**WEEK 1:-**

1. List the name of the users logged in and their total count without displaying any other detail

Ans- $who –q

1. Find out your terminals device name.

Ans- $tty

1. Display current date in the form of dd/mm/yyyy.

Ans- $date +%d/%m/%Y

1. Find out your machines name and version of the operating system.

Ans- $uname –nr

1. Clear the screen and place the cursor at row 12, column 25

Ans- $clear $ tput cup 12 25

1. Find the decimal equivalent of 1101001.

Ans- $bc ibase=2 ...

1. Find out the users who are idling.

Ans- $who -Hu

1. Use man to get help

Ans- $man tty

1. Ensure that bc displays the results of all divisions using three decimal places.

Ans- $ bc Scale=3

**WEEK 2:-**

1. Create a directory in ur home directory(cse, two subdirectories cprogs and projestcs under cse)while being in ur home directory

Ans- mkdir cse, cd cse, mkdir cprogs, mkdir projects

1. Change to the directory projects

Ans- cd cse, cd projects

1. Create a file called biodata and store ur name, age, sex, address in it.

Ans- cat > biodata.txt, info

1. Make a copy of the file biodata into another file text within the directory cprogs

Ans- cd .. , cp biodata.txt text.txt

1. Move the file text from cprogs to projects

Ans- mv text.txt / cse / projects

1. Combine the contents of the file biodata and text into another file datatext

Ans- cat biodata.txt text.txt > datatext.txt

1. Rename the file text to newtext

Ans- mv text.txt newtext.txt

1. Change the permissions of the file nextext to rw- rw-rw-

Ans- chmod 666

1. List all filenames starting with ‘a’ or ‘b’ or ‘m’

Ans- Find .typeF –exec grep –l “a”,”b”,”m” {} \

1. List all filenames that end with a digit

Ans- Find ./typeF –name ‘\*[:0-9] [0-9].txt’

1. List all files in the current directory whose second character is a digit

Ans- ls –d-- ? [0-9] \*

1. Use commands to create a directory in ur home directory called keepout whose contents can be read only be you

Ans- cd .. , cd .. , mkdir keepout, chmod 711

**WEEK 3:**

1. List all files beginning with character ‘a’ on the screen and also store them in a file called file1.

Ans- ls [a] \* | tee file1

1. Sort the output of who and display on screen along with total number of users. The same output except the number of users should be stored in a file file1.

Ans- who –q | sort, who –Hu | cat >> file1

1. Double space a file

Ans- pr –d file1

1. Select lines 5 to 10 of a file

Ans- head -10 file1 | tail –n -5

1. Find the user name and group id from the file /etc/passwd using the cut command.

Ans- cut –d “:” –f 1,4 /etc/passwd

1. Extract the names of the users from /etc/passwd after ignoring the first 10 entries.

Ans- cut –d “:”-f 1 /etc/passwd | tail –n +11

1. Sort the file /etc/passwd on GUID (primary) and UID (secondary) so that the users with the same GUID are placed together. User with a lower UID should be placed higher in the list.

Ans- cut –d “:” –f 3,4 /etc/passwd | sort –n

1. List from /etc/passwd the UID and the user having the highest UID.

Ans- sort –t “:” –r –n –k 3 /etc/passwd | cut –d “:” –f 1,3 | head -1

1. Device a sequence which lists the five largest files in the current directory

Ans- ls –lS | head -6

1. Remove duplicate lines from a file.

Ans- uniq file1

1. Count the frequency of occurrences of words in a file.

Ans- sort file1 | uniq –c

1. Find "long" listing of the smallest 5 files in the /etc directory whose name contains the string ".conf", sorted by increasing file size.

Ans- ls –lSr /etc/\*.conf | head -5

1. What would you type at the command line to get a sorted list, with no duplicates, of all the users logged into the local network?

Ans- who | uniq | sort

1. What would you type at the command line to find all files in your home directory that are more than a week old and end with .bak?

Ans- find –mtime +7 –name “\*.bak” –ls

1. What would you type at the command line to find out how many total lines are contained in all the files ending in .c in the current directory, printing only the total number of lines?

Ans- wc –l \*.c

**WEEK 4:**

1. Find out the PID of your login shell.

Ans- ps

1. Remove the header line from the ps output.

Ans- ps --no-headers

1. List all processes that you are currently running on your machine, sorted by the command name in alphabetical order. The output should consist only of the processes you are running and nothing else (i.e. if you are running 6 processes, the output should only have 6 lines).

