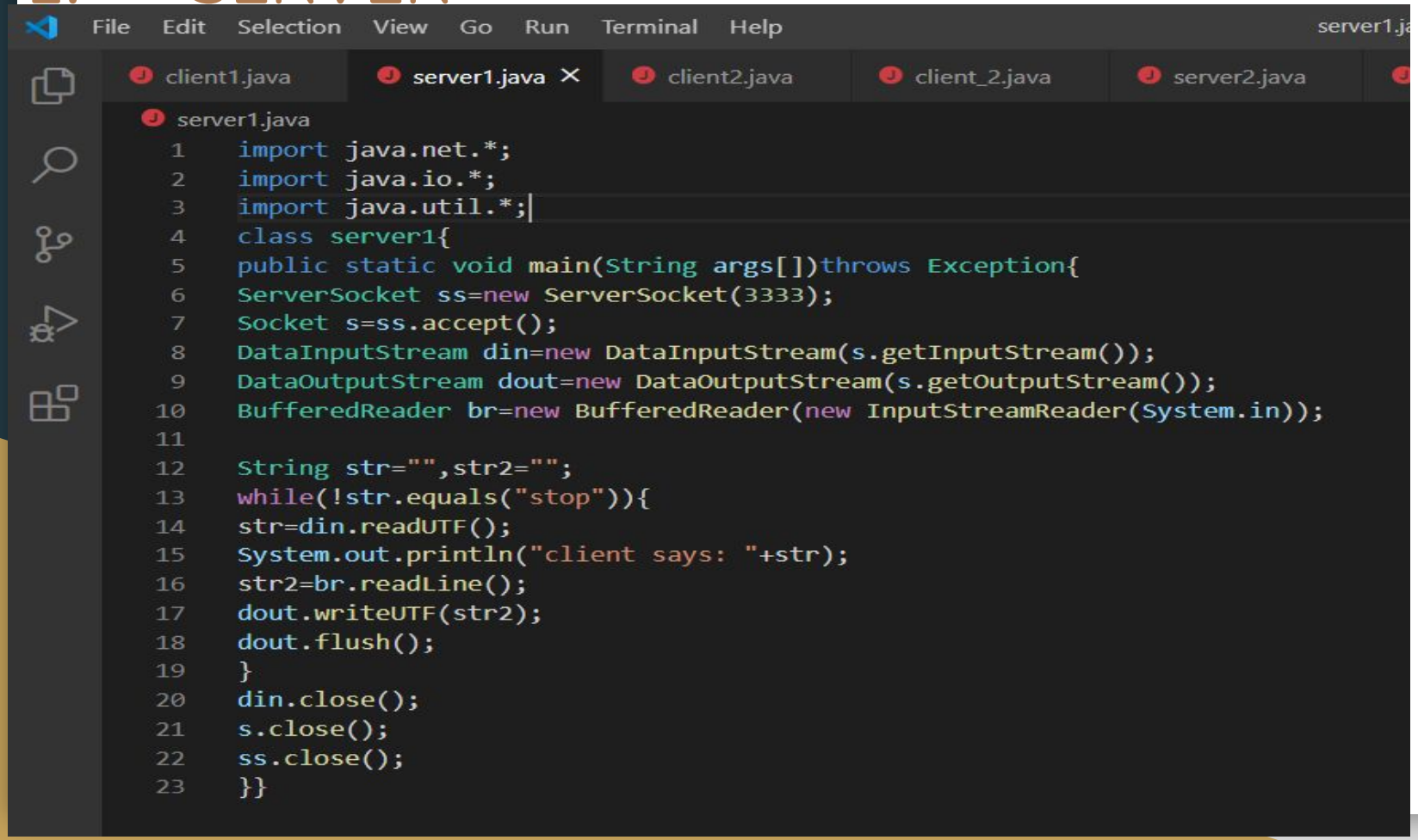
16-09-2020
LAB EXERCISES

DONE BY,
K.SASI KIRAN
MCA(R)
2019202049

1: SERVER


File Edit Selection View Go Run Terminal Help
client1.java
server1.java
client2.java
client_2.java
server2.java
server1.java
1 import java.net.*;
2 import java.io.*;
3 import java.util.*;
4 class server1{
5 public static void main(String args[])throws Exception{
6 ServerSocket ss=new ServerSocket(3333);
7 Socket s=ss.accept();
8 DataInputStream din=new DataInputStream(s.getInputStream());
9 DataOutputStream dout=new DataOutputStream(s.getOutputStream());
10 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
11
12 String str="",str2="";
13 while(!str.equals("stop")){
14 str=din.readUTF();
15 System.out.println("client says: "+str);
16 str2=br.readLine();
17 dout.writeUTF(str2);
18 dout.flush();
19 }
20 din.close();
21 s.close();
22 ss.close();
23 }}

CLIENT

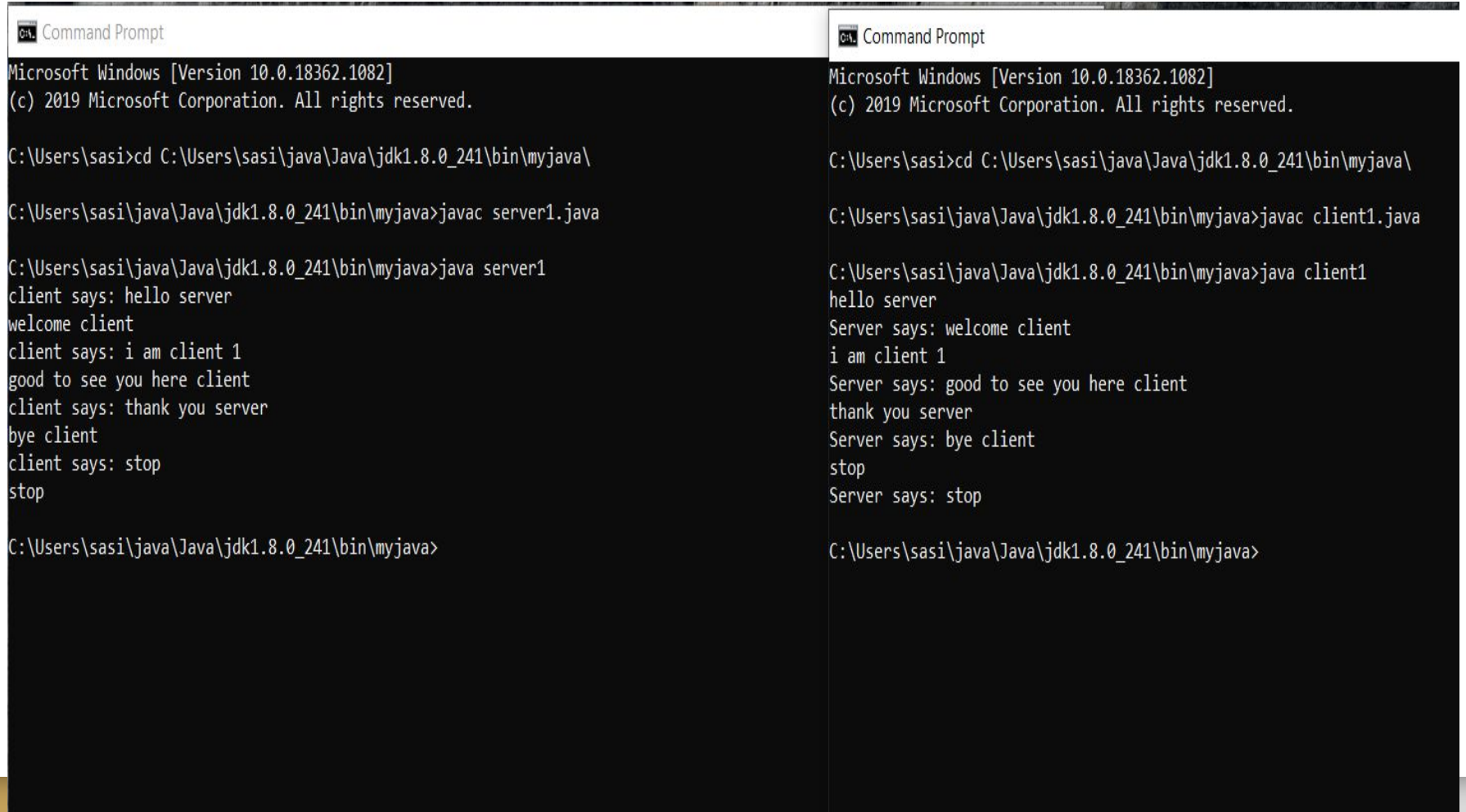
File Edit Selection View Go Run Terminal Help clien

