

Write a program

A) to show the use of echo .

B) to read the keywords in shell programming.

Student Name: Parikshit sharma

Branch: CSE(IOT)

Semester: 3rd

Subject Name OS LAB

UID: 19BCS4520

Section/Group IOT/A

Date of Performance:

Subject Code: CSP-210

Aim/Overview of the practical:

A) to show the use of echo .

B) to read the keywords in shell programming.

Task to be done:

Echo is commonly used in shell scripts to display a message or output the results of other commands.

As most of you know, we can define variables in our shell scripts to make our scripts more independent of hard coding. Below are the ways we can assign values to variables in a shell script

Commands & Explanations

1 The shell will substitute all variables, wildcard matching, and special characters before passing the arguments to the echo command.

2 Although not necessary, it is a good programming practice to enclose the arguments passed to echo in double or single quotes.

3 When using single quotes " " the literal value of each character enclosed within the quotes will be preserved. Variables and commands will not be expanded

Use the `$command` expression to include the command output in the `echo`'s argument. The following command will display the

Use `ANSI escape sequence` to change the foreground and background colors or set text properties like underscore and bold

Reads the data until you press the enter key. Here, we won't specify the variable where to store the data. By default the data is stored in a variable called **REPLY**. So we can invoke the data as **\$REPLY**

`read -p "prompt_text"`

Here we read the data along with some hint text . The hint text helps the user in what he/she has to enter . **-p** here stands for the prompt . The hint text also called the prompt text.

`read variable`

The variable is the place holder for the text you enter. The whole unit of text is stored in variable that you provide.

`read -n`

This **-n** option allows the user to enter only the specific length of characters. It won't allow you to enter more than the given number of characters.

read -s

This option **-s** means secret or secure whatever is used to read the sensitive data. Generally, when you type entering the data it appears in the terminal.

Output: Image of sample output to be attached here

```
parikshit@kali:~$ echo hello world
hello world
parikshit@kali:~$ echo parikshit sharma
parikshit sharma
parikshit@kali:~$ myvar="mark"
parikshit@kali:~$ echo $
$
parikshit@kali:~$ echo $myvar
mark
parikshit@kali:~$ x=10
parikshit@kali:~$ echo "the value of x is $x"
the value of x is 10
parikshit@kali:~$ echo -e 'some\text'
some    ext
parikshit@kali:~$ echo -e 'some \n  text'
some
    text
parikshit@kali:~$ echo $USER
parikshit
parikshit@kali:~$ echo "The date is: $(date+%D)"
bash: date+%D: command not found
The date is:
parikshit@kali:~$ echo "The date is: $(date +%D)"
The date is: 09/26/20
parikshit@kali:~$ echo -e "\003[1;37mWHITE"
[1;37mWHITE
parikshit@kali:~$ █
```

```
parikshit@kali:~$ echo "hellow world"
hellow world
parikshit@kali:~$ read
parikshit
parikshit@kali:~$ echo "hello";echo"worls"
hello
bash: echoworls: command not found
parikshit@kali:~$ echo "hello"; echo "World"
hello
World
parikshit@kali:~$ echo "what is your name?"; read name
what is your name?
parikshit sharma
parikshit@kali:~$ echo "what is your name?"read name; echo "hello $name"
what is your name?read name
hello parikshit sharma
```

```
parikshit@kali:~$ read
the information
parikshit@kali:~$ echo $REPLY
the information
parikshit@kali:~$ read -p "ENTER YOUR NAME: "
ENTER YOUR NAME: Parikshit
parikshit@kali:~$ echo "MY NAME IS $REPLY"
MY NAME IS Parikshit
parikshit@kali:~$ read name
parikshit sharma
parikshit@kali:~$ echo "my name is $name"
my name is parikshit sharma
parikshit@kali:~$ read -n 5 -p "enter 5 character only: "
enter 5 character only: 12345parikshit@kali:~$
parikshit@kali:~$ echo $REPLY
12345
parikshit@kali:~$ read -p "enter username: " username
enter username: parikshit
parikshit@kali:~$ read -s -p "enter the password: " password
enter the password: parikshit@kali:~$
parikshit@kali:~$ echo "username: $username password: $password"
username: parikshit password: 1234
parikshit@kali:~$ █
```

Learning outcomes (What I have learnt):

1. Use of echo

2. To read the keyword in shell programming

3. display output in terminal

4 linux programing

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			