```
# Python program to implement server side of chat room.
import socket
import select
import sys
from thread import *


server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

# checks whether sufficient arguments have been provided
if len(sys.argv) != 3:
    print "Correct usage: script, IP address, port number"
    exit()

# takes the first argument from command prompt as IP address
IP_address = str(sys.argv[1])

# takes second argument from command prompt as port number
Port = int(sys.argv[2])

"""
binds the server to an entered IP address and at the
specified port number.
The client must be aware of these parameters
"""
server.bind((IP_address, Port))

"""
listens for 100 active connections. This number can be
increased as per convenience.
"""
server.listen(100)

list_of_clients = []

def clientthread(conn, addr):

    # sends a message to the client whose user object is conn
    conn.send("Welcome to this chatroom!")

    while True:
            try:
                message = conn.recv(2048)
                if message:

                    """prints the message and address of the
                    user who just sent the message on the server
                    terminal"""
                    print "<" + addr[0] + "> " + message

                    # Calls broadcast function to send message to all
                    message_to_send = "<" + addr[0] + "> " + message
                    broadcast(message_to_send, conn)

                else:
                    """message may have no content if the connection
                    is broken, in this case we remove the connection"""
                    remove(conn)

            except:
                continue

"""Using the below function, we broadcast the message to all
clients who's object is not the same as the one sending
the message """
```

```
def broadcast(message, connection):
    for clients in list_of_clients:
        if clients!=connection:
            try:
                clients.send(message)
            except:
                clients.close()

                # if the link is broken, we remove the client
                remove(clients)

"""The following function simply removes the object
from the list that was created at the beginning of
the program"""
def remove(connection):
    if connection in list_of_clients:
        list_of_clients.remove(connection)

while True:

    """Accepts a connection request and stores two parameters,
    conn which is a socket object for that user, and addr
    which contains the IP address of the client that just
    connected"""
    conn, addr = server.accept()

    """Maintains a list of clients for ease of broadcasting
    a message to all available people in the chatroom"""
    list_of_clients.append(conn)

    # prints the address of the user that just connected
    print addr[0] + " connected"


    # creates and individual thread for every user
    # that connects
    start_new_thread(clientthread,(conn,addr))

conn.close()
server.close()
```