client1.java × server1.java client2.java client_2.java server2.java

client1.java

```
1   import java.net.*;
2   import java.io.*;
3   import java.util.*;
4   class client1{
5   public static void main(String args[])throws Exception{
6   Socket s=new Socket("localhost",3333);
7   DataInputStream din=new DataInputStream(s.getInputStream());
8   DataOutputStream dout=new DataOutputStream(s.getOutputStream());
9   BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
10
11  String str="",str2="";
12  while(!str.equals("stop")){
13  str=br.readLine();
14  dout.writeUTF(str);
15  dout.flush();
16  str2=din.readUTF();
17  System.out.println("Server says: "+str2);
18  }
19
20  dout.close();
21  s.close();
22  }}
```

OUTPUT



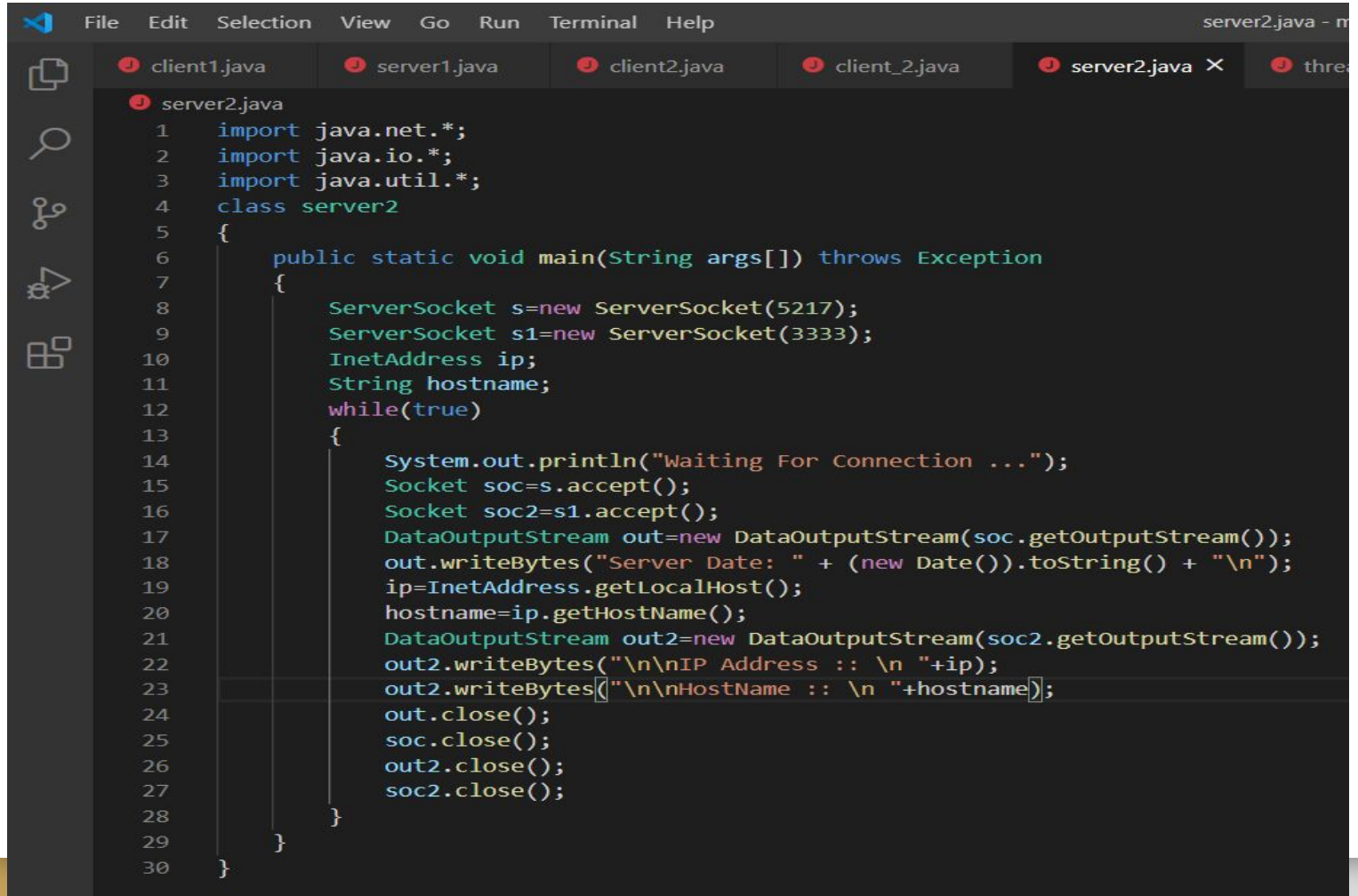
Command Prompt

```
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\sasi>cd C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava\

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>javac server1.java

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java server1
client says: hello server
welcome client
client says: i am client 1
good to see you here client
client says: thank you server
bye client
client says: stop
stop

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>
```

