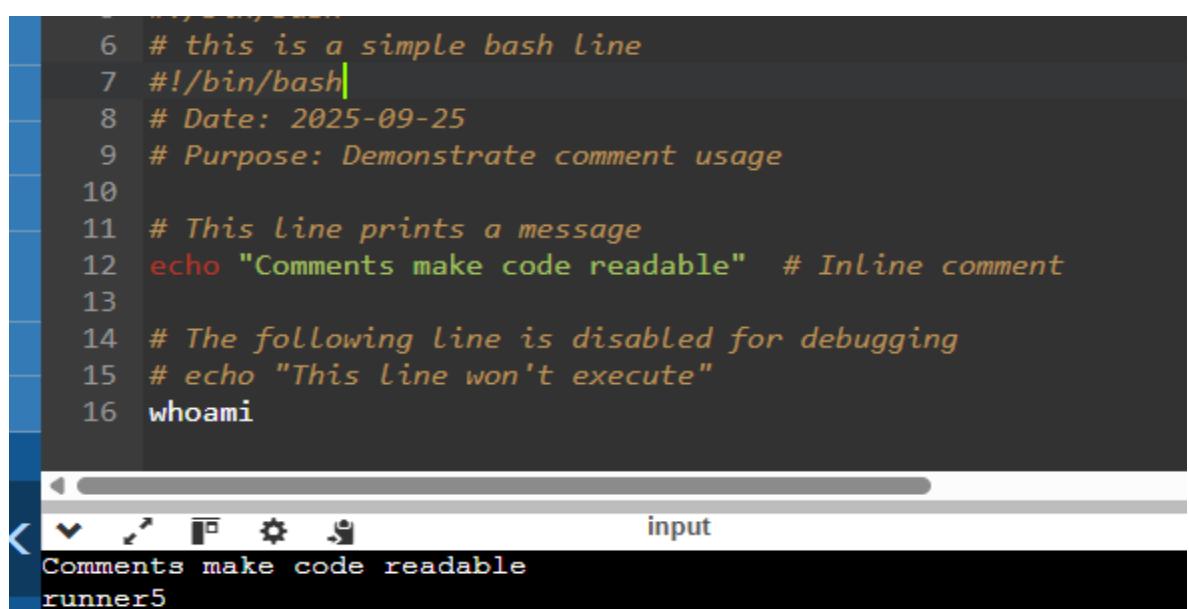
The terminal window shows the script content and its execution. The output includes the welcome message, the script name, and the user's login information.

3. What is the purpose of comments (#) in a shell script? (CO4)

Ans: Comments are the set of statements which gives us information about the task we as an individual or the team as a whole.

PURPOSE:

- Documentation: Gives idea of what is happening in code.
- Code Organization: Breaking code into chunks for better readability.
- Debugging: Easy to debug by commenting out the unnecessary lines
- Maintenance: Easy to maintain when working as an organization.

