

Practical No. 1

Linux Installation

41

Install your choice of Linux distribution using a USB drive

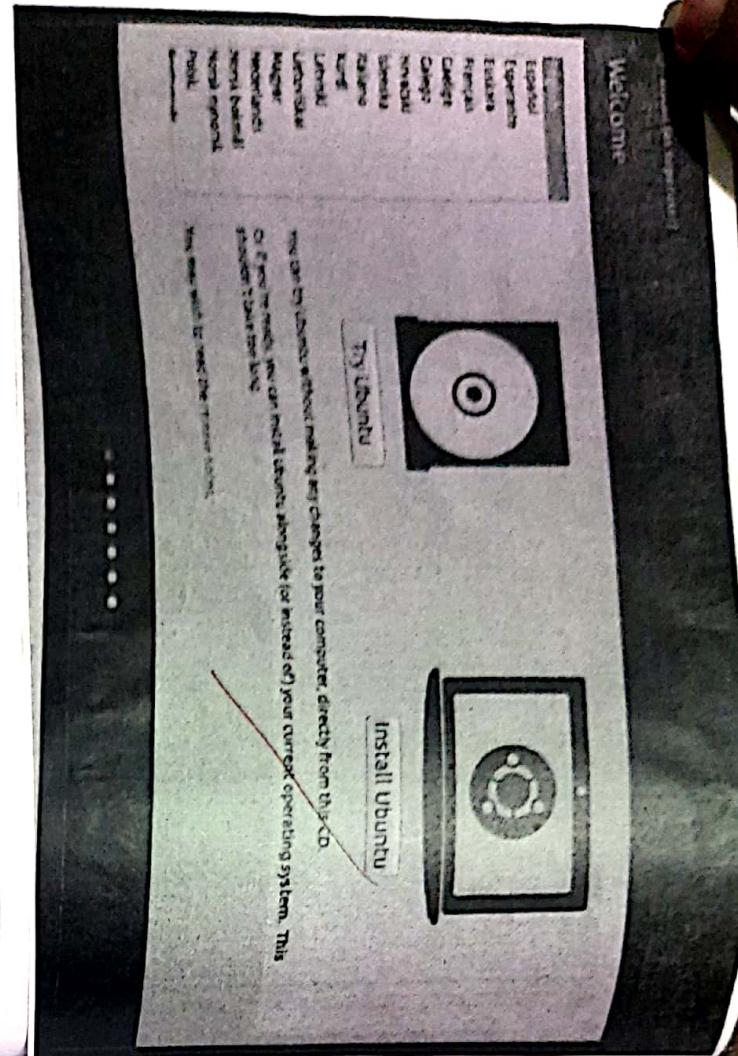
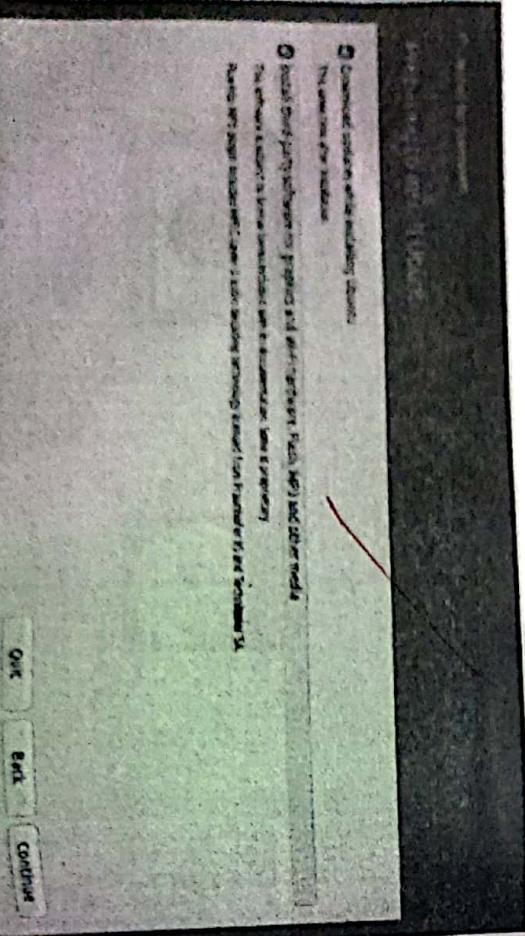
Most newer computers can boot from USB. You should see a welcome screen prompting you to choose your language and giving you the option to install Ubuntu on the USB.

If your computer doesn't automatically do so, you might need to press the F12 key to bring up the boot menu, but be careful not to hold it down that can cause an error message.

Prepare to Install Ubuntu.

We recommend you plug your computer into a power source. You should always make sure you have enough space on your computer to install Ubuntu. We advise you to select Download updates while installing and Install this third-party software now.

You should also stay connected to the internet so you can get the latest updates while you install Ubuntu.



If you are not connected to the internet, you will be asked to select a wireless network if available. We advise you to connect during the installation so we can ensure your machine is up to date.

2. Allocate drive space. Use the checkboxes to choose whether you had like to install Ubuntu alongside another operating system, delete your existing operating system and replace it with Ubuntu or if you have an advanced user choose the 'something else' option.

3. Begin the Installation:

- Depending on your previous selection, you can now verify that you have chosen the way in which you would like to install Ubuntu. The installation process will begin when you click the 'Install Now' button. Ubuntu needs about 4.5GB to install so add a few extra GB to allow for your files.

Install (xserver-xorg)
Where are you?

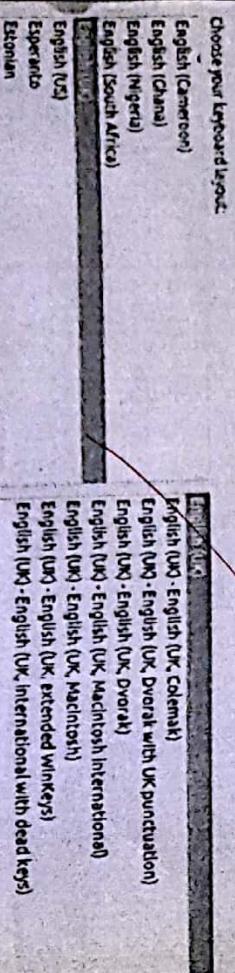


4. Select your location.

If you are connected to Internet be sure automatically to Internet correct and click 'Forward' to proceed. If you are unsure of your location click on the map and you are in our find it. TIP : If you are having problems connecting to the internet, use the menu in the top right hand corner to select a network.

5. Select your preferred keyboard layout.

~~Click on the language option you need. If you're not sure, click the 'Select Keyboard Layout' button for help.~~



Type here to test your keyboard
Detect Keyboard Layout

Back Continue

53

6. Enter your login password details:

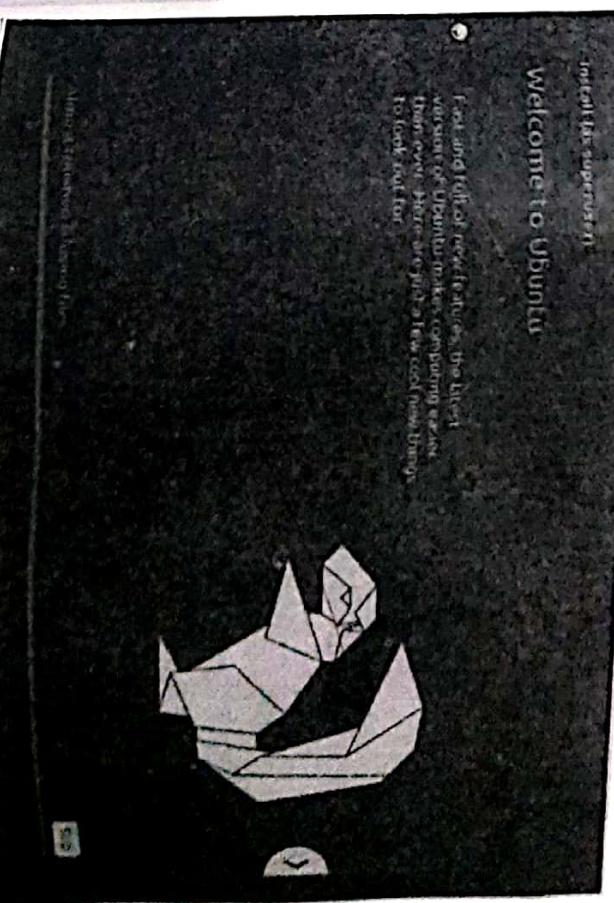
Who are you?

7. Scan issue about ubuntu while the system install

8. That's it.

10/10/27

Installation is complete. You need to restart the computer in order to use the new installation.



54

Your computer's name: Washington
The name you want to use for your account.

Pick a username: bob

Choose a password: Good password!

Confirm your password:

Log in automatically
 Require my password to log in
 Erase my home folder

Practical NO. 2.

Aim: Install gcc package and removing

- a. Install gcc package and then remove it.

step 1: First type 'gcc -v' to know if you have already installed gcc compiler or not. If the output is blank then it means that you don't have installed

step 2: Type 'sudo apt-get install gcc'. After the following command installation will take place.

step 3: Type sudo 'apt-get install build-essential'. This will install all the libraries required for C and C++ programming language.

NOW TO UNINSTALL GCC COMPILER

In GCC 5.1.0, although there is no top-level uninstall target, some directories do have it, in particular gcc so you can do:

Type: cd build / gcc
sudo make uninstall

This does not remove everything that was installed, but it removes major executable like gcc, g++, CPP... contained in that directory.