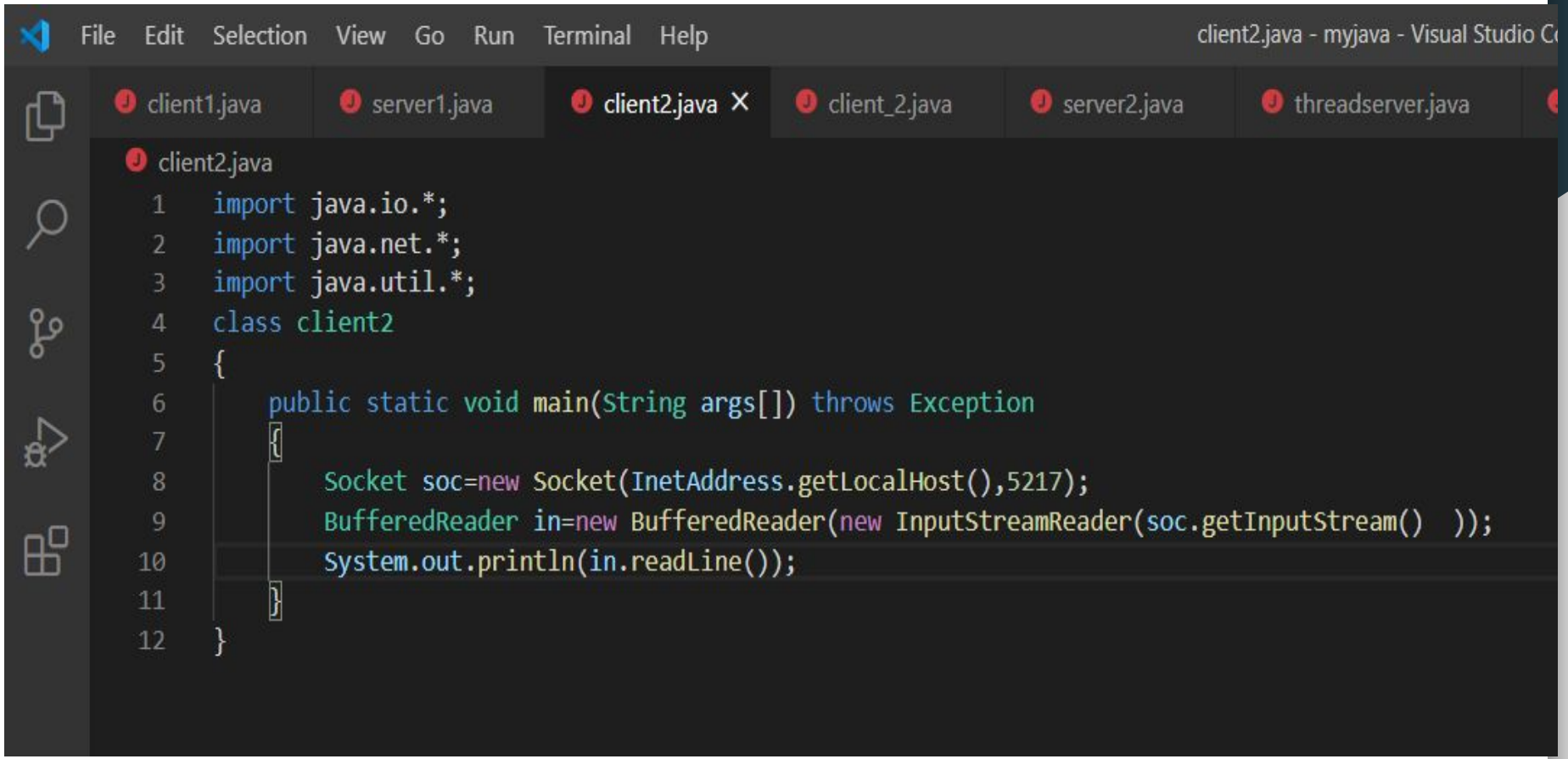
Command Prompt

```
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\sasi>cd C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava\

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>javac client1.java

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java client1
hello server
Server says: welcome client
i am client 1
Server says: good to see you here client
thank you server
Server says: bye client
stop
Server says: stop

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>
```