Ans- ps aux –sort + COMMAND

1. Display the files in the current directory that contain the string CSE UEMK in either upper- or lowercase.

Ans- grep –il ‘CSEUEMK’ \*

1. Store in a variable the number of lines containing the word CSE in the files CSE1, CSE2 and CSE3.

Ans- var= ‘grep CSE cse[1-3] | we –l’, echo, var

1. If you did not have the wc command, how would you use grep to count the number of users currently using the system?

Ans- who | grep –c “.\*”

1. Remove blank lines from a file using grep.

Ans- grep –v “^$” aa1

1. List the ordinary files in your current directory that are not writable by the owner.

Ans- ls –l | grep –v ‘^..w’

1. Locate lines ending and beginning with a dot and containing anything between them.

Ans- grep ‘^\..\*\.$’ mca4

1. Locate lines that are less than 100 characters in length.

Ans- grep ‘^.{0,99\}$’ \*file1

**WEEK 5:**

1. Write a shell script to find out whether an integer input through the keyboard is an odd number or an even number.

Ans- clear, echo”Enter a number:”, read n, echo”Result:”, if [exper $n % 2’ ==0], then, echo”$n is even”, else, echo”$n is odd”, fi

1. Write a shell script to find out whether any year input through the keyboard is a leap year or not. If no argument is supplied the current year should be assumed.

Ans- clear, echo”Enter a year:”, read year, if [‘exper $year % 4’ –eq 0], then, echo”$year is a leap year”, else, echo”$year is not a leap year”, fi

1. Write a shell script to find the maximum of three numbers provided as command line arguments.

Ans- echo”Enter number1:”, read n1, echo”Enter number2:”, read n2, echo”Enter number3:”, read n3, if [$n1 –gt $n2] && [$n1 –gt $n3], then, echo $n1, elif [$n2 –gt $n1] && [$n2 –gt $n3], then, cho $n2, else, echo $n3, fi

1. Write a shell script to check whether a given number is prime or not

Ans- echo”Enter a number:”, read n, i = 2, Flag = 0, while test $i –le ‘exper $n/2’, do, if test ‘exper $n % $i’ –eq 0, then, Flag= 1, fi, i= ‘exper $i + 1’, done if test $Flag –eq1, then, echo”The number is not prime”, else, echo”The number is prime”, fi

1. Write a shell script to find the factorial value of any integer entered through the keyboard.

Ans- echo”Enter a number”, read num, fact = 1, for((i=2; i <= num; i++)){ fact = $((fact \* i)) }, echo $fact

1. Write a shell script to generate all combinations of 1, 2 and 3

Ans- clear, for i in 1 2 3, do, for j in 1 2 3, do, for k in 1 2 3, do, echo $i $j $k, done, done, done

1. Write a shell script to print all prime numbers in a given range.

Ans- echo enter m and n

read m n

**for** a **in** $(seq **$m** **$n**)

**do**

**k**=0

**for** i **in** $(seq 2 $(expr **$a** - 1))

**do**

**if** [ $(expr **$a** % **$i**) -**eq** 0 ]

**then**

**k**=1

**break**

**fi**

**done**

**if** [ **$k** -**eq** 0 ]

**then**

echo **$a**

**fi**

**done**

1. Write a shell script to calculate the sum of digits of any number entered through keyboard.

Ans- **echo -n "Enter number : "  
read n  
# store single digit  
sd=0  
# store number of digit  
sum=0  
# use while loop to caclulate the sum of all digits  
while [ $n -gt 0 ]  
do  
    sd=$(( $n % 10 )) # get Remainder  
    n=$(( $n / 10 ))  # get next digit  
    sum=$(( $sum + $sd )) # calculate sum of digit  
done  
echo  "Sum of all digit  is $sum"**

**WEEK 6:**

1. Write a shell program that takes a number from user and prints the reverse of the number.

Ans- read -p "Enter a number: " number

temp=$number

while [ $temp -ne 0 ]

do

    reverse=$reverse$((temp%10))

    temp=$((temp/10))

done

echo "Reverse of $number is $reverse"

1. Write a shell script to determine whether two numbers input through keyboard are prime to each other.

Ans- echo”Enter number1:”, read n1, echo”Enter number2, read n2, i = 2, Flag = 0, while test $i –le ‘exper $n/2’, do, if test ‘exper $n % $i’ –eq 0, then, Flag= 1, fi, i= ‘exper $i + 1’, done if test $Flag –eq1, then, echo”The number is not prime”, else, echo”The number is prime”, fi

1. Write a shell script to find whether a number is divisible by 11.

Ans- **echo "Enter any Number"**

**read n**

**r=`expr $y % 11`**

**if [ $r -eq 0 ]**[**first post**](mailto:govsdm@gmail.com)

**then**

**echo $r " is divisible by 11"**

**else**

**echo $r " is not divisible by 11"**

**fi**

1. Write a shell script that produces a shell calculator to perform the following operations: Addition• Subtraction• Multiplication• Division•

Ans- echo "Enter Two numbers : "

read a

read b

# Input type of operation

echo "Enter Choice :"

echo "1. Addition"

echo "2. Subtraction"

echo "3. Multiplication"

echo "4. Division"

read ch

# Switch Case to perform

# calculator operations

case $ch in

  1)res=`echo $a + $b | bc`

  ;;

  2)res=`echo $a - $b | bc`

  ;;

  3)res=`echo $a \\* $b | bc`

  ;;

  4)res=`echo "scale=2; $a / $b" | bc`

  ;;

esac

echo "Result : $res"

1. Write a shell script to print the following pattern for any number of lines:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

Ans- rows=5

for((i=1; i<=rows; i++))

do

  for((j=1; j<=rows - i; j++))

  do

    echo -n "  "

  done

  for((j=1; j<=2\*i - 1; j++))

  do

    echo -n "\* "

  done

  echo

done

1. Write a shell script to test whether a given string is palindrome or not.

Ans- echo "Enter the string "

read str

**for** i **in** $**(**seq 0 ${#str}) ; do

revstr=$**{**str:$i:1**}**$revstr

done

echo "The given string is " $str

echo "Its reverse is " $revstr

**if** **[** "$str" = "$revstr" **]**; **then**

echo "It is a palindrome."