99
10/02

Practical NO. 3.

Aim: Utilization of grep, man commands

Documentation:

a. finding info documentation from the command line.

bring up the page from the grep command
Bring up the usage section.

Any To find info about any command 'info' is used then syntax of 'info' command is:
`info (command name)`

We are 'grep' to find the info about the 'grep' command

- Open the terminal (`ctrl + A + T`) and type `info grep`.
- After typing command following output will be displayed onto your screen.
- You can also scroll through pages using (`space = up`) & (`Backspace = down`) keys.

Another more summarised form of showing info is the 'man' command. The command is same as 'info', but requires data.

b. finding man pages from the end line:
Bring up the main page for 'ls'
command scroll down the example
section.

To use the 'man' command. Simply type
man (command name). Now we are
going to find the manual for 'ls'
command.

Simply type: 'man ls'

c. finding man pages by topic: what man
pages are available for document

d. finding man pages by section
line: bring up the man page for the
print lib function, which manual page section
are library function.

Simply type: 'man zip'
man ls

The number corresponds to what section of
manual page it is from; 1 is user command
while 8 is system stuff. The man page for
man itself explain it and list the standard
output.

→ Shows all certain items that have different pages in different sections (eg print) as command open in section 3; in case like that you can pass the section no. to the man before the page name to whose which on you need to use 'man -a' to show every matching page in a show.

Practical No. 4.

Command line operation.

a. Install new package on your system.

→ sudo apt-get install (package name)

b. Remove the package installed

→ sudo apt-get remove (package name)

c. Find the password file in / using find command.

- ✓ # find / -name password
- / user / share / doc / misc - ldap - 253 | grep password
- / etc / pam.d / password
- / etc / passwd

find the directory password file under root and one level down.

find / -maxdepth 2 -name password
• /etc / passwd

find the password file under root and 2 level down
✓ # find / -maxdepth 3 -name passwd

→ You can tell what section or term falls in with 'man -k' (equivalent to proper command) It will do substituting matches too, so you even need to use 'term' to limit it.

99

8.

• /usr / bin / password
• /etc / pam.d / password
• /etc / password

find the password file between sub-directories
find / level 2 & u

find - maxdepth 3 -maxdepth 5 -name password

• /usr / bin / password
• /etc / pam.d / password

d. Create a symbolic link to the file y on
found in ~~want~~ step .

ln -s file1 file2

e. Create an empty file example .txt & move
it to /tmp ~~absolutely~~ using relative pathname

touch example.txt
mv example.txt /tmp

f. delete the file moved to /tmp in previous
~~step~~ by absolute method.

rm /tmp /example.txt

g. find the location of ls , ps , bash commands
which ls
ls : /bin / ls | www | share | man | man1 | ls.1 .gz

whereis ps
ps : /bin / ps | www | share | man | man1 | ps.1 .gz

whereis bash
bash : /bin / bash | etc | bash. bashrc | www | share | man | man1 | ps.1 .gz

~~man / man1 / bash.1 .gz~~

~~10/10~~

Practical

۱۵

File Options

1. Exposure mounted film systems on your computer.

Fs	Filesystem	1K-blocks	Used	Available	Use%	Mounted on
Filesystem		494436	0	494436	0%	/dev
udev		102416	3616	98740	4%	/run/udev
tmpfs		7892728	3383372	3326024	51%	/run
/dev/sda1		512076	216	511860	1%	/dev/shm
tmpfs		5120	4	5116	1%	/run/lock
tmpfs		512076	0	512076	0%	/sys/fs/cgroup

```
cat /etc/cgconfig.conf
```

2

What are the different ways of exploring mounted file system on Linux?

3.
Copying text from files.
cp command, mv command.

4. Archiving and backup ~~the~~ work directory using tar, gzip and bzip2 commands.

```
jebas@jeba-VirtualBox:~/jeb$ ls -l
total 0
jebas@jeba-VirtualBox:~/jeb$ touch ss.txt
jeba@jeba-VirtualBox:~/jeb$ mv gg.txt ss.txt
cat: gg.txt: No such file or directory
jeba@jeba-VirtualBox:~/jeb$ cat ss.txt
Linux
jeba@jeba-VirtualBox:~/jeb$ cat dd.txt
dd.txt
```

```

jeba@jeba-VirtualBox:~/jebS ls
aa.txt.gz  bb.txt.gz
jeba@jeba-VirtualBox:~/jebS cat >aa.txt
hello world
^C
jeba@jeba-VirtualBox:~/jebS cat >bb.txt
this is linux
^C
jeba@jeba-VirtualBox:~/jebS diff aa.txt bb.txt
jeba@jeba-VirtualBox:~/jebS cat >bb.txt
this is Linux
^C
jeba@jeba-VirtualBox:~/jebS diff aa.txt bb.txt
jeba@jeba-VirtualBox:~/jebS diff aa.txt bb.txt
jeba@jeba-VirtualBox:~/jebS diff aa.txt bb.txt
< hello world
> this is Linux
jeba@jeba-VirtualBox:~/jebS gzip aa.txt
jeba@jeba-VirtualBox:~/jebS gzip bb.txt
jeba@jeba-VirtualBox:~/jebS diff aa.txt.gz bb.txt.gz
Binary files aa.txt.gz and bb.txt.gz differ

```

5. Use ~~diff~~ command to compare ~~diff~~ of two files.

6. Use patch command to patch a file again.

10/07

```

jeba@jeba-VirtualBox:~/jebS cat >hi.txt
hi
hi
^C
jeba@jeba-VirtualBox:~/jebS cat >hi1.txt
hello
hello
^C
jeba@jeba-VirtualBox:~/jebS diff -u hi.txt hi1.txt >sam.patch
^C
jeba@jeba-VirtualBox:~/jebS patch <sam.patch
patching file hi1.txt
jeba@jeba-VirtualBox:~/jebS cat sam.patch
--- hi1.txt    +22:14:55.463569834 +0530
+++ hi1.txt    @@ -1,3 +1,3 @@
hi
^C
hello
^C
jeba@jeba-VirtualBox:~/jebS

```

Practical No. 6.

Use Environment.

a. Which account you are logged in? How do you find out? & whoami

```
jeba@jeba-VirtualBox:~$ w
20:35:01 up 1 min, 1 user, load average: 0.70, 0.79, 0.38
USER      TTY      FROM             :0
jeba@jeba-VirtualBox:~$ w -s
20:35:14 up 1 min, 1 user, load average: 0.60, 0.77, 0.37
USER      TTY      FROM             :0
jeba@jeba-VirtualBox:~$ w -h
4:38 /sbin/upstart --user
jeba@jeba-VirtualBox:~$ whoami
jeba@jeba-VirtualBox:~$ w -f
20:36:12 up 1 min, 1 user, load average: 0.41, 0.69, 0.37
USER      TTY      LOGIN@    IDLE   JCPU   PCPU
jeba@jeba-VirtualBox:~$ w -t
20:32 5:36 9.08s 0.33s /sbin/upstart --user
```

```
jeba@jeba-VirtualBox:~$ whoami
jeba@jeba-VirtualBox:~$ who
jeba@jeba-VirtualBox:~$ whoami
jeba@jeba-VirtualBox:~$ who -l
jeba@jeba-VirtualBox:~$ who -1
jeba@jeba-VirtualBox:~$ who -t
jeba@jeba-VirtualBox:~$ who -v
jeba@jeba-VirtualBox:~$ who -u
jeba@jeba-VirtualBox:~$ who -w
```

b. Display /etc/shadow file using cat command and understand the importance of shadow file. How it's different than passwd file.

As with the passwd file, each field in the shadow file is also separated with ":" characters, and are as follows:

```
jeba@jeba-virtualBox:~$ sudo cat /etc/shadow
[jeba]:password:0:99999:7:::
[root]:18240:0:99999:7:::
[daemon]:*:16911:0:99999:7:::
[bin]:*:16911:0:99999:7:::
[sys]:*:16911:0:99999:7:::
[games]:*:16911:0:99999:7:::
[man]:*:16911:0:99999:7:::
[lp]:*:16911:0:99999:7:::
[mail]:*:16911:0:99999:7:::
news:::16911:0:99999:7:::
```

- Username : up to 8 characters usually all lowercase. Case-sensitive the username in the /etc/passwd file.
- pass word , 13 character entry (eg. ::) indicates a direct match to required to log in (usually a password is not and a "*" entry (eg. ::+) indicates the account has been disabled.
- The number of days (since Jan 1, 1970) since the password was last changed.
- The number of days before password may be changed (0 indicates it may be changed at any time).
- The number of days after which password must be changed (99999 indicates user can keep his or her password unchanged for many years.)
- The number of days to warn user of an expiring password (7 for a full week)
- The number of days after pass word expires that account is disabled.
- The number of days since Jan 1, 1970 than an account has been disabled.
- A reserved field for possible future use.