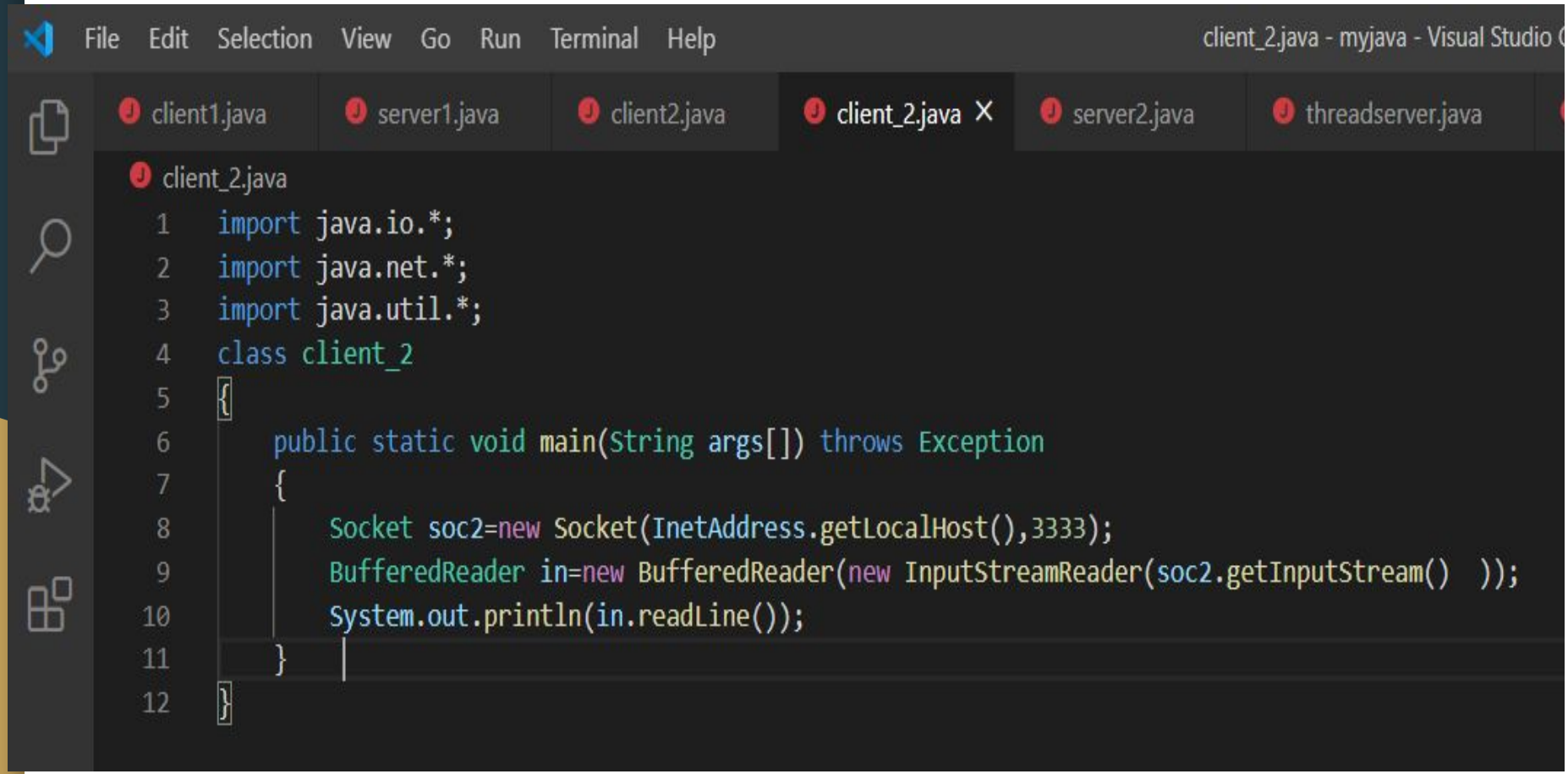
2: SERVER

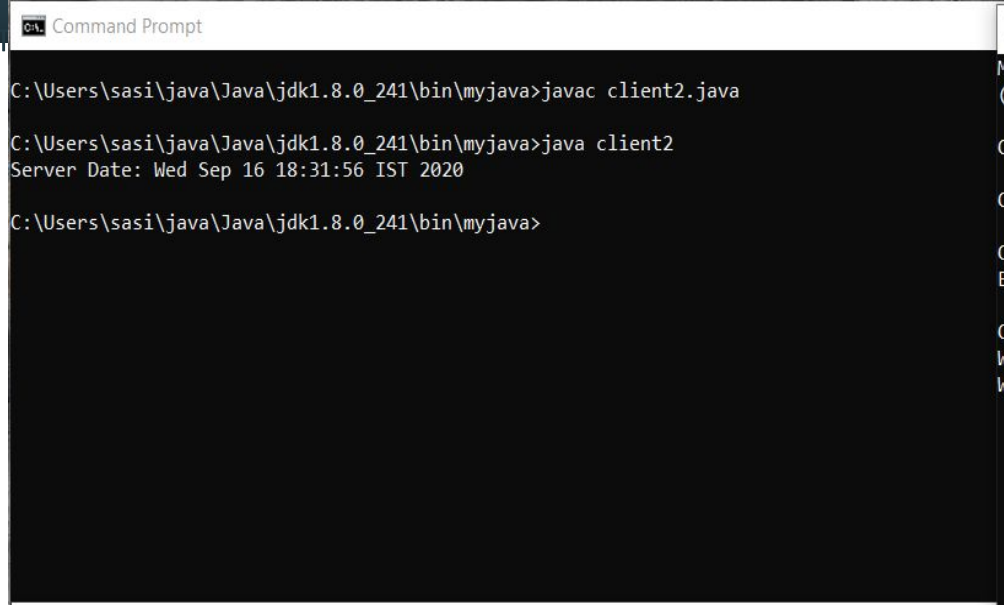

File Edit Selection View Go Run Terminal Help
server2.java - m
client1.java
server1.java
client2.java
client_2.java
server2.java
server2.java
1 import java.net.*;
2 import java.io.*;
3 import java.util.*;
4 class server2
5 {
6 public static void main(String args[]) throws Exception
7 {
8 ServerSocket s=new ServerSocket(5217);
9 ServerSocket s1=new ServerSocket(3333);
10 InetAddress ip;
11 String hostname;
12 while(true)
13 {
14 System.out.println("Waiting For Connection ...");
15 Socket soc=s.accept();
16 Socket soc2=s1.accept();
17 DataOutputStream out=new DataOutputStream(soc.getOutputStream());
18 out.writeBytes("Server Date: " + (new Date()).toString() + "\n");
19 ip=InetAddress.getLocalHost();
20 hostname=ip.getHostName();
21 DataOutputStream out2=new DataOutputStream(soc2.getOutputStream());
22 out2.writeBytes("\n\nIP Address :: \n "+ip);
23 out2.writeBytes("\n\nHostName :: \n "+hostname);
24 out.close();
25 soc.close();
26 out2.close();
27 soc2.close();
28 }
29 }
30 }

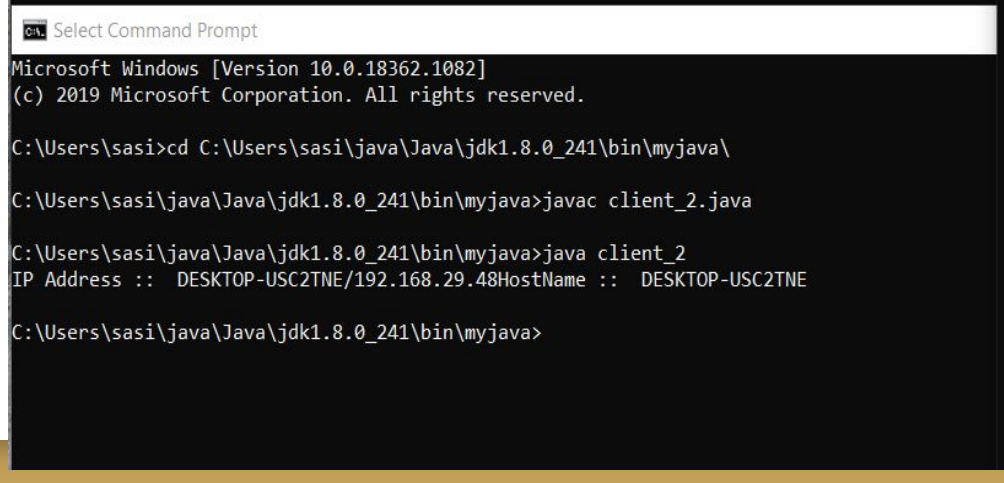
CLIENT 1

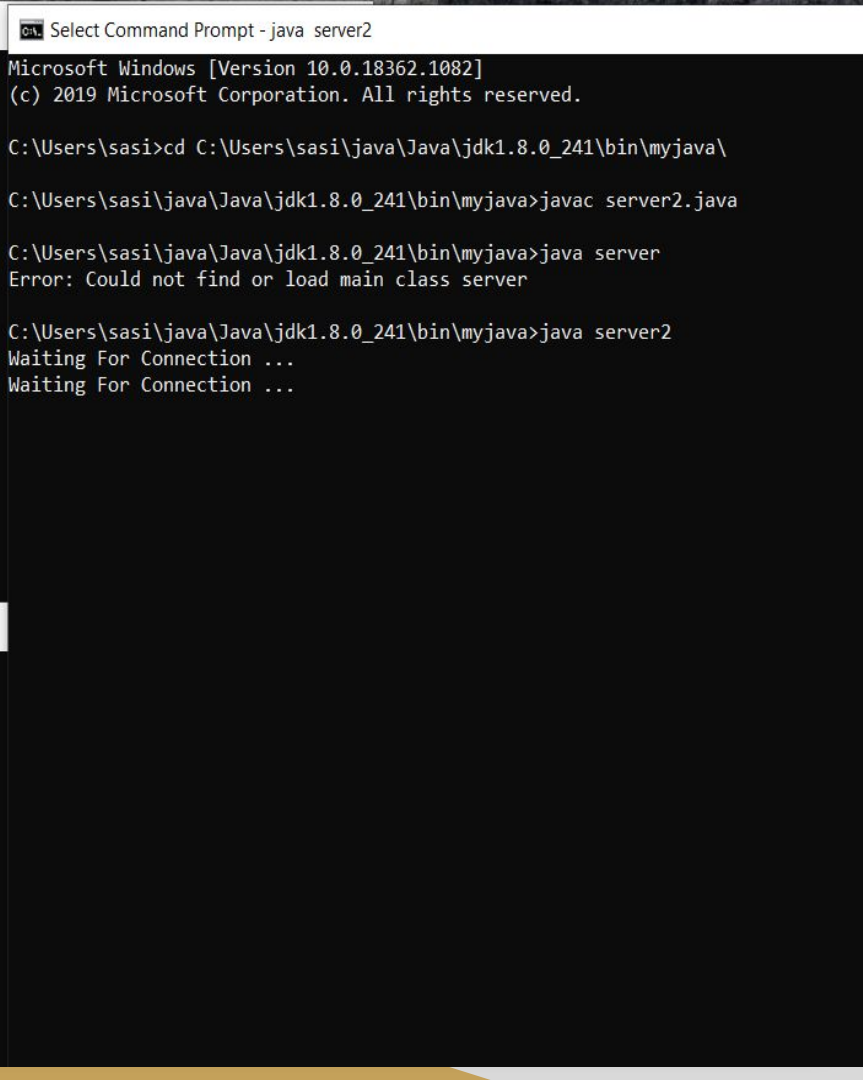

File Edit Selection View Go Run Terminal Help
client2.java - myjava - Visual Studio C
client1.java
server1.java
client2.java
client_2.java
server2.java
threadserver.java
client2.java
1 import java.io.*;
2 import java.net.*;
3 import java.util.*;
4 class client2
5 {
6 public static void main(String args[]) throws Exception
7 {
8 Socket soc=new Socket(InetAddress.getLocalHost(),5217);
9 BufferedReader in=new BufferedReader(new InputStreamReader(soc.getInputStream()));
10 System.out.println(in.readLine());
11 }
12 }

