week 1 program :

1) list all the attributes of ls command

2) which cmd is used to count words & characters.

3) list the different types of files.

4) Difference b/w cut & cat.

5) use of echo

6) Difference b/w $#, $@, $*.

7) How to set different file permission.

8) How to access process id

9) Differentiate b/w pid & ppid.

10) write a shell script to print the sum of all digits.

11) write a shell script to display current date, time, username & current directory.

12) write a shell script to add two command line args & display err if necessary.

13) write a shell script to print square of a given number

14) write a shell script to accept two file names & display the content of it & print their permission'

15) write syntax for if, while, for & also write simple pgm to demonstrate them.

Answers :

1) ls -l

ex: ls -l

```
drwxr-xr-x 17 akrasrotti staff 544 Feb 16 11:52 myApp
```

The attributes are :

i) File mode :- drwxrwxrwx

ii) Number of links

iii) Owner name

iv) group name

v) Number of bytes in file

vi) Date & Time of last modification

vii) Pathname

2) Word count is used

```
wc -w : word count

  wc -w word.txt
  12 word.txt

wc -m : Character count

  wc -m word.txt
    67 word.txt
```

3) i) Regular

ii) Directory

iii) Symbolic Link

iv) Socket

v) Device File.

4) Cut is used to cut out selected portions of each line of a file

Cat is used to concatenate & print files.

5) echo is used to write to the standard output.

PTO->

6) $#: Number of arguments

$*: Gives all arguments as a single string

$@: Gives all arguments as an array

7) File permission can be set using Chmod Command

ex: chmod 755 ~~file.txt~~ file.txt

It can be:

i) absolute

ii) Relative

ex: Chmod +X file.txt

8) Using ps command

ex:

```
PID    TTY      TIME     CMD
24958  ttys000  0:00.12  -zsh
```

9) PID stands for process ID, which means identification number for currently running process in memory.

PPID stands for parent process ID, which is the PID of the parent process responsible for creating the current process.

10)

```
#!/bim/bash
# Program to print sum of all digits
echo -n "Enter the number:"
read num
sum=0
while (($num > 0))
do
    trailing=$(($num %10))
    sum=$(($sum + $trailing))
    num=$(($num/10))
done
echo "The sum is $sum"
```

```
11) #! /bim/bash
 # Program to Display current date, time, username, current directory
  currdate = `date | cut -d ' ' -f 2-3`
  currtime = `date | cut -d ' ' -f 5`
   echo "Current Date is $ currdate"
   echo "current Time is $ currtime"
   echo "Username is `whoami`"
   echo "Present directory is `pwd`"

12) #! /bim/bash
  # Program to add two command line arguments
   if ((  $#==2))
   then
       sum= $(( $1+$2))
       echo "The sum is $sum"
   else
       echo "Arguments provided incorrectly"
   fi

13) #! /bim/bash
 # Program to print reverse
 echo -n "Enter the number"
 read num
 rev = 0
 while (($num > 0))
 do
    trailing = $(( $num % 10))
    rev = $(( $rev *10 + $trailing ))
    num = $(( $num/10))
 done
 echo "The Reverse is $ rev"

14) #! /bim/bash
 # Program to display files
 if (($#!=2))
 then
 echo "Arguments provided incorrectly"
 else
```

```
    echo "The contents of $1:"
    echo `cat $1`
    echo -n "The permission of $1:"
    echo `ls -l $1 | cut -d ' ' -f1`
    echo "The contents of $2:"
    echo `cat $2`
    echo -n "The permission of $2:"
    echo `ls -l $2 | cut -d ' ' -f1`
fi
```

15)
```
if [condition]
then
    statements
fi

while [condition]
do
    statements
done

for variable in 1 2 3 4 5 ...N
do
    statements
done
```

osc:
```
#!/bin/bash
#program to demonstrate of while, For
i=1
if (( $i==1))
then echo "i is 1"
fi
i=5
echo "while loop:"
while (( $i >=0))
do
    echo $i
    i=$(($i-1))
done
```

```
echo "For loop:"
for i in 5 4 3 2 1 0
do
    echo $i
done
```