Each field in a password entry is separated with colon ":" colon characters, and are as follow:

- Username , up to 8 characters . case-sensitive usually all lowercase.
- An "x" in the password field . password are stored in the " /etc/shadow " file.
- Numeric user Id . This is assigned by " adduser " script . Unix uses the field plus the following group field , to identify which files belong to the user.
- User's home directory . Usually "/home/username" . An user's personal files , web pages , mail forwarding , etc will be stored here.
- User's shell account . Often to "/bin/bash" to provide access to the bash-shell

- c. get your current working directory

Ans

```
jeba@jeba-VirtualBox:~$ sudo cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin:/usr/sbin/nologin
```

- d. Explore different ways of getting command history, how to run previously executed command without typing it.

the history
! line number.

```
jeba@jeba-VirtualBox:~$ history
1 who
2 whoami
3 who -l
4 clear
5 x -s
6 x -h
7 x -f
8 x -r
9 clear
10 cat /etc/shadow
11 sudo cat /etc/shadow
12 clear
13 sudo cat /etc/passwd
14 pwd
15 clear
16 history
jeba@jeba-VirtualBox:~$ !3
who -l
Login    2020-01-15 20:30
jeba@jeba-VirtualBox:~$
```

o.

Practical No. 7
Create, modify, search and navigate a file in editor.

- Creating file.
To create a file, on the terminal type vi followed by filename.
- Modifying the file:
To modify a file, on the vi editor, type 'o'
- Navigate.

Movement in four directions.

Key ↗ ↘ ↙ ↖
Action.

Move cursor up
Move cursor down
Move cursor left
Move cursor right

```
jeba@jeba-VirtualBox:~$ alias m="mkdir new"
jeba@jeba-VirtualBox:~$ m
jeba@jeba-VirtualBox:~$ ls
Desktop  Downloads  Documents  Examples.desktop
Music    Pictures  Templates  Videos
jeba@jeba-VirtualBox:~$
```

- e. Create alias to most commonly used commands
Alias command instruct the shell to replace one string with another string while executing the commands.
The alias label = "command"
~~the alias label = "command"~~

10/102

Legend navigation.

key

Action:

b Moves back to the beginning of the word.

e Moves forward to the end of the word.

w Moves forward to the beginning of the word.

o (zero) Move to first character of line.

s Move to the end of line.

scrolling:

Action:

Ctrl + b scrolls forward.

Ctrl + e scrolls backward.

Ctrl + d scrolls half page.

Ctrl + u scrolls half page backward.

b. Learn all vi commands
replace, highlighted commands like search
(i) Replace syntax:

```
jeb@jeba-VirtualBox ~
Hello
This is my Linux example
Welcome
Welldone
This is vi Editor
Thank you
```

```
:g/my/s/!/our/igc
jeb@jeba-VirtualBox ~
Hello
This is my Linux example
Wellcome
THiLLiNE Vi Editor
Thank you
```

```
jeb@jeba-VirtualBox ~
Hello
This is our Linux example
Welcome
Welldone
This is vi Editor
Thank you
```

(iii) highlight
use set hlsearch

```
jeba@jeba-VirtualBox: ~  
Hello  
This is our Linux example  
Welcome  
Welldone  
This is Vi Editor  
Thank you  
:  
:set hlsearch
```

(iii) show the line number.
use set nu.

```
jeba@jeba-VirtualBox: ~  
1 Hello  
2 This is our Linux example  
3 Welcome  
4 Welldone  
5 This is Vi Editor  
6 Thank you  
:  
:set nu
```

Practical No. 8.

a. Use of sudo to change user privileges to root.

Create an user named user 1

To give some user root privileges edit /etc/sudoers
and using visudo. Enter new lines as highlighted
below.

```
jeba@jeba-VirtualBox:~  
[sudo] password for jeba:  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
jeba@jeba-VirtualBox:~$
```

```
# Please consider adding local content in /etc/sudoers.d/ instead of  
# directly modifying this file.  
# See the man page for details on how to write a sudoers file.  
Defaults env_reset  
Defaults mail_badpass  
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/  
# Host alias specification  
# User alias specification  
# Cmnd alias specification  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
user1  ALL=(ALL:ALL) ALL
```

b. Identify operations that require sudo privileges.

```
jeba@jeba-VirtualBox:~$ su user1  
Password:  
user1@jeba-VirtualBox:~/home/jeba$ mkdir folder1  
mkdir: cannot create directory 'folder1': Permission denied.  
user1@jeba-VirtualBox:~/home/jeba$ sudo mkdir folder1  
[sudo] password for user1:  
user1 is not in the sudoers file. This incident will be reported.
```

c. Modify expiration date for new user using
password aging.

```
jeba@jeba-VirtualBox:~$ sudo chage -i user1  
Last password change : Jan 20, 2020  
Password expires : never  
Password inactive : never  
Account expires : never  
Minimum number of days between password change : 0  
Maximum number of days between password change : 99999  
Number of days of warning before password expires : 7
```

(Signature)

```
jebajeba-VirtualBox:~$ sudo userdel user1
[judo] password for jebajeba-VirtualBox:~$ su user1
No passed entry for user 'user1'
jebajeba-VirtualBox:~$
```

~~user1~~ ~~user1~~ ~~user1~~ ~~user1~~ ~~user1~~

```
jebajeba-VirtualBox:~$ sudo chage -1 user1
jebajeba-VirtualBox:~$ sudo chage -1 user1
Last password change: 25/05/2020 - 1 year(s)
Password expires: never
Password inactive: never
Account expires: never
Number of days between password change: 90
Number of days before password change: 7
Number of days of warning before password change: 7
jebajeba-VirtualBox:~$
```

~~user1~~ ~~user1~~ ~~user1~~ ~~user1~~ ~~user1~~

```
jebajeba-VirtualBox:~$ sudo chage -1 user1
jebajeba-VirtualBox:~$ sudo chage -1 user1
Last password change: 25/05/2020 - 1 year(s)
Last password change: 25/05/2020 - 1 year(s)
Password expires: never
Password expires: never
Account expires: never
Account expires: never
Number of days between password change: 90
Number of days between password change: 90
Number of days before password change: 7
Number of days before password change: 7
jebajeba-VirtualBox:~$
```

```
jeba@jeba-VirtualBox:~$ ifconfig
enp0s3      Link encap:Ethernet HWaddr 08:00:27:0e:6b:69
            inet addr:10.0.2.15 Brdcast:10.0.2.255 Mask:255.255.255.0
              Inet6 addr: fe80::c0c0:53a0%enp0s3 Brdcast:fe80::ff:fe:6b:69/64 Scope:Link
                UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                RX packets:2 errors:0 dropped:0 overruns:0 frame:0
                TX packets:73 errors:0 dropped:0 overruns:0 frame:0
                collisions:0 txqueuelen:1000
                RX bytes:1160 (1.1 kB) TX bytes:8514 (8.5 kB)
```

```
lo          Link encap:Local Loopback
            inet addr:127.0.0.1 Mask:255.0.0.0
              Inet6 addr: ::1/128 scope:Host
                UP LOOPBACK RUNNING MTU:65536 Metric:1
                RX packets:53240 errors:0 dropped:0 overruns:0 frame:0
                TX packets:53240 errors:0 dropped:0 overruns:0 carrier:0
                collisions:0 txqueuelen:1
                RX bytes:4225872 (4.2 MB) TX bytes:4225872 (4.2 MB)
```

```
jeba@jeba-VirtualBox:~$ hostname
jeba-VirtualBox
jeba@jeba-VirtualBox:~$
```

a.

Get IP address of your machine using ifconfig.

83

Practical no. 9.

b. get hostname of your machine.

c.

Using ping to check the network connectivity to remote machines.

d.

Use of dig command.

```
jeba@jeba-VirtualBox:~$ ping www.google.com
PING www.google.com (172.217.31.196) 56(84) bytes of data.
64 bytes from maa03s28-in-f4.1e100.net (172.217.31.196): icmp_seq=1 ttl=54 time=84.8 ms
64 bytes from maa03s28-in-f4.1e100.net (172.217.31.196): icmp_seq=2 ttl=54 time=84.8 ms
64 bytes from maa03s28-in-f4.1e100.net (172.217.31.196): icmp_seq=3 ttl=54 time=87.1 ms
64 bytes from maa03s28-in-f4.1e100.net (172.217.31.196): icmp_seq=4 ttl=54 time=93.5 ms
64 bytes from maa03s28-in-f4.1e100.net (172.217.31.196): icmp_seq=5 ttl=54 time=86.9 ms
64 bytes from maa03s28-in-f4.1e100.net (172.217.31.196): icmp_seq=6 ttl=54 time=98.0 ms
64 bytes from maa03s28-in-f4.1e100.net (172.217.31.196): icmp_seq=7 ttl=54 time=90.9 ms
[1]+  Stopped                  ping www.google.com
jeba@jeba-VirtualBox:~$
```

```
jeba@jeba-VirtualBox:~$ dig www.google.com
; <>> DiG 9.10.3-P4-Ubuntu <>> www.google.com
; global options: +Cmd
; Got answer:
;-->HEADER<- opcode: QUERY, status: NOERROR, id: 52068
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: ; QUESTION SECTION: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
; www.google.com.

; ANSWER SECTION:
www.google.com.          91      IN      A       172.217.166.100
; Query time: 152 msec
; SERVER: 127.0.1.1#53(127.0.1.1)
; WHEN: Mon Jan 20 22:40:06 IST 2020
; MSG SIZE rcvd: 59
```

18.

