**Question 2**

**B.1.1 Design algorithm(s)/flowchart(s) for the problem**

**Client Side done using UDP**

**Step 1**:Create the socket object for carrying the data

DatagramSocket ds = new DatagramSocket();

InetAddress ip = InetAddress.getLocalHost();

Initialize byte buf[] = null;

**Step 2** :loop while user not enters "bye"

while (true) {

Display "Enter the equation in the format:"

Display "'operand1 operator operand2'"

String inp = sc.nextLine();

buf = new byte[65535];

**Step 3**:convert the String input into the byte array.

buf = inp.getBytes();

**Step 4**:Create the datagramPacket for sending the data.

DatagramPacket DpSend= new DatagramPacket(buf, buf.length, ip, 1234);

**Step 5:**invoke the send call to actually send the data.

ds.send(DpSend);

**Step 6**:break the loop if user enters "bye"

if (inp.equals("bye")) {

break;

} buf = new byte[65535];

DatagramPacket DpReceive= new DatagramPacket(buf, buf.length);

ds.receive(DpReceive);

Display"Answer = "+ new String(buf, 0, buf.length) }}}

**Step 7:**Stop

**Server Side done using UDP**

**Step 1:**Create a socket to listen at port 1234

DatagramSocket ds = new DatagramSocket(1234);

Initialize byte[] buf = null;

Initialize DatagramPacket DpReceive = null;

Initialize DatagramPacket DpSend = null;

while (true)

{buf = new byte[65535];

**Step 2:**create a DatgramPacket to receive the data.

DpReceive = new DatagramPacket(buf, buf.length);

**Step 3:**receive the data in byte buffer.

ds.receive(DpReceive);

String inp = new String(buf, 0, buf.length);

**Step 4:**To remove extra spaces.

inp=inp.trim();

Display "Equation Received:- " + inp;

**Step 5:** Exit the server if the client sends "bye"

if (inp.equals("bye"))

{Display "Client sent bye.....EXITING"

break;}

**Step 6:**Use StringTokenizer to break the equation into operand and operation

StringTokenizer st = new StringTokenizer(inp);

int oprnd1 = Integer.parseInt(st.nextToken());

String operation = st.nextToken();

int oprnd2 = Integer.parseInt(st.nextToken());

**Step 7:** perform the required operation.

if (operation.equals("+"))

result = oprnd1 + oprnd2;

else if (operation.equals("-"))

result = oprnd1 - oprnd2;

else if (operation.equals("\*"))

result = oprnd1 \* oprnd2;

else

result = oprnd1 / oprnd2;

Display ("Sending the result...");

String res = Integer.toString(result);

**Step 8:**clear the buffer after every message.

buf = res.getBytes();

**Step 9:**get the port of client.

int port = DpReceive.getPort();

DpSend = new DatagramPacket(buf, buf.length,

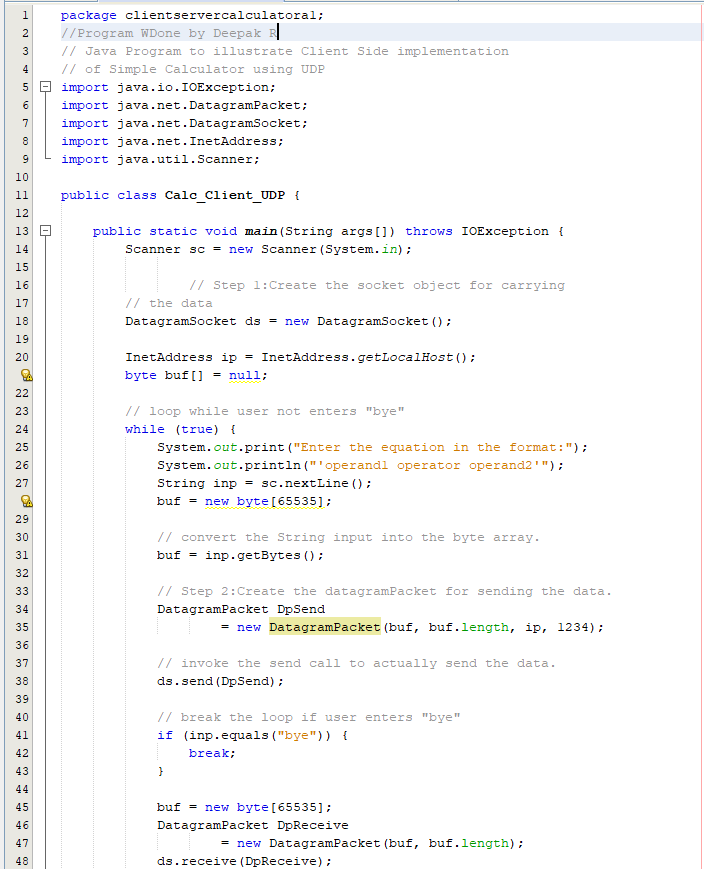
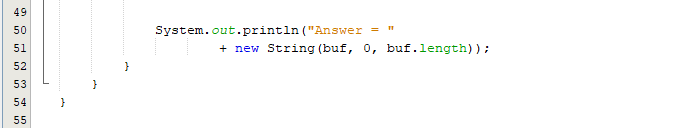
InetAddress.getLocalHost(), port);