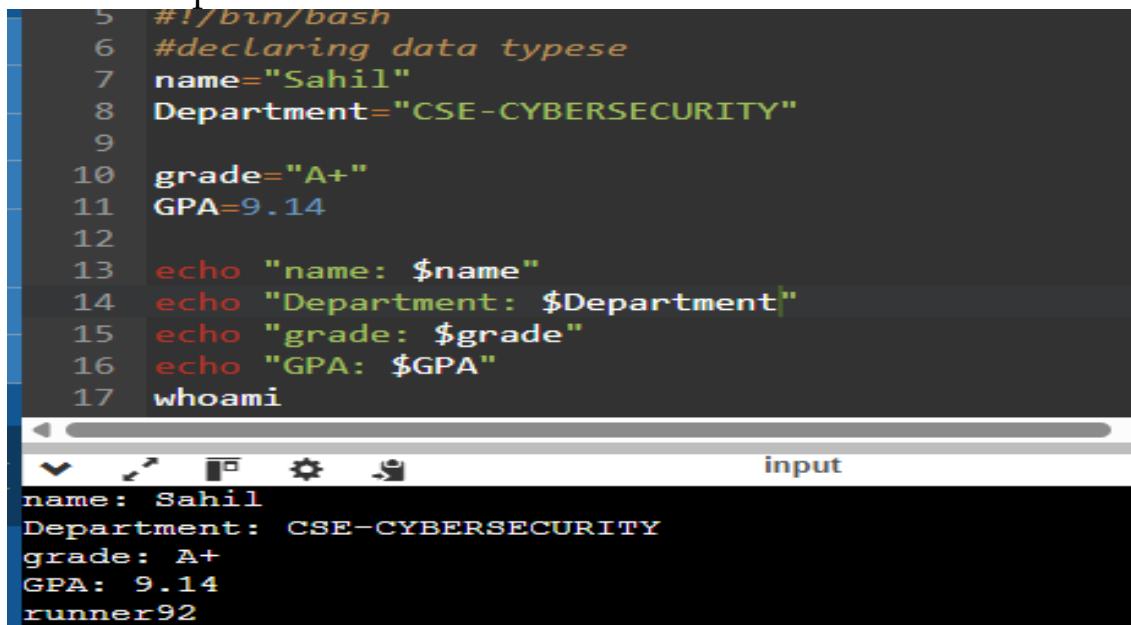


```
6 # this is a simple bash line
7 #!/bin/bash
8 # Date: 2025-09-25
9 # Purpose: Demonstrate comment usage
10
11 # This line prints a message
12 echo "Comments make code readable" # Inline comment
13
14 # The following line is disabled for debugging
15 # echo "This line won't execute"
16 whoami
```

The terminal window shows the script content and its execution. The output includes the welcome message and the user's login information.

4. How do you declare variables (**int, float, double, string, Boolean, and char** in a shell script? (CO4)

Ans: In bash scripting , variables are declared without explicit type declaration. The same variable in bash can hold different types of data. Example:

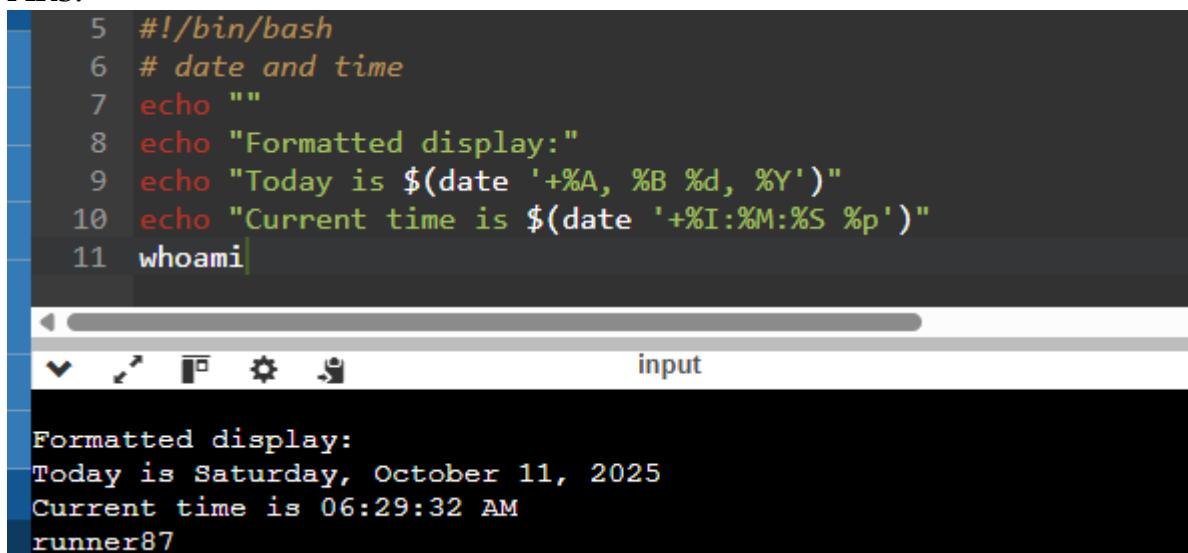


```
5 #!/bin/bash
6 #declaring data types
7 name="Sahil"
8 Department="CSE-CYBERSECURITY"
9
10 grade="A+"
11 GPA=9.14
12
13 echo "name: $name"
14 echo "Department: $Department"
15 echo "grade: $grade"
16 echo "GPA: $GPA"
17 whoami
```

name: Sahil
Department: CSE-CYBERSECURITY
grade: A+
GPA: 9.14
runner92

5. Write a shell script to **display the current date and time** of the system. (CO4)

Ans:



```
5 #!/bin/bash
6 # date and time
7 echo ""
8 echo "Formatted display:"
9 echo "Today is $(date '+%A, %B %d, %Y')"
10 echo "Current time is $(date '+%I:%M:%S %p')"
11 whoami|
```