e. Trouble shooting network using tracert, route command.

```
jeba@jeba-VirtualBox:~$ traceroute www.google.com
traceroute to www.google.com (172.217.166.100), 30 hops max, 60 byte packets
1 10.0.2.2 (10.0.2.2) 0.190 ms 0.143 ms 0.151 ms
2 * *
3 10.0.2.2 (10.0.2.2) 68.568 ms 68.486 ms 68.405 ms
jeba@jeba-VirtualBox:~$
```

```
jeba@jeba-VirtualBox:~$ route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
default         10.0.2.2      0.0.0.0       UG    100    0        0 enp0s3
10.0.2.0        *             255.255.255.0 U        100    0        0 enp0s3
link-local      *             255.255.0.0   U        1000   0        0 enp0s3
jeba@jeba-VirtualBox:~$
```

f. Use of arp command.

```
jeba@jeba-VirtualBox:~$ arp
Address          HWtype  HWaddress           Flags Mask          Iface
10.0.2.2        ether   52:54:00:12:35:02  C            enp0s
```

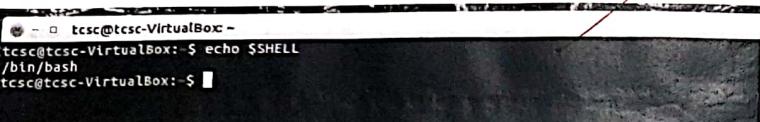
g. Use of host command.

```
jeba@jeba-VirtualBox:~$ host -V
host 9.10.3-P4-Ubuntu
jeba@jeba-VirtualBox:~$
```

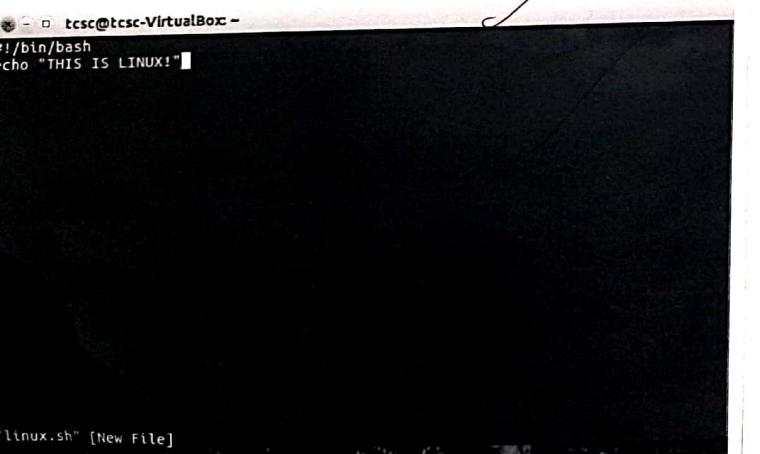
2. Use of netstat command and Nmap command.

```
jeba@jeba-VirtualBox:~$ netstat -an | grep 'Active UNIX' | grep 'DGRAM'  
Proto Recv-Q Local Address           Foreign Address     State  
Active UNIX domain sockets (w/o servers)  
Proto RefCnt Flags       Type      State          I-Node Path  
dgram    [ ]        DGRAM  
unix    :3      [ ]        DGRAM  
syslog  [ ]        DGRAM  
unix    :16     [ ]        DGRAM  
dev.log [ ]        DGRAM  
unix    [ ]        DGRAM  
socket  [ ]        DGRAM  
unix    :3      [ ]        DGRAM  
unix    :3      [ ]        STREAM  CONNECTED  9684  /run/systemd/journal/  
unix    :3      [ ]        STREAM  CONNECTED  44042 /run/systemd/journal/  
unix    :3      [ ]        STREAM  CONNECTED  43113 /run/systemd/journal/  
unix    :3      [ ]        STREAM  CONNECTED  42988 @/tmp/dbus-Cyntei7AOG  
unix    :3      [ ]        STREAM  CONNECTED  42696 @/tmp/dbus-CMGG6G7P5  
stdout  [ ]        STREAM  CONNECTED  13242 /run/systemd/journal/  
stdout  [ ]        STREAM  CONNECTED  43113 /run/systemd/journal/  
unix    :3      [ ]        STREAM  CONNECTED  43013 /run/systemd/journal/  
unix    :3      [ ]        STREAM  CONNECTED  42935
```

```
jeba@jeba-VirtualBox:~$ nmap www.google.com  
Starting Nmap 7.01 ( https://nmap.org ) at 2020-01-20 22:51 IST  
Nmap scan report for www.google.com (216.58.196.68)  
Host is up (0.044s latency).  
Other addresses for www.google.com (not scanned): 2404:6800:4007:811::2004  
FQDN record for 216.58.196.68: bom05s11-in-f4.1e100.net  
Not shown: 998 filtered ports  
PORT      STATE SERVICE  
80/tcp    open  http  
443/tcp   open  https  
Nmap done: 1 IP address (1 host up) scanned in 20.32 seconds  
jeba@jeba-VirtualBox:~$
```



```
tcsc@tcsc-VirtualBox: ~
tcsc@tcsc-VirtualBox: ~$ echo $SHELL
/bin/bash
tcsc@tcsc-VirtualBox: ~
```

```
tcsc@tcsc-VirtualBox: ~
#!/bin/bash
echo "THIS IS LINUX!"
```

linux.sh [New File]

Practical NO.10.

Aim: SHELL SCRIPTING

Basics of shell scripting.

- To get a shell, you need to start a terminal.
- To see what shell you have, run: echo \$SHELL
- In Linux, the dollar sign stands for ~~shell~~ variable.
- The echo command just returns whatever you type in.
- #!/bin/bash - It is called shebang. It is written at the top of a shell script and it passes the instruction to the program /bin/bash.

Echo \$ SHELL

```
vi filename.sh
#!/bin/bash
echo "This is Linux!"
```

chmod 777 filename.sh

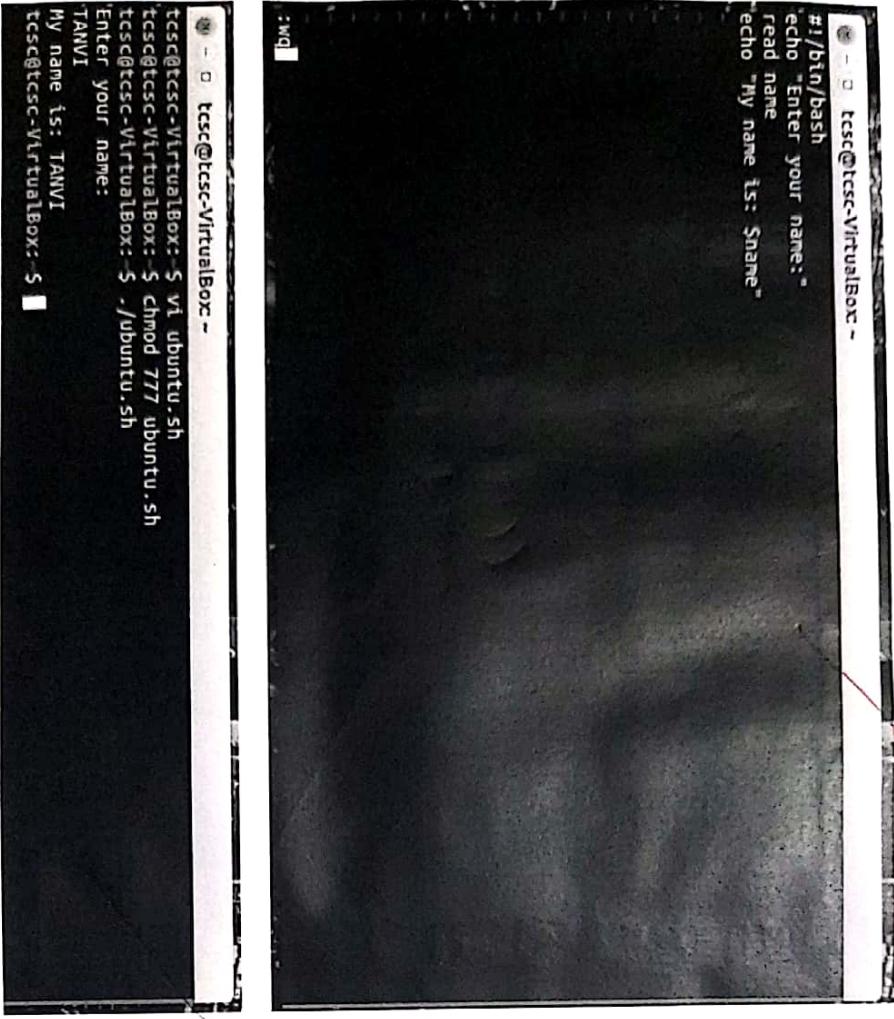
./filename.sh

steps to write and execute a shell script.

Shell script is just a simple text with .sh extension, having executable permission.