ds.send(DpSend);}}}

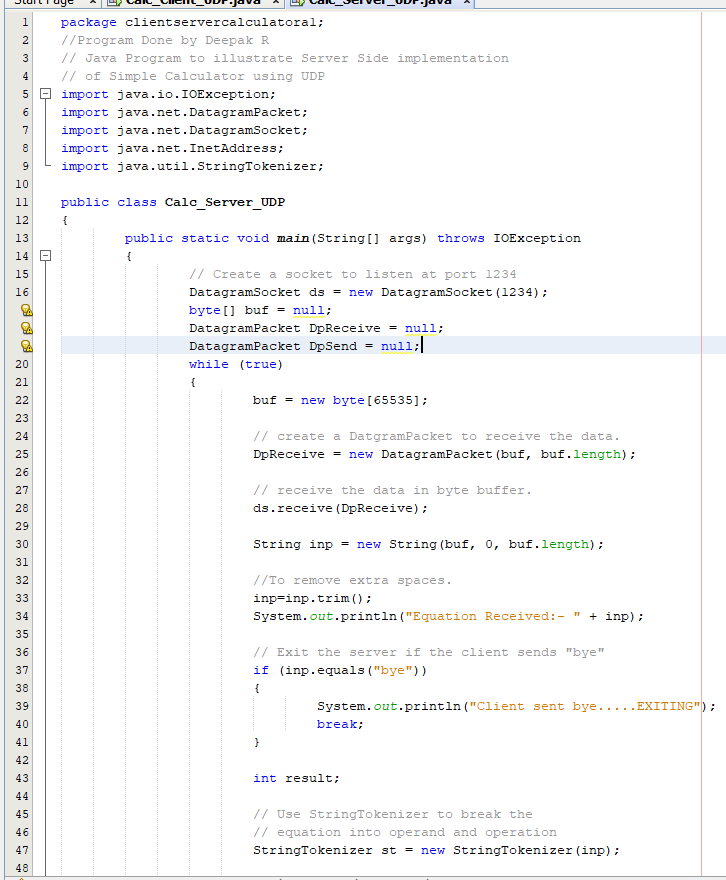
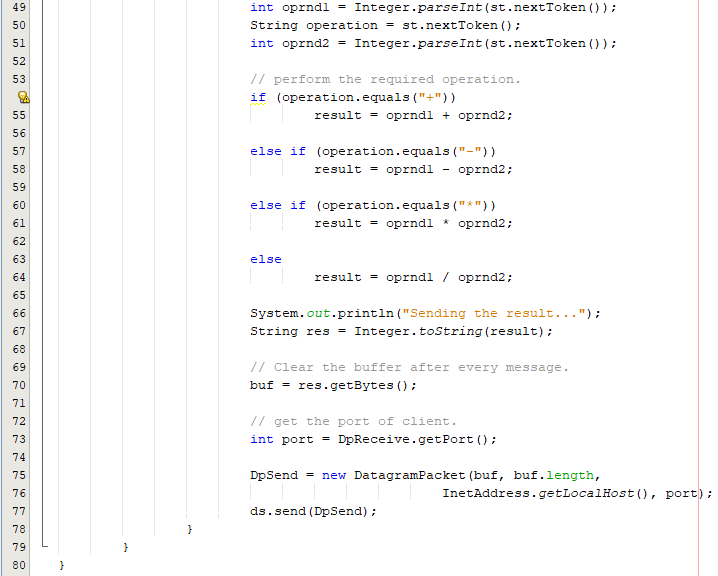
**Step 10:**Stop

**B.1.2 Implementation**

**Client side**

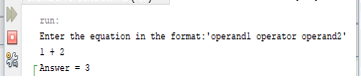
**Server Side**

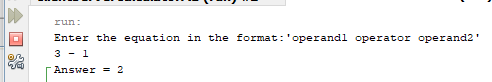
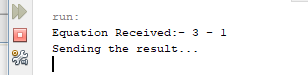
 

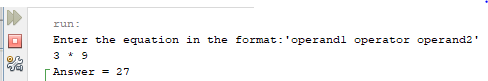
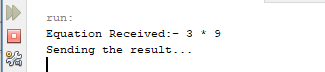
**B.1.3 Results and Analysis**

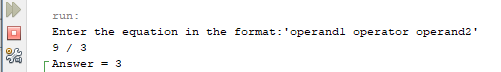
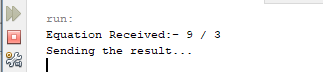
**Test Cases**

**Client Side Server Side**

The calculator server accepts requests to add, subtract, multiple, and divide a pair of integers. :We run the server program first and then the client one to run Program Successfully. When we enter our operands in client side Server calculate the given operation and send back result to client and Result gets Displayed on Client Side.