Formatted display:
Today is Saturday, October 11, 2025
Current time is 06:29:32 AM
runner87

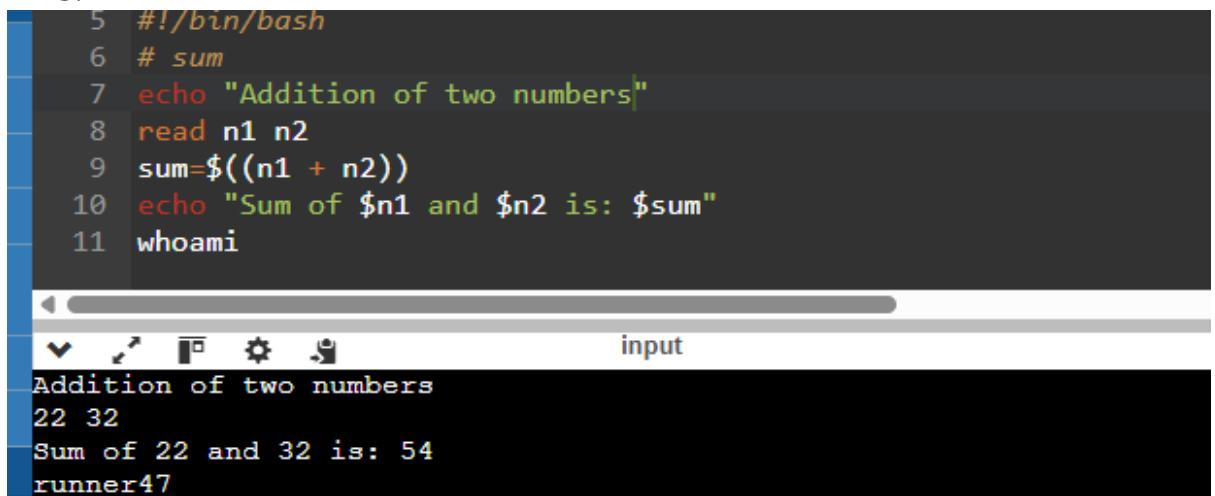
6. Explain the difference between a constant and a variable in bash script. (CO4)

Ans:

| Aspect | Variable | Constant |
|-------------|------------------------|---------------------------------|
| Mutability | Can be changed anytime | Remains unchanged once declared |
| Purpose | Store the varying data | Store set of fix values |
| Declaration | variable_name=value | Declare -r |
| | | |

7. Write a shell script to **read two integer number from the user** and compute the sum of both the number. (CO4)

Ans:



The screenshot shows a terminal window with a dark background. At the top, there is a code editor pane containing a bash script. The script starts with a shebang (#!/bin/bash), followed by a comment (# sum), an echo command (echo "Addition of two numbers"), a read command (read n1 n2), a calculation assignment (sum=\$((n1 + n2))), an echo command (echo "Sum of \$n1 and \$n2 is: \$sum"), and a whoami command (whoami). Below the code editor is a terminal pane with a light gray background. It displays the output of the script: "Addition of two numbers", followed by the user input "22 32", and then the calculated sum "Sum of 22 and 32 is: 54". The bottom right corner of the terminal pane says "runner47".

```
5 #!/bin/bash
6 # sum
7 echo "Addition of two numbers"
8 read n1 n2
9 sum=$((n1 + n2))
10 echo "Sum of $n1 and $n2 is: $sum"
11 whoami
```

8. What is the use of **source** command in shell scripting? (CO4)

Ans: The source command in bash shell scripting is responsible for executing commands from a file in the current shell environment, instead of creating a new subshell.

Important Uses:

- Importing functions from other scripts.