- a. Open terminal.
- b. Navigate to the place where you want to create script using cd command.
- c. Touch filename.sh.
- d. Go filename.sh.
- e. chmod 777 filename.sh
- f. sh filename.sh or ./filename.sh.

```
tccs@tccs-VirtualBox:~$ vi linux.sh
tccs@tccs-VirtualBox:~$ chmod 777 linux.sh
THIS IS LINUX!
tccs@tccs-VirtualBox:~$ ./linux.sh
```



```
tcscc@tcscc-VirtualBox:~$ ./ubuntu.sh
#!/bin/bash
echo "Enter your name:"
read name
echo "My name is: $name"
tcscc@tcscc-VirtualBox:~$ ./ubuntu.sh
Enter your name:
TANVI
My name is: TANVI
tcscc@tcscc-VirtualBox:~$
```

Program to display your name.

```
#!/bin/bash
echo "Enter your name"
read name
echo "My name is : $name"
```

Program to find the sum of two variables.

vi filename.sh
#!/bin/bash

a=100
b=25

sum=\$((a+b))
echo "Sum is:\$sum"

```
tcscc@tcscc-VirtualBox:~$ vi llinux2.sh
tcscc@tcscc-VirtualBox:~$ chmod 777 llinux2.sh
tcscc@tcscc-VirtualBox:~$ ./llinux2.sh
Sum is:125
tcscc@tcscc-VirtualBox:~$
```

~~Program
numbers
values
sum
passed during
execution.~~

91

"lin.sh" 3 lines, 46 characters

```
[1] /bin/bash
sum=$((S1+S2))
echo "sum is:$sum"
```

90

```
tcsc@tcsc-VirtualBox:~$ vi lin.sh
tcsc@tcsc-VirtualBox:~$ chmod 777 lin.sh
tcsc@tcsc-VirtualBox:~$ ./lin.sh 50 70
sum is:120
tcsc@tcsc-VirtualBox:~$
```

```
[1] /bin/bash
sum=$((S1+S2))
echo "sum is:$sum"
```

■

■

Sed

Sed command or Stream Editor is very popular utility offered by Linux system. It is mainly used for text substitution find & replace but it can perform other text manipulations like insertion, deletion, search etc. With sed, we can edit complete files without actually having to open it.

1. Displaying partial text of a file.

With sed, we can view only part of a file than seeing whole file.

2. To display all except some lines.

To display all contents of a file except for some portion use option 'd'.

```
tcsc@tcsc-VirtualBox:~$ subjects offered in cs
datastructure
database management
linux
python
green tech
softskill
stats
calculus
computer basic
```

```
tcsc@tcsc-VirtualBox:~$ vi cs.txt
tcsc@tcsc-VirtualBox:~$ sed -n 3,5p cs.txt
database management
linux
python
tcsc@tcsc-VirtualBox:~$
```

```
tcsc@tcsc-VirtualBox:~$ sed 3,5d cs.txt
subjects offered in cs
datastructure
green tech
softskill
stats
calculus
computer basic
tcsc@tcsc-VirtualBox:~$
```

```
tscsc@tscsc-VirtualBox: ~$ vi linux.sh  
tscsc@tscsc-VirtualBox: ~$ chmod 777 linux.sh  
tscsc@tscsc-VirtualBox: ~$ ./linux.sh  
THIS IS LINUX!  
tscsc@tscsc-VirtualBox: ~$
```

```
tscsc@tscsc-VirtualBox: ~$ sed 's/cs/computer/' cs.txt  
subjects offered in computer  
datastructure  
database management  
linux  
python  
green tech  
softskill  
stats  
calculus  
computer basic
```

```
tscsc@tscsc-VirtualBox: ~$ sed '6 s/cs/computer system /' cs.txt  
subjects offered in cs  
datastructure  
database management  
linux  
python  
green tech  
softskill  
stats  
calculus  
computer basic
```

3. To delete a line.

Use line numbers followed by d.

33

Search and replacing a string
's' option is for searching a word.

Replacing a string on a particular line.
Use line number with 's' option.

~~To do my best option "C"~~

~~To do my best option "C"~~

...this is Linux™
subjects offered in CS
classroom lecture
distance management
E-mail
WWW
green "Tech"
softskill
etc

```
tscs@tscs-virtualbox:~$ sed "N;N" /tmp/testfile | less
```

8 Appending lines

To add some line with sed follows.

use * before every line and & in

```
tcsc@tcsc-VirtualBox:~$ sed -e 's/.*/Thanks &/' cs.txt
Thanks subjects offered in cs
Thanks datastructure
Thanks database management
Thanks linux
Thanks python
Thanks green tech
Thanks softskill
Thanks stats
Thanks calculus
Thanks computer basic
```

35

```

file_obj = open("abc.txt", "w") # file open (write mode)
file_obj.write("Computer science subject" + "\n")
file_obj.write("STATS\n Python in DS\n")
file_obj.close()

# File obj = open ("abc.txt", "r")
str1 = file_obj.read()
print ("The output of read method : ", str1)
file_obj.close()

The output of read method : Computer science
# headlines
file_obj = open ("abc.txt", "r")
str3 = file_obj.readlines()
print ("The output of readlines method : " + str3)
file_obj.close()

The output of readlines method : Computer science
subject
STATS
Python
DS

```

27

Practical No. 1.

Objective: Demonstrate the use of different file accessing mode, different attributes and read method.

Step 1: Create a file object using open method and use the write accessing mode.

Step 2: Now open the file in read mode and then use read(), readline(), readlines() and store the output in variable and display content of variable.

Step 3: Now use the file object for finding the names of the file, the file mode in which it is opened.

Step 4: Now open the file object in which write mode. write some content close subsequently.

file object is read mode, display updated contents of file object and close again open parameter in "wt" mode & display the output passed subsequently.

Step 5 : Open the updated file object and close again open parameter in "wt" mode & display the output

Now open file object in append mode & write method write content close the file object again.

```
# file writing
a = file obj.name
print ("Name of file (name attribute) : ", a)
>>> (Name of the file (name of attribute, abc.txt))
b = file obj.closed
print (close) attribute = : False
>>> (close) attribute = : True
c = file obj.mode
print ("file mode", "c")
>>> ("file mode", "w")
d = file obj reference
print ("softspace:", 0)
```

```
# wr mode
file obj = open ("abc.txt", "w")
file obj.write ("Bhavin Dhuven")
file obj.close ()
```

```
# wrk mode
file obj = open ("abc.txt", "w")
file obj.write ("Bhavin Dhuven")
file obj.close ()
```

```
# read mode
file obj = open ("abc.txt", "r")
str1 = file obj.read ()
print ("Output of r", str1)
print ("Output of read mode",
      str1)
```

```
>>> ("Output of read mode",
      str1)
```

```

# append mode
file obj = open ("abc.txt", "a")
file obj.write ("Data structure")
file .close ()
file object = open ("abc.txt", "a")
str3 = file obj.read ()
print ("Output of append mode", str3)
file obj.close ()
>>> ("Output of append mode", "Akiran", "Data structure")

```

Step 7: Open the file object in read mode
 declare a variable & perform file object
 dot file method and store the output
 consequently in variable.

steps : Use the seek method with the argument
 with opening the file object in read
 mode & closing subsequently.

Step 8: Open file object with read mode also
 use the readlines method & close the
 file object consequently in and print
 the same for counting the length

```

tell ()
file obj = open ("abc.txt", "r")
pos = file obj.tell ()
print ("tell()", pos)
file object .close ()
>>> [ "tell () : ", pos]

# seek ()
file obj = open ("abc.txt", "r")
file obj.seek (0, 0)
str4 = file obj.read (1, 0)
str5 = file obj.read (1, 0)
print ("The beginning of the lines is", str4)
print ("The beginning of the lines is", str5)

```

Practical NO. 2.

class odd:

```
def __iter__(self):
    self.num = i
    return self
    def __next__(self):
        if (self.num <= 10):
            num = self.num
            self.num += 2
            return num
        else:
            raise StopIteration
```

1. Write using 'Iterable' object for displaying the odd numbers in range 1 to 10.

Algorithm:

step 1: Define a `__iter__()` with argument and initialize the value & return the value

step 2: Define the `next()` with an argument to compare the upper limit by using a conditional statement.

step 3: Now write an object of the given class and pass the object in the `iter` method.

```

class power {
    public -- open -- (self):
        self = 0
    return self
}

```

```
def next() - (self):
```

```
if self p <= 10
```

```
num = self.p
```

```
self . l += 1
```

```
p0 = 2 ** num
```

```
print ("2 ** l . self p^-1 , = " , p0)
```

```
return p0
```

```
else :
```

```
raise stopIteration.
```

```

p >> power()
x = power()
next()
2 ** 0 = 2
x.next()
2 ** 3 = 8

```

Q2. WAP using an ITR for calculating the power of a given no for instance, number entered is 2 then value calculated should be $1, 2, 2^2, 2^3, 2^4, 2^5$

Algorithm:

Step 1: Define open() with argument and initialise value and return the value.

Step 2: Now define next() with an argument and compare the upper limit with an argument.

Step 3: Now create object of the given class. and pass the object to open() method.

class fact:

def __init__(self):

self.b = 1

return self

def __mult__(self):

if self.b <= 10

num = self.b

self.b += 1

for i in range(1, num+1):

for i=1

print("scanf b.%d", i, fact).

else:

raise an exception.

Step 8: Now create an object of the other class & compare it with an argument in the item() method.

18

32

Ques: WAP for factorial of number in its 10 range.

Ans:

Step 1: Define an item() with argument & initialize the value and return the value.

Step 2: The item() method with an argument and compare the upper limit using a conditional statement.

Step 3: Now create an object of the other class & compare it with an argument in the item() method.

Practical No. 3.

Aim: Demonstration the use of exception handling.

SS
`def acc-age():
 print("Enter your age")
 age = int(input())
 if age > 50 or age < 10:
 raise ValueError`

`else:
 print("Your age is", age)`

`valid = False
while not valid:`

`try:
 age = acc-age()`

`except ValueError:
 print("Your age is not in range")`

Algo:

Theory: An exception is an event which occurs during execution of program which disrupts the normal flow of the program. An exception must be handled immediately, it terminate the program.

Step 1: Define a function that accept the age of the student through standard input.

Step 2: Use the 'try except' conditional statement to check and raise value error if necessary.

