Socket Programming :-

Using TCP/IP sockets, write a client server program to make client sending file name and server to send back the contents.

- * A server has bind() method which binds specific IP and port so that it can listen to incoming requests
- * Passing an empty string means that the server can listen to incoming connections from other computers as well.
- * Server is now in listening mode.
- * At last, we make a while loop and start to accept all incoming connections and close those connections.

Client TCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("Enter file name")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('\n From server\n')
print(filecontents)
clientSocket.close()
```

Output is been later, Server TCP.py

→ The server codes is :-

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket. listen(1)

while 1 :
   print ("Server is ready to receive")
   connection Socket, addr = serversocket. accept()
   file = open(sentence, "r")
   l = file read (1024)
   connectionsocket. send (l. encode())
   print ("sent contents' +sentence)
   file. close ()
```

Output :-

```
The server is ready to receive          } server side
Contents of server TCP. py
The server is ready to receive

Enter file name :- ServerTCP. py

From Server :

from socket import *
serverName = ' 127. 0. 0. 1'
server Port = 12000

connectionSocket. send (l. encode())
 file. close ()
 connection socket. close ()
```