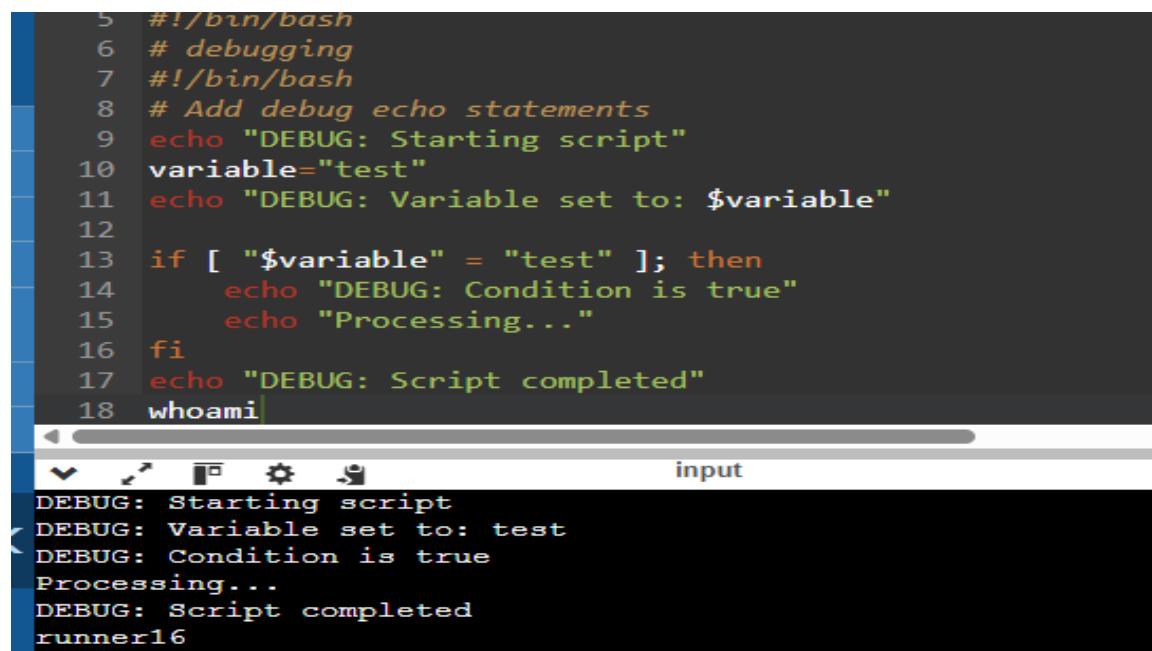
- b. Execution of scripts that modify the current environment
 - c. Load configuration files.
9. How can you **debug** a shell script? Give two methods. (CO4)

Ans: Two methods to debug a shell script is:

- a. Using -x option: Create a file name “example.sh”

```
$bash -x example.sh - Execution time debugging
```

- b. Using echo statements and DEBUG function:



```
5  #!/bin/bash
6  # debugging
7  #!/bin/bash
8  # Add debug echo statements
9  echo "DEBUG: Starting script"
10 variable="test"
11 echo "DEBUG: Variable set to: $variable"
12
13 if [ "$variable" = "test" ]; then
14     echo "DEBUG: Condition is true"
15     echo "Processing..."
16 fi
17 echo "DEBUG: Script completed"
18 whoami
```

input

```
DEBUG: Starting script
DEBUG: Variable set to: test
DEBUG: Condition is true
Processing...
DEBUG: Script completed
runner16
```

10. Write a bash script to **create and delete a file**. (CO4)

Ans:

```
6 # Simple file create and delete
7
8 filename="example10.txt"
9
10 # Create file
11 echo "Creating file..."
12 echo "Linux Assignment!" > "$filename"
13 echo "File created: $filename"
14
15 # Show file exists
16 ls -l "$filename"
17
18 # Delete file
19 echo "Deleting file..."
20 rm "$filename"
21 echo "File deleted: $filename"
22
23 # Verify deletion
24 if [ ! -f "$filename" ]; then
25     echo "File successfully removed"
26 fi
27 whoami
```

```
input
Creating file...
File created: example10.txt
-rw-r--r-- 1 runner50 runner50 18 Oct 11 06:53 example10.txt
Deleting file...
File deleted: example10.txt
File successfully removed
runner50
```

NOTE: All the above images are tested on onlineGDB compiler