Step 3: Use the while loop to check boolean expression hold true or not till the loop with proper input is given.

`Output:`
`Enter your age : 15
Your age is not in range.`
`Enter your age : 32
Your age is not in range.`
`Enter your age : 17.`

def divide(a,b)

ans = a/b

return ans

while True:

try:

a = int(input("enter the 1st no"))

b = int(input("enter the 2nd no"))

ans = divide(a,b)

print(ans)

break

except ZeroDivisionError:

print("error!")

Output:

enter the 1st no: 1

enter the 2nd no: 1

1

enter the first no: 1

enter the second no: 0

error!

Demonstrate the use of zero Division Error.

Alg.

step 1: Use try block and accept the input using raw - - input - - () and - - except block

step 2: Define a function with two param. to divide the number entered by the user

step 3: Define while loop to the boolean expression holds true.

step 4: The except with zero division error then demonstration goes.

import re
 $s = "hello 123u abc 4567"$

$p = re.compile(r"\d+", s)$

$m = re.findall(p, s)$

$m = re.findall(r"\d+", s)$

Output:

[1234, 5567]

[hello, abc]

Algo:

Step 1: Applying a pre - integrated string and pattern in and display the output.

Step 2: As used to match all the decimal digit whereas pi is used to match decimal digit.

Practical No. 4

Aim: Demonstrate the use of regular express.

Theory: Regular expression represent the sequence of character which is mainly used for finding & supplying the given pattern in a pattern.

Import re:

```
s = "Python is an important language"
m = re.search("In Python", s)
print(m)
```

35
find matching string at beginning of a sequence.

algo:

step 1: Import the module and apply a string.

step 2: Use search() with () function and using as parameters to search for a particular string.

step 3: Now display the output.

step 4: Now use the conditional statement for user to know whether the string we know & was found or not.

Output:

```
>> re.match object = s[0:6]
      match "python" >
```

match found.

import re
 string = "python is important"
 $m1 = re.findall("lwt", string)$
 $m2 = re.findall("lwt", string)$
 print(m1)
 print(m2)

Extraction words with and without from a sentence.

0. Extraction words with and without from a sentence.
- step 1: Import re module and apply a string.
- step 2: Use find all () to extract a word from a given string.
- step 3: To extract the word along with space and space.
- step 4: Now display the output

```
import re
string = "python is important"
```

a. Extracting the first and last word of a string.

step 1: Import re module and apply a string

step 2: Use find all () in while loop and to find the last one.

step 3: now display the result.

output:

```
[python]
[important]
```

Import re

string = "ab ,findall("mlw",string)

n₁

n₂

n₃

print(n₁)

print(n₂)

print(n₃)

Output:

```
>> [abc]
```

```
[kscse.edu]
[abc, kscse, edui]
```

step 1: Display the output.

b. Extracting the

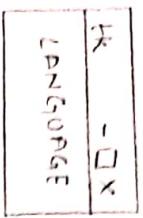
step 1: Import re module and applying for step 2: Then find all () to the username we have and lots of email - id

step 3: Display the output.

```
[abc]
[kscse.edu]
[abc, kscse, edui]
```

parent window:
 8. from tkinter import *
 9. host = Tk()
 10. l = Label(host, text = "LANGUAGE")
 11. l.pack()

Output:



Step 1: Use the `tkinter` library for importing the features of the text widget.

Step 2:

Create an object using the `Tk()`.

Step 3: Create a variable using the `Label` and use `text` method.

Step 4: Use the `mainloop()` for triggering of the corresponding above function event.

2.

```

from tkinter import *
host = Tk()
l = Label(host, text = "LANGUAGE")
l.pack()
l1 = Label(host, text = "CS!", bg = "BLUE",
           fg = "GREY", font = "10")
l1.pack(side = LEFT, padx = 20)
l2 = Label(host, text = "CS", bg = "GREEN",
           fg = "GREY", font = "20")
l2.pack(side = LEFT, pady = 30)
l3 = Label(host, text = "CS!", bg = "RED",
           fg = "GREY", font = "10")
l3.pack(side = TOP, ipadx = 40)

```

Step 1: Use `tkinter` library for importing the features of the text widget.

Step 2: Create a variable from the `text` method and position it on the parent window.

steps: Use the pack() method along with object created from the Text()

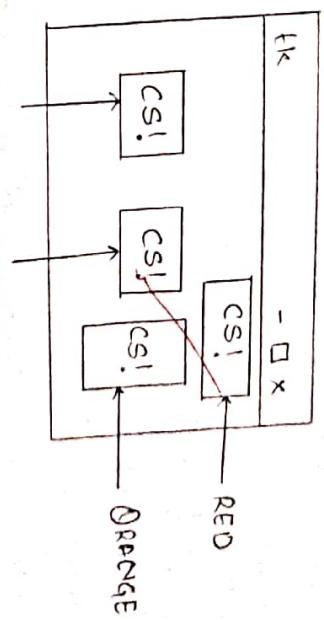
1. side = LEFT, padx = 20
2. side = LEFT, pady = 30
3. side = TOP, ipadx = 40
4. side = TOP, ipady = 50

steps: Use the mainloop() for triggering corresponding events.

steps: Repeat above steps with the Label() which takes arguments.

1. Name of parent window.
2. Text attribute defining string
3. bg colour
4. The foreground and then use pack() with a relevant padding attributes.

Output :



l1 = Label (root, text = "cs!", bg = "orange")
 l2 = Label (root, text = "cs!", bg = "blue")
 l3 = Label (root, text = "cs!", bg = "grey")
 l4 = Label (root, text = "cs!", bg = "red")
 l5 = Label (root, text = "cs!", bg = "orange")
 l6 = Label (root, text = "cs!", bg = "green")
 l7 = Label (root, text = "cs!", bg = "black")
 l8 = Label (root, text = "cs!", bg = "white")

Practical No. 6.

Radio button.
From Tkinter import *

root = Tk()
root.geometry ("500x500")

def select ():
 print ("you just selected " + str (var.get ()))
 selection = "you just selected " + str (selection, bg = "white",
 fg = "green")

l1.pack (side = top)

var = StringVar()
l1 = Listbox ()

l1.insert (1, "list 1")
l1.insert (2, "list 2")

l1.pack (anchor = N)
H1 = Radiobutton (root, text = "option 1", variable
= var, value = "option1",
command = select)

H1 = pack (anchor = N)

H2 = Radiobutton (root, text = "option 2", variable
= var, value = "option 2",
command = select)

H2 = pack (anchor = N)
root.mainloop ()

1. Import the relevant module from the
library named tkinter write an object
with the parent window.

Step 1: Import the relevant module from the
library named tkinter write an object
with the parent window.

Step 2: Use the parent window object along
with the geometry () dealing specific
pixel size of parent window.

Step 3: Now define a function which tells that
user about the given selection mode
from multiple option.

Step 4: Now define the parent window and
define the option with control variable.

Step 5: Use the listbox () and insert option
on the parent window along with
the pack () with anchor attribute.

Step 6: Create an object from radio button
which will take argument.

step 7: Now call the pack() for radio object so created and specify the argument using anchor attribute.

step 8: Use the mainloop() with parent object.

step 1: Import relevant method from the tkinter library.

step 2: Create a parent object corresponding to the parent window.

step 3: Use the geometry() for laying of the window.

step 4: Create an object and use the scrollbar()

step 5: Use the pack() along with the scrollbar object with side and fill attribute.

step 6: Use the mainloop() with the parent object

scrollbar()

from tkinter import *

root = Tk()

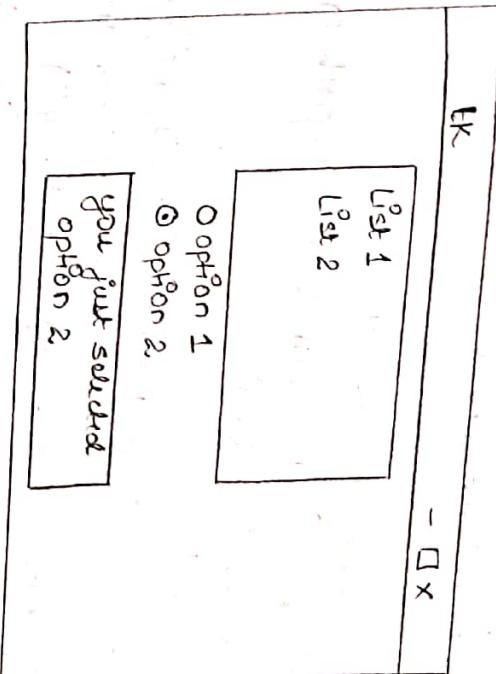
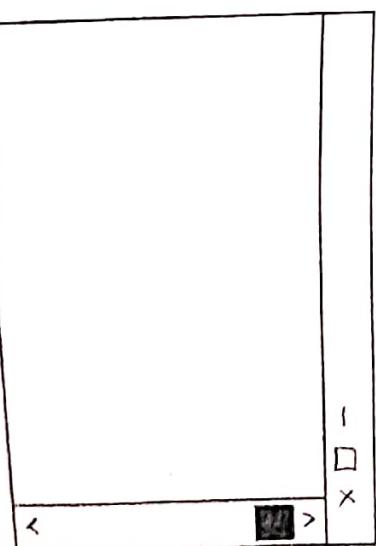
root.geometry("500x500")

s = scrollbar()

s.pack(side = "right", file = "y")

root.mainloop()

output:



3. Using frame widget.

~~Step 3: Using frame widget~~

```

from tkinter import *
window = Tk()
window.geometry("680x500")
window.title = "number"
label = Label(window)
label.pack()
frame = Frame(window)

frame.pack()

listbox = Listbox(frame, width = 20, height = 20,
                  font = ("Times New Roman", 10))

listbox.pack(side = "left", fill = "y")

scrollbar = Scrollbar(frame, orient = "vertical")

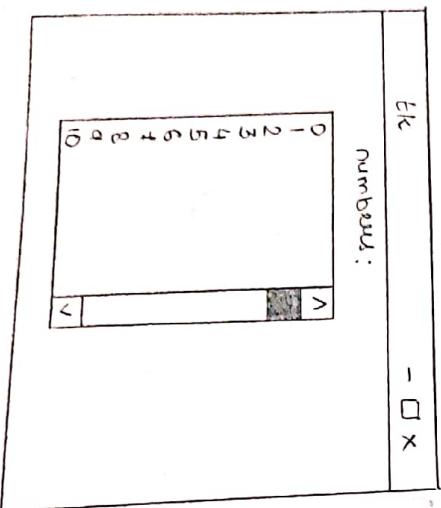
scrollbar.config(command = listbox.yview)

scrollbar.pack(side = "right", fill = "y")

for i in range(100):
    listbox.insert(END, str(i))

```

Output:



Step 1: Import the relevant libraries from the tkinter method.

Step 2: Create a corresponding object of the parent window.

Step 3: Use the geometry manager with pixel size on any other suitable pixel value.

Step 4: Use the label widget along with the parent object created and subsequently use the pack method.

Step 5: Use the frame widget along the parent object created and use the pack method.

Step 6: Use the listbox method along with the attributes like width, length, height.

Create a listbox method object use pack() for the same.

Step 7: Use the scrollbar()

Step 8: Use the mainloop()

4.

11. Python Input *

```

from tkinter import *
window = Tk()
window.geometry ("600x500")
window.mainloop()

frame = Frame (window)
frame.pack ()

frame = Frame (window)
leftframe = frame (side = "left")
leftframe.pack (side = "left")

rightframe = frame (window)
rightframe.pack (side = "right")

b1 = Button (frame, text = "select", activebackground
             = "red", bg = "blue")
b2 = Button (frame, text = "modify", activebackground
             = "yellow", bg = "black")
b3 = Button (frame, text = "ADD", activebackground
             = "blue", bg = "red")
b4 = Button (frame, text = "EXIT", activebackground
             = "red", bg = "green")

b1.pack (side = "LEFT", padx = 20)
b2.pack (side = "RIGHT", padx = 30)
b3.pack (side = "Bottom", pady = 40)
b4.pack (side = "TOP")

```

Step 1: Import relevant method from `tkinter` library.

Step 2: Define the object corresponding to parent window and define the size of parent window.

Step 3: Now define the `frame` object.

Step 4: Create another `frame` object termed as the `left frame` and put on the parent window `LEFT side`.

Step 5: Similarly define the `RIGHT frame` and subsequently define the button object placed onto the given frame.

Step 6: Now use the `pack()` along with the `side` attribute.

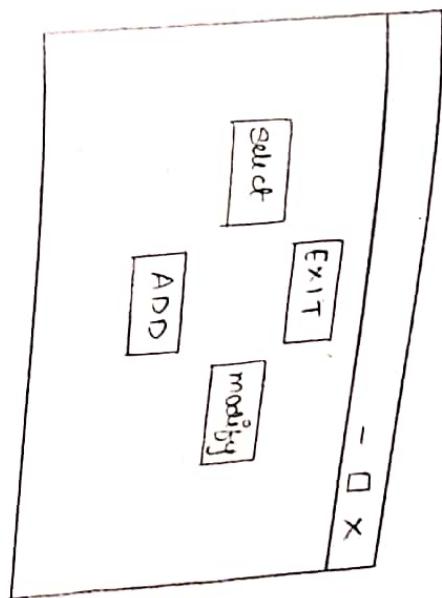
Step 7: Similarly create the button object corresponding to the `MODIFY` operation put in into `frame` object on `side: RIGHT`.

Step 8: Create another button object & place it on the `RIGHT frame` & label the button as `ADD`.

Step 9: Add another button & put it on the top of frame and label it as Exit.

Step 10: Use the pack() simultaneously for all the objects & finally use the mainloop().

Output:



Practical NO. 7.

GUI Components.

- step 1: Import the relevant method from tkinter library.
- step 2: Import tkMessageBox.
- step 3: Define a parent window object along with the parent window
- step 4: Define a function which will use tkMessageBox with showinfo method along with info window attribute
- step 5: Declare a button with parent window object along with the command attribute.
- step 6: place the button widget onto the parent window and finally call the mainloop()

step1: Import the relevant method from the tkinter library along with parent window object declared.

step2: Use the parent window object along with minsize function for window size.

step3: Define a function main , declare parent window object and use config(), title(), minsize(), label(), as well as button() and use pack() & mainloop simultaneously.

Step 4: Similarly define the function second and use the attribute accordingly.

Step5: Declare another function button along with parent object and declare button with attribute like FLAT, RIDGE, GROOVE, RAISED, SUNKEN, along with the relief widget.

Step6: Use the mainloop()

Multiple window.

Different button (Relief)

from Tkinter import *

root = Tk()

root.minsize(300, 300)

def main():

top = Tk()

top.config(bg="black")

top.title("HOME")

top.minsize(300, 300)

l = Label(top, text="SAN FRANCISCO
In Places of Interest : In golden gate
Bridge In Lombard street In Chinatown
In coit tower")

l.pack()

b1 = Button(top, text="next", command=second)

b1.pack(side=RIGHT)

b2 = Button(top, text="exit", command=terminate)

b2.pack(side=LEFT)

top.mainloop()

31 def second():

top2 = Tk()

top2.config(bg = "orange")

top2.title("About us!")

top2.minsize(300, 300)

L = Label(top2, text = "Created by: S. Raja")

"In For more details contact to our
official account")

L.pack()

b3 = Button(top2, text = "prev", command = main)

b3.pack(side = LEFT)

b2 = Button(top2, text = "exit", command = terminate)

b2.pack(side = Right)

top2.mainloop()

def button():

top3 = Tk()

top3.geometry(300 x 300)

b1 = Button(top3, text = "flat button", relief = FLAT)

b1.pack()

b2 = Button(top3, text = "groove button", relief = GROOVE)

b2.pack()

b3 = Button(top3, text = "raised button", relief = RAISED)

b3.pack()

```
b4 = Button (top3, text = "sunken button", relief = sunken)
b4.pack()
b4 = Button (top3, text = "ridge button", relief = ridge)
b4.pack()
top3.mainloop()

def terminate():
    quit()

b5 = Button (root, text = "TOUR details",
             command = main)
b5.pack()

b6 = Button (root, text = "Button Details",
             command = button)
b6.pack()

root.mainloop()
```

Practical NO.8.

GUI Component.

Step 1: Import relevant method from the tkinter library.

Step 2: Create parent window object and use the config method.

Step 3: Define a function finish with the messagebox widget which will display a message.

Step 4: Define a function info use a listbox widget along with the object of the same. use the listbox object along with insert method and then use the grid() with ipadx attribute.

Step 5: Define a function about us with label widget and text attribute and use the grid()

Step 6: Use photoimage widget with file and filename with gif attribute

```

from tkinter import *
root = Tk()
root.config(bg="grey")
root.title("warning", "This will end the program")
def finish():
    messagebox.askokcancel("warning", "This will end the program")
    quit()
def info():
    list1 = Listbox()
    list1.insert(1, "Co. Name: Apple")
    list1.insert(2, "Product: iPhone")
    list1.insert(3, "Language: swift")
    list1.insert(4, "OS: IOS")
    list1.grid(ipadx=30)
def aboutus():
    list2 = Label(text="About us")
    list2.grid(ipadx=30)
    list3 = Label(text="Steve Jobs theatre March 2020")
    list3.grid(ipadx=24)
p1 = PhotoImage(file="download.gif")
f1 = Frame(root, height=35, width=5)
f1.grid(row=1, column=0)
f2 = Frame(root, height=250, width=500)
f2.grid(row=1, column=1)
p2 = p1.subsample(5, 5)
l1 = Label(f1, image=p2, relief=FLAT)
l1.grid(row=0, column=0, padx=20, pady=15)
l2 = Label(f2, image=p1, relief=SUNKEN)
l2.grid(padx=25, pady=10)
b1 = Button(f1, text="Information", relief=SUNKEN,
            command=info)
b1.grid(row=0, column=0)

```

b2 = Button (f1, text = "About us", relief = SUNKEN,
command = aboutus)

step 7: Create a frame object along with the
frame () along with parent window.

object height and width specified.