**else**

echo "It is not a palindrome."

fi

1. Write a shell script which counts the number of consonants and vowels in a given sentence.

Ans- #!/bin/sh

echo -n "Enter a line of text: "

read string

numCount=$(echo $string | grep -o "[0-9]" | wc --lines)

vowCount=$(echo $string | grep -o -i "[aeiou]" | wc --lines)

consCount=$(echo $string | grep -o -i "[bcdfghjklmnpqrstvwxyz]" | wc --lines)

echo "The given string has $vowCount vowels, $consCount consonants and $numCount numbers in it."

1. Write a shell script to display the list of users as well as the number of users connected to the system.

Ans- #!/bin/bash

#here we are you going to develope a script for various options on user accounts

echo -e "\n

[ 1 ] for listing all the user accounts name \n

[ 2 ] for counting the number of logged-in user accounts \n

[ 3 ] for listing the names of currently logged-in users\n

[ 4 ] for checking the groups to which the current user belong \n"

#Now take user input

read userInput

#Now we will use switch cases for various input operations

case $userInput in

1)

#syntax lslogins <option[=output field]>

lslogins -o USER

;;

2)

#syntax who <option> <user optional>

#grep used to filter

who --count | grep users

;;

3)

#-q option is to count the number of users and print the logged-in users.

# instead of -q, --count can also be used.

# -v is used to exclude any pattern

who -q | grep -v users

;;

4)

#syntax groups <option> [USERNAME]

groups

;;

\*)

echo -e "Please Enter Correct Input \n"

;;

Esac

**WEEK 7:**

1. Write a shell script that displays a list of all files in the current directory to which you have read, write and execute permissions

Ans- **for var in \***

**do**

**if test -r $var -a -w $var -a -x $var -a ! -d $var**

**then**

**ls $var**

**fi done**

1. Write a shell script that lists files by modification time when called with *lm and by access time when called with la. By default, the script should show the listing of all files in the current directory.*

Ans- **case $1 in**

**lm) ls -lt ;;**

**la) ls -lut;;**

**\*) ls -l;;**

**esac**

1. Write a shell script to display the files created or updated within fourteen days from the current date

Ans- **find -atime -14 -mtime -14 | sort –u**

1. Develop a shell script which displays all files with all attributes those have been created or modified in the month of November

Ans- **for var in \***

**do**

**set -- `ls -l $var` if test “$6” = “Nov”**

**then**

**ls -l $var**

**fi**

**done**

**WEEK 8:**

1. Write a shell script, which reports names and sizes of all files in a directory (directory should be supplied as an argument to the shell script) whose size exceeds 100 bytes. The filenames should be printed in decreasing order of their sizes. The total number of such files should also be reported.

Ans- **if test $# -ne 1**

**then**

**echo “Please give a directory name and try again”**

**exit**

**fi**

**cd $1 find -size +100b | sort –nr**

**echo “Total number of such files:”**

**find -size +100b | grep -c “.\*”**

1. Write a shell script that shows the names of all the non-directory files in the current directory and calculates the sum of the size of them.

Ans- filepath=” /home/deb/Downloads/game.zip”, size= $(find $filepath –printf % s), echo”The size of file is $size Bytes”

1. Write a shell script to list the name of files under the current directory that starts with a vowel.

Ans- **echo “Required files are:”**

**ls | grep “^[aeiou]”**

1. Write a shell script which receives two filenames as arguments and checks whether the two file‟s contents are same or not. If they are same then the second file should be deleted.

Ans- **if test $# -ne 2**

**then**

**echo “Please give two filenames.”**

**exit**

**fi**

**cmp -s $1 $2**

**if test $? -eq 0**

**then echo “$1 and $2 are same”**

**rm $2**

**else**

**echo “$1 and $2 are not same”**

**fi**

**WEEK 9:**

1. A file called list consists of several words. Write a shell script which will receive a list of filenames, the first of which would be list. The shell script should report all occurrences of each word in list in the rest of the files supplied as arguments.

Ans- #!/bin/sh, if [$# -lq 0], then, echo”no arguments”, else, for “ “ “, “ <$1> temp, shift, for i in $\*, do, -lr “ “ “, “ <$i> temp1, y= “ we –l < temp”, j=1, while [$j –le $y], do, x= ‘head –n $j temp | tail -1’, c= ‘grep –c “$x” temp-1’, echo $x $c, j = ‘expr $j1’, done, done, fi

1. Write a shell script which deletes all lines containing the word *UNIX in the files supplied as arguments to this shell script.*

Ans-