CLIENT 2

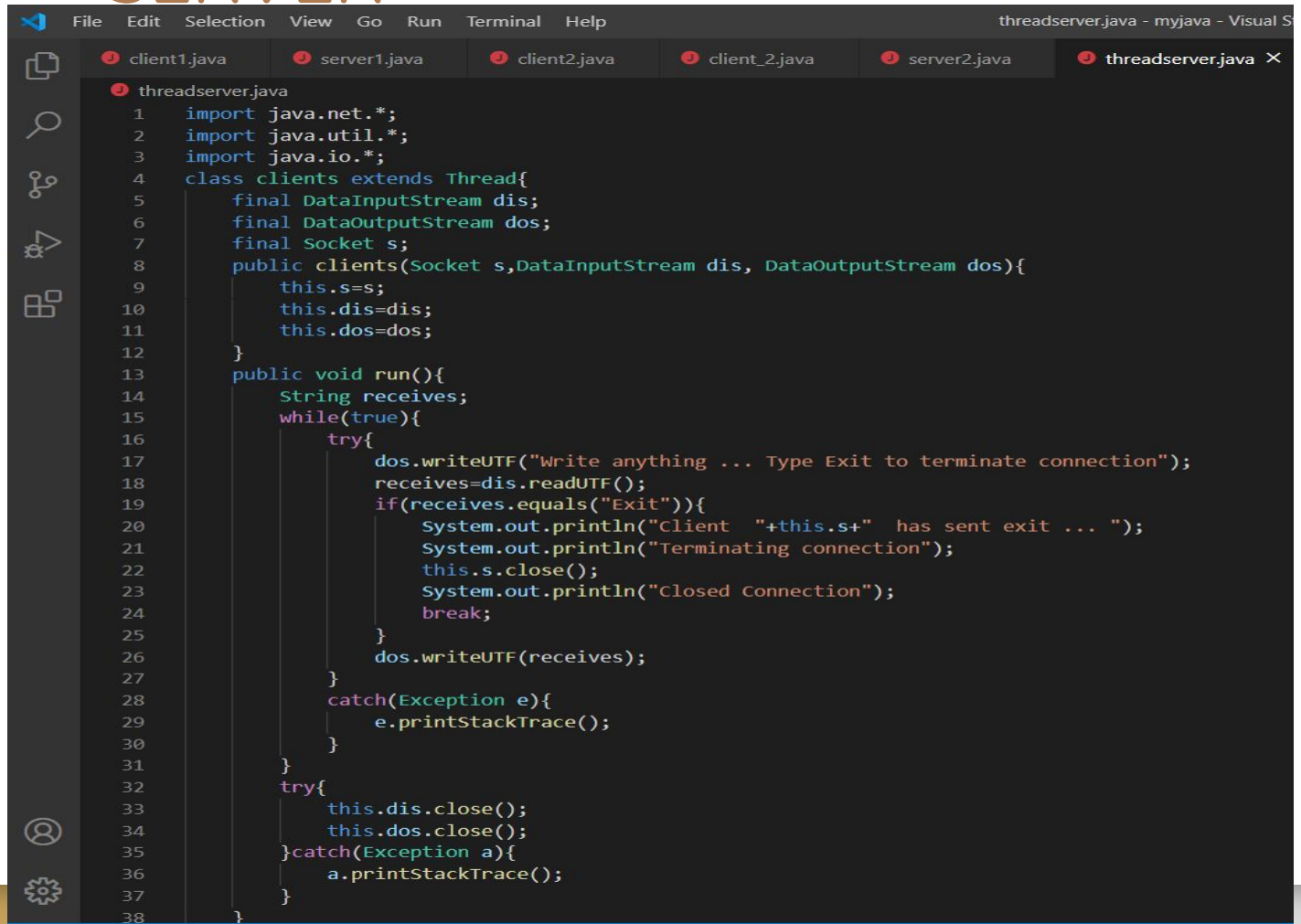

File Edit Selection View Go Run Terminal Help
client_2.java - myjava - Visual Studio
client1.java
server1.java
client2.java
client_2.java
server2.java
threadserver.java
client_2.java
1 import java.io.*;
2 import java.net.*;
3 import java.util.*;
4 class client_2
5 {
6 public static void main(String args[]) throws Exception
7 {
8 Socket soc2=new Socket(InetAddress.getLocalHost(),3333);
9 BufferedReader in=new BufferedReader(new InputStreamReader(soc2.getInputStream()));
10 System.out.println(in.readLine());
11 }
12 }

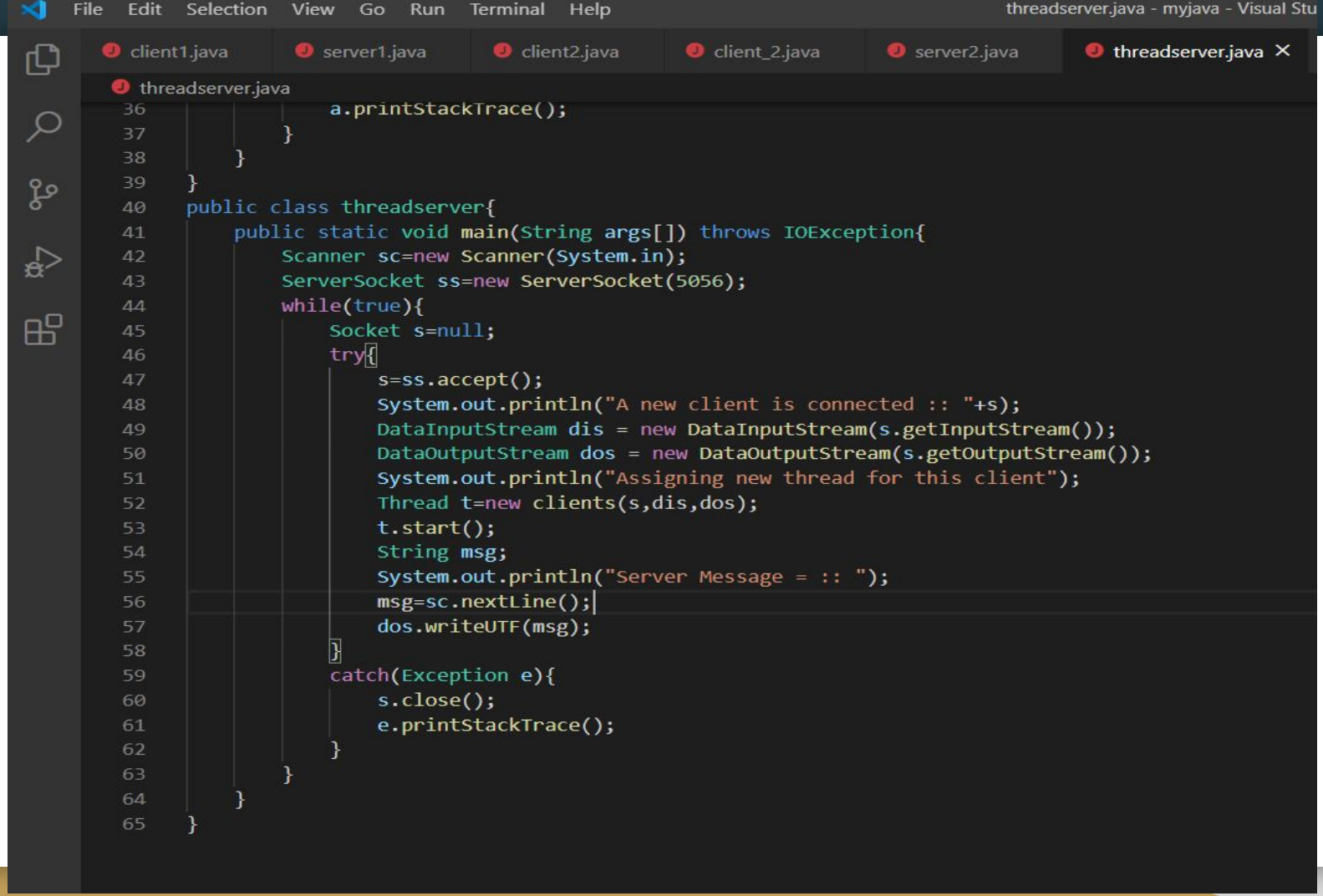

Command Prompt
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>javac client2.java
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java client2
Server Date: Wed Sep 16 18:31:56 IST 2020
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>


Select Command Prompt
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\sasi>cd C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava\
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>javac client_2.java
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java client_2
IP Address :: DESKTOP-USC2TNE/192.168.29.48HostName :: DESKTOP-USC2TNE
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>

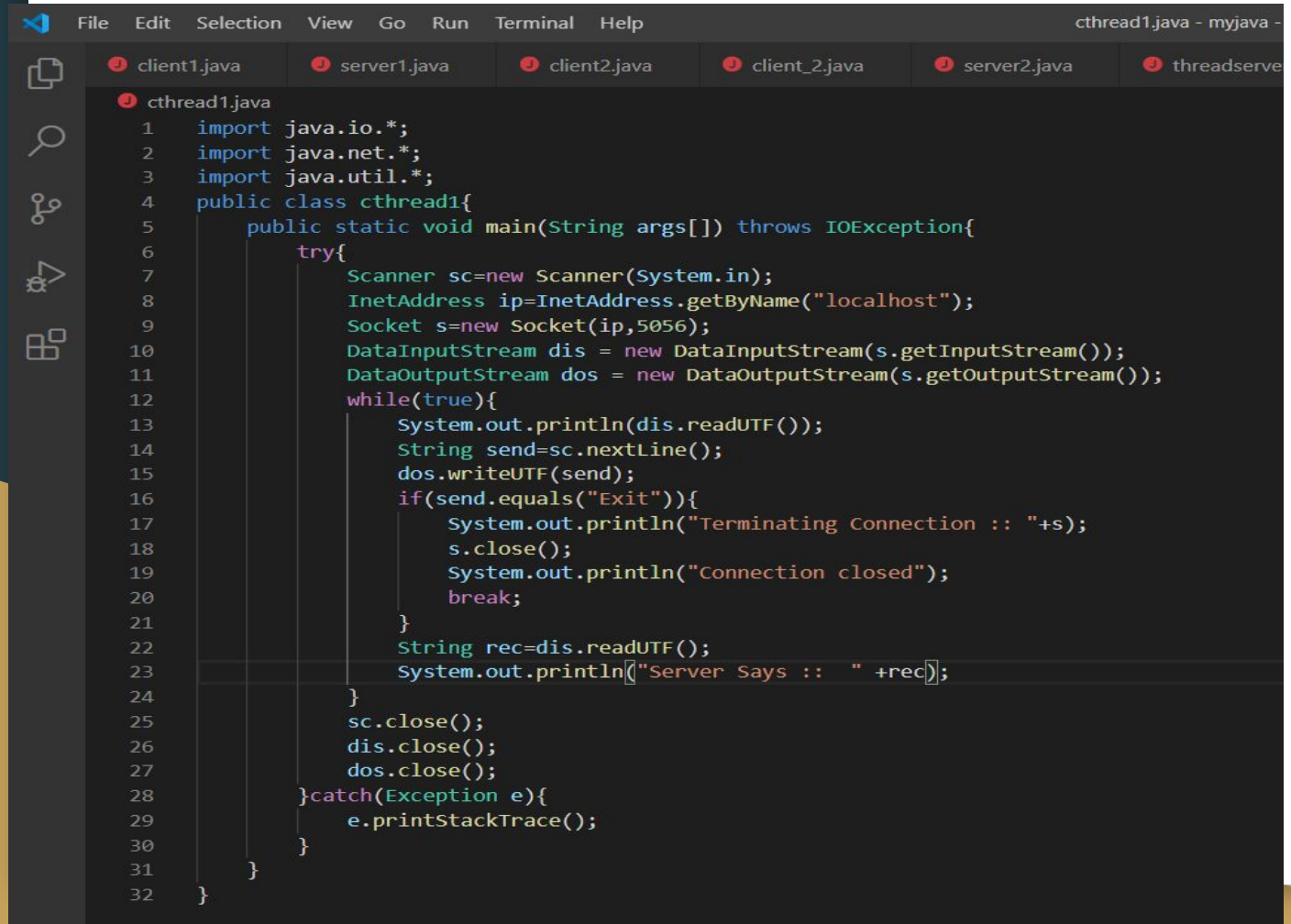

Select Command Prompt - java server2
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\sasi>cd C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava\
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>javac server2.java
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java server
Error: Could not find or load main class server
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java server2
Waiting For Connection ...
Waiting For Connection ...

3: SERVER

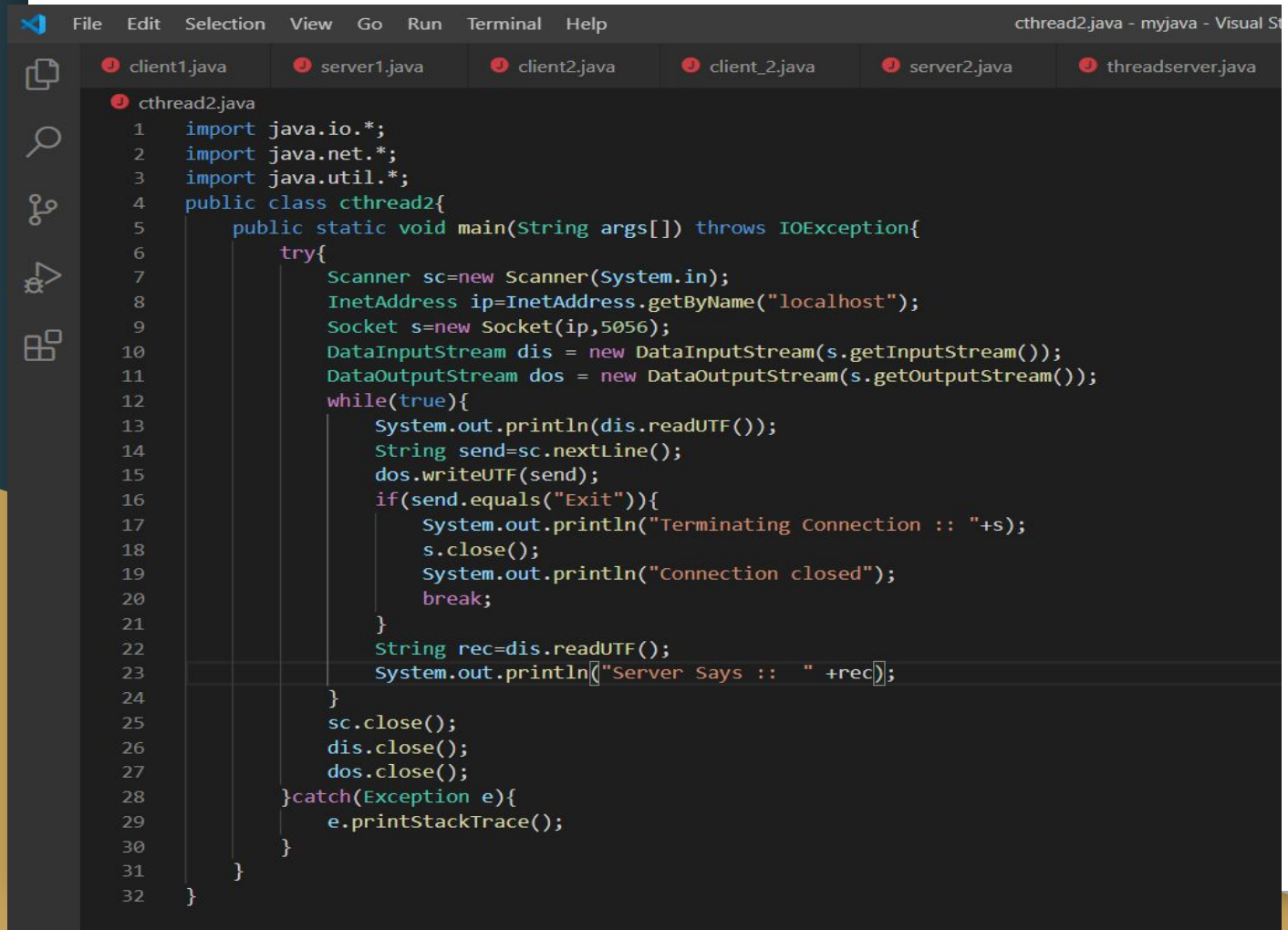

File Edit Selection View Go Run Terminal Help
threadserver.java - myjava - Visual S
client1.java
server1.java
client2.java
client_2.java
server2.java
threadserver.java
threadserver.java
1 import java.net.*;
2 import java.util.*;
3 import java.io.*;
4 class clients extends Thread{
5 final DataInputStream dis;
6 final DataOutputStream dos;
7 final Socket s;
8 public clients(Socket s,DataInputStream dis, DataOutputStream dos){
9 this.s=s;
10 this.dis=dis;
11 this.dos=dos;
12 }
13 public void run(){
14 String receives;
15 while(true){
16 try{
17 dos.writeUTF("Write anything ... Type Exit to terminate connection");
18 receives=dis.readUTF();
19 if(receives.equals("Exit")){
20 System.out.println("Client "+this.s+" has sent exit ... ");
21 System.out.println("Terminating connection");
22 this.s.close();
23 System.out.println("Closed Connection");
24 break;
25 }
26 dos.writeUTF(receives);
27 }
28 catch(Exception e){
29 e.printStackTrace();
30 }
31 }
32 try{
33 this.dis.close();
34 this.dos.close();
35 }catch(Exception a){
36 a.printStackTrace();
37 }
38 }


File Edit Selection View Go Run Terminal Help
threadserver.java - myjava - Visual Stu
client1.java
server1.java
client2.java
client_2.java
server2.java
threadserver.java
threadserver.java
a.printStackTrace();
}
}
}
public class threadserver{
public static void main(String args[]) throws IOException{
Scanner sc=new Scanner(System.in);
ServerSocket ss=new ServerSocket(5056);
while(true){
Socket s=null;
try{
s=ss.accept();
System.out.println("A new client is connected :: "+s);
DataInputStream dis = new DataInputStream(s.getInputStream());
DataOutputStream dos = new DataOutputStream(s.getOutputStream());
System.out.println("Assigning new thread for this client");
Thread t=new clients(s,dis,dos);
t.start();
String msg;
System.out.println("Server Message = :: ");
msg=sc.nextLine();
dos.writeUTF(msg);
}
catch(Exception e){
s.close();
e.printStackTrace();
}
}
}
}

CLIENT 1

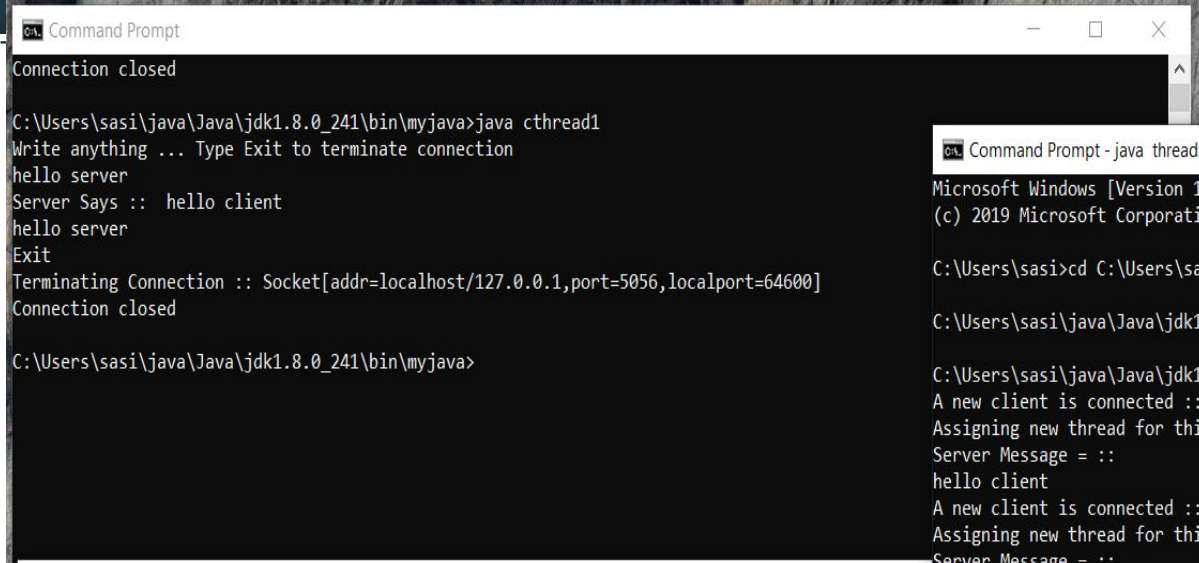

File Edit Selection View Go Run Terminal Help
cthread1.java - myjava -
client1.java
server1.java
client2.java
client_2.java
server2.java
threadserve
cthread1.java
1 import java.io.*;
2 import java.net.*;
3 import java.util.*;
4 public class cthread1{
5 public static void main(String args[]) throws IOException{
6 try{
7 Scanner sc=new Scanner(System.in);
8 InetAddress ip=InetAddress.getByName("localhost");
9 Socket s=new Socket(ip,5056);
10 DataInputStream dis = new DataInputStream(s.getInputStream());
11 DataOutputStream dos = new DataOutputStream(s.getOutputStream());
12 while(true){
13 System.out.println(dis.readUTF());
14 String send=sc.nextLine();
15 dos.writeUTF(send);
16 if(send.equals("Exit")){
17 System.out.println("Terminating Connection :: "+s);
18 s.close();
19 System.out.println("Connection closed");
20 break;
21 }
22 String rec=dis.readUTF();
23 System.out.println("Server Says :: " +rec);
24 }
25 sc.close();
26 dis.close();
27 dos.close();
28 }catch(Exception e){
29 e.printStackTrace();
30 }
31 }
32 }

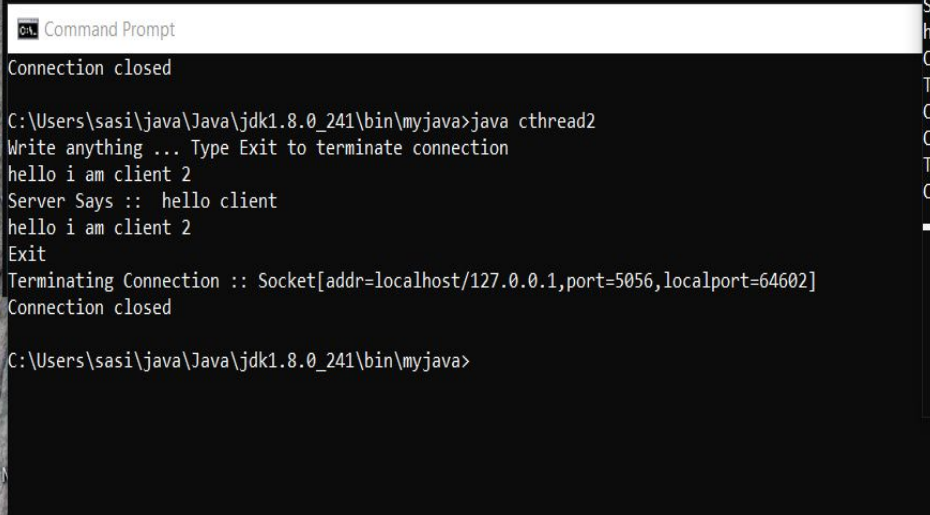
CLIENT 2

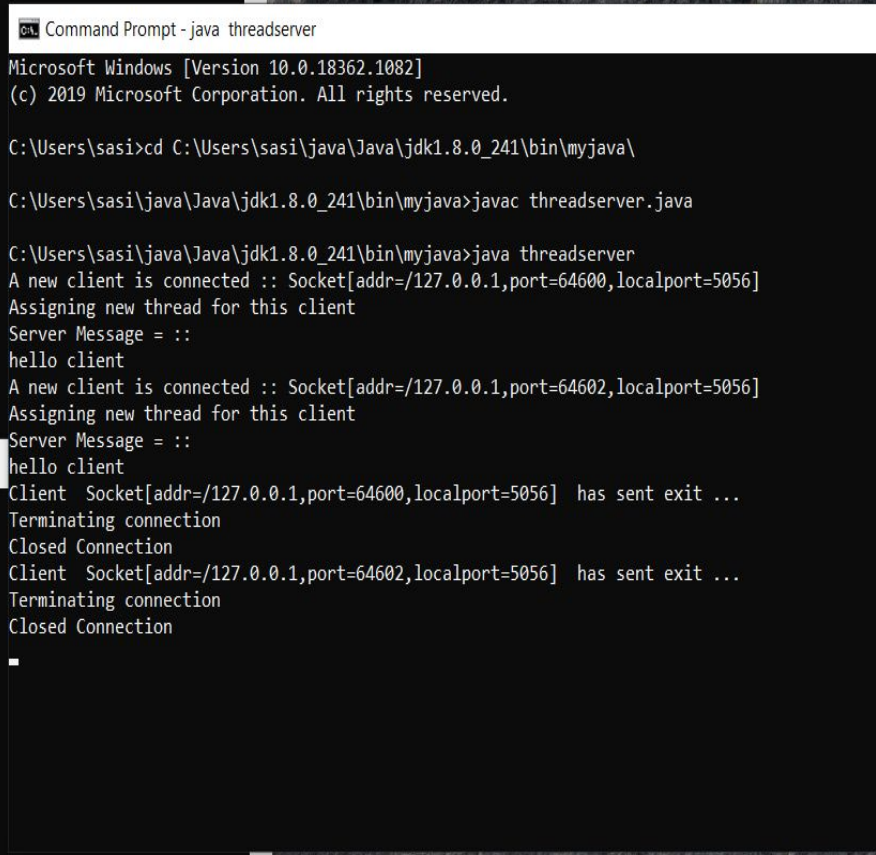



```java
import java.io.*;
import java.net.*;
import java.util.*;
public class cthread2{
    public static void main(String args[]) throws IOException{
        try{
            Scanner sc=new Scanner(System.in);
            InetAddress ip=InetAddress.getByName("localhost");
            Socket s=new Socket(ip,5056);
            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
            while(true){
                System.out.println(dis.readUTF());
                String send=sc.nextLine();
                dos.writeUTF(send);
                if(send.equals("Exit")){
                    System.out.println("Terminating Connection :: "+s);
                    s.close();
                    System.out.println("Connection closed");
                    break;
                }
                String rec=dis.readUTF();
                System.out.println("Server Says ::  " +rec);
            }
            sc.close();
            dis.close();
            dos.close();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

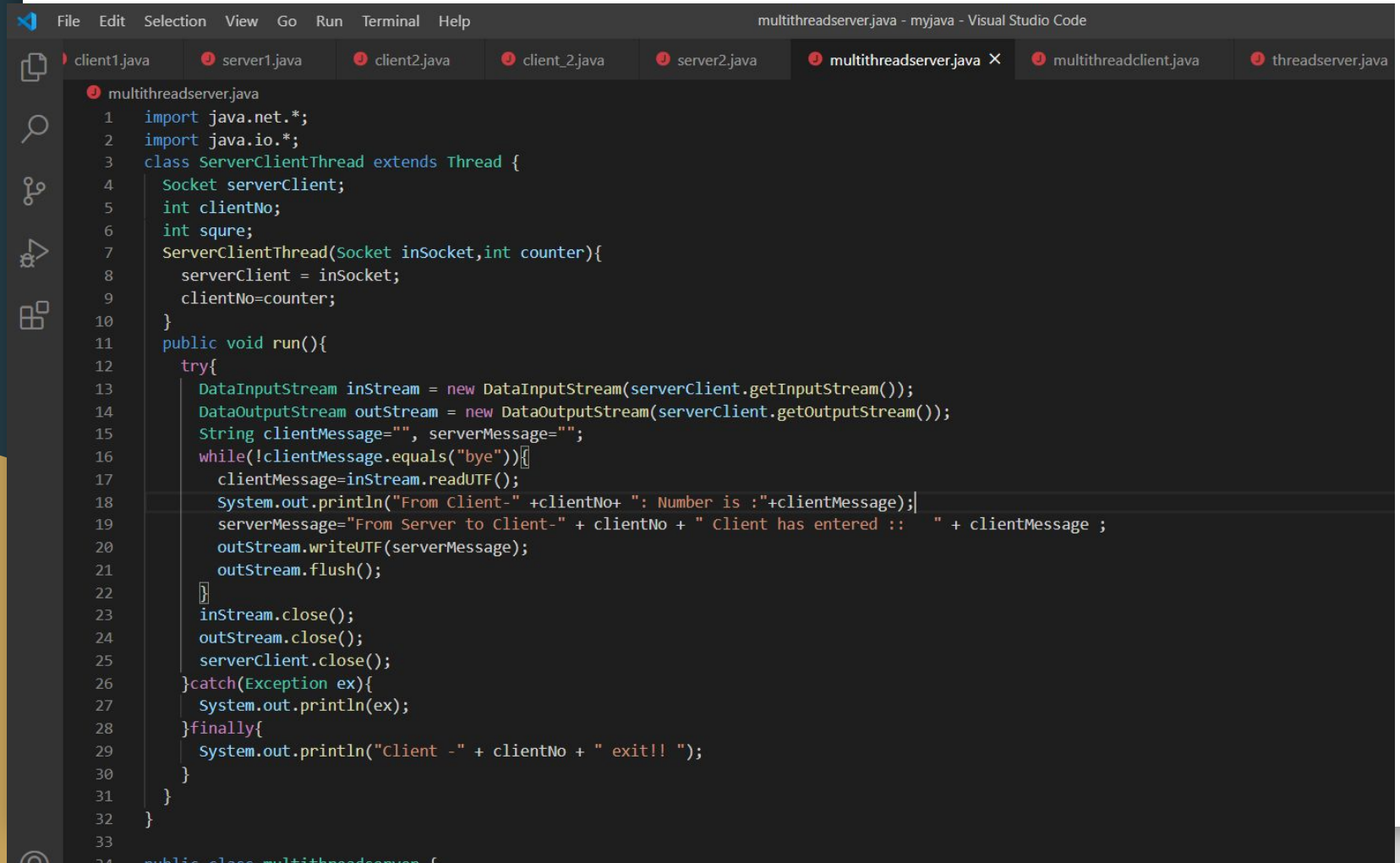
OUTPUT


Command Prompt
Connection closed
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java cthread1
Write anything ... Type Exit to terminate connection
hello server
Server Says :: hello client
hello server
Exit
Terminating Connection :: Socket[addr=localhost/127.0.0.1,port=5056,localport=64600]
Connection closed
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>


Command Prompt
Connection closed
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java cthread2
Write anything ... Type Exit to terminate connection
hello i am client 2
Server Says :: hello client
hello i am client 2
Exit
Terminating Connection :: Socket[addr=localhost/127.0.0.1,port=5056,localport=64602]
Connection closed
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>


Command Prompt - java threadserver
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\sasi>cd C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava\
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>javac threadserver.java
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java threadserver
A new client is connected :: Socket[addr=/127.0.0.1,port=64600,localport=5056]
Assigning new thread for this client
Server Message = ::
hello client
A new client is connected :: Socket[addr=/127.0.0.1,port=64602,localport=5056]
Assigning new thread for this client
Server Message = ::
hello client
Client Socket[addr=/127.0.0.1,port=64600,localport=5056] has sent exit ...
Terminating connection
Closed Connection
Client Socket[addr=/127.0.0.1,port=64602,localport=5056] has sent exit ...
Terminating connection
Closed Connection