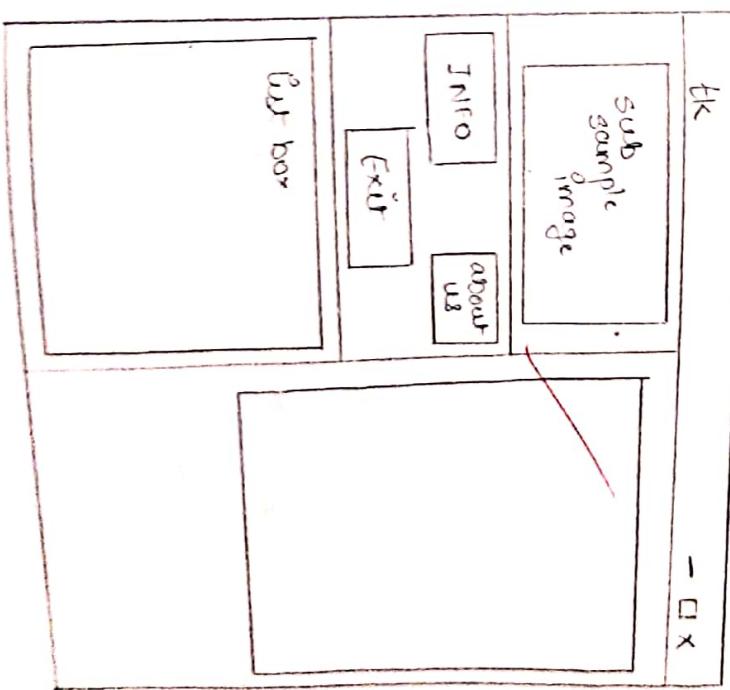
and use the grid () with row
& column attribute specified.

steps: Similarly create another frame obj
as discussed by step 7.

step 9: Create another object & else the
subsample (su)

step 10: Use label widget along with the
frame object relief attribute and
use the grid () .

step 11: Now create button object dealing with
different section of frame.



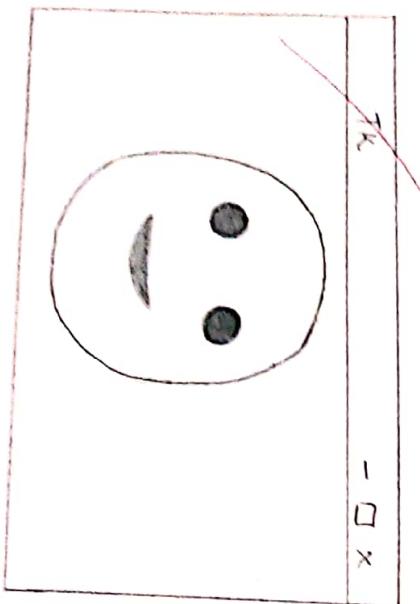
52 Tkinter Input

from tkinter import *
root = Tk()
c = canvas(root, width=500, height=500)

c.pack()

for c.create_oval(5, 50, 360, 350) outline="black"
eye1 = c.create_oval(125, 125, 275, 175, fill="black")
eye2 = c.create_oval(125, 125, 275, 175, fill="red")
nose = c.create_oval(125, 125, 275, 175, fill="red")

root.mainloop()



Practical No. 6.

Aim: Demonstrate the use of GUI by creating a human face and converting it into functioning.

Also:

Step 1: Import relevant method from tkinter library.

Step 2: Create a parent window object using Tk()

Step 3: Create a object of canvas and place it on parent window.

Step 4: Use the methods to define the face and its features.

Step 5: Finally use the mainloop()

from tkinter import *

windows = Tk()

fahrenheit = DoubleVar()

Fahrenheit = set(32)

def convert = (celcius)

fahrenheit set((9.0 / 5.0) * (celcius + 32))

l1 = Label(windows, text="Temp Value")

l1.grid(row=0, column=0)

e = Entry(windows, textvariable=fahrenheit)

e.grid(row=0, column=1)

celcius = IntVar()

l2 = Label(windows, text="Fahrenheit")

l2.grid(row=2, column=0, span=2)

windows.mainloop()

- (iii) Algo:
1. Import the relevant method from tkinter library.
2. Use the entry box widget to take the input of a DoubleVar() and store it in variable.
3. Define a function that convert celcius into fahrenheit by using $F = \frac{9}{5} \times c + 32$.
4. Use a label to display the fahrenheit value.
5. Use the button widget to call the convert function and change the fahrenheit variable value.
6. Use the mainloop() method.

Practical No. 1

```
from tkinter import *
def fact(n):
    if n == 0 or n == 1:
        return
    else:
        return n * fact(n)
```

```
# c():
result = fact(int(entry_box.get()))
text.config(result - result))
```

```
h = Tk()
entry_box = Entry(h)
B = Button(h, text, calculate, command.i())
b.pack()
h.mainloop()
```

Algo:

- Import the relevant method from the tkinter library.
- Define a function to calculate the factorial of a number.
- Use the entry widget to the input of a number.
- Use the button () widget to call the function in step 2 and calculate the factorial.
- Use the label () widget to display the factorial.
- Finally use the mainloop ()

	-	x
	<input type="button" value="5"/>	
<input type="button" value="Calculate"/>		

120

(ii) ~~algo:~~

Define a function for add , sub , multiplication . division.

iii Import the relevant method from tkinter library.

iv Use the entry () widget to take the input from double var()

v Use the button widget to call each of the function in step i.

vi Use the label widget to display method.

Finally, use the mainloop method.

```

from tkinter import *
def add (x,y)
    return (x+y)+y+0
def sub (x,y)
    return x-y
def mul (x,y)
    return x*y
def div (x,y)
    return x/y

n = Tk()
x = DoubleVar()
y = DoubleVar()
z = DoubleVar()

m1 = Radiobutton (w, text = "Add", variable = x
m2 = Radiobutton (w, text = "Sub", variable = x
m3 = Radiobutton (w, text = "Divide", variable = x
m4 = Radiobutton (w, text = "Multiple", variable = x
lambda : add(x.get(),y.get()))
lambda : sub(x.get(),y.get())
lambda : mul(x.get(),y.get())
lambda : div(x.get(),y.get())

m1.pack()
m2.pack()
m3.pack()
m4.pack()

```

32

m1 = entry (text = "enter an operand", textvariable = x)

m2 = entry (text = "enter an operand", textvariable = y)

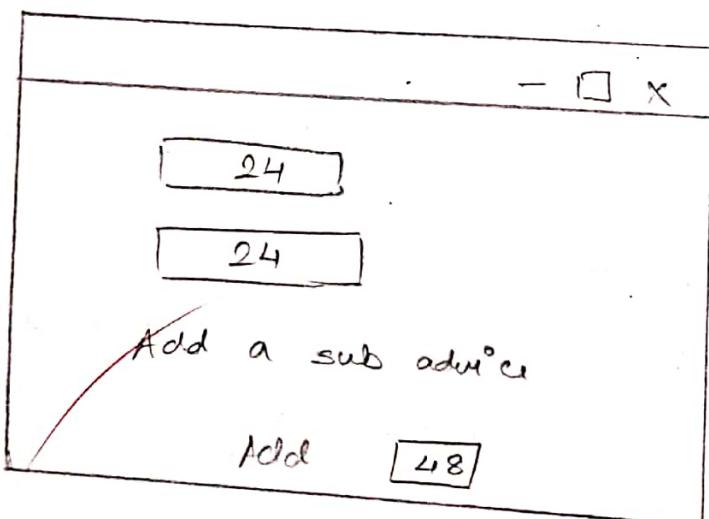
l = winfo (w, variable = z)

m1 = pack()

m2 = pack()

z = l.pack()

Output:



Practical No. 8

Server.

Algorithm:

Step 1. Import socket from the python library to implement it in the program.

2. Define a function server - prog() to state its function.

3. Show a method get last name () this socket is host variable.

4. Such port address in a variable such that port should be above 1024.

5. Get instance socket . socket method and keep it in a variable.

6. Take the bind host address and port together.

7. Configure the no. of client the server can listen simultaneously.

8. Accept no connection and print the connection add.

```
import socket  
def server - prog():  
    host = socket.gethostname()  
    port = 5000  
    server - socket - socket.socket()  
    server - socket . bind((host, port))  
    server . socket () . listen(2)  
    conn . add = server . socket . accept()  
    print ("connection from : " + str (add))  
    while True:  
        data = conn . receive (1024) . decode()  
        if not data:  
            break  
        print ("from non - user " + str (data))  
        data = input ('→')  
        server . prog()
```

```
import socket
def client_program():
    host = socket.gethostbyname()
    port = 5000
    client_socket = socket.socket()
    client_socket.connect((host, port))
    message = input("→ ")
    while message.lower() != "bye":
        client_socket.send(message.encode())
        data = client_socket.recv(1024).decode()
        message = input("→ ")
```

Client :

1. Import socket from python built-in library.
2. Define a user define function for client program.
3. Get host name from an store it in a variable.
4. Store a port number in a variable then.
5. Connect to the server using connect() method.
6. Achieve the response with recv() method with port no.
7. Show the terminal and run again the input.
8. Call the def function and run the program.

Practical 9.

Aim: Database connectivity (using sqlobject)

Algo:

1. Import the relevant libraries for the database connection and the operating system functionality.
2. Create an object for making connection to the given database.
3. Further create an object corresponding to the user area for execution of different querying subject
4. Use the ~~new class~~ object so created for implementing the structure
5. Use the ~~result~~ method for implementation
6. Now use the ~~fetchall()~~ method along with the user object for displaying the values onto the screen.

Import os, sqlobject

conn = sqlobject.connect("sqlite:///students.db")

cur = conn.cursor()

cur.execute('create table data (name char, roll no int)')

cur.execute('insert into data values ("Dhruv", 10), ("Trupti", 11), ("Smit", 9)')

conn.commit()

cur.execute('select name from data')

cur.fetchall()

cur.close()

Output:

["Dhruv", "Trupti", "Smit"]

4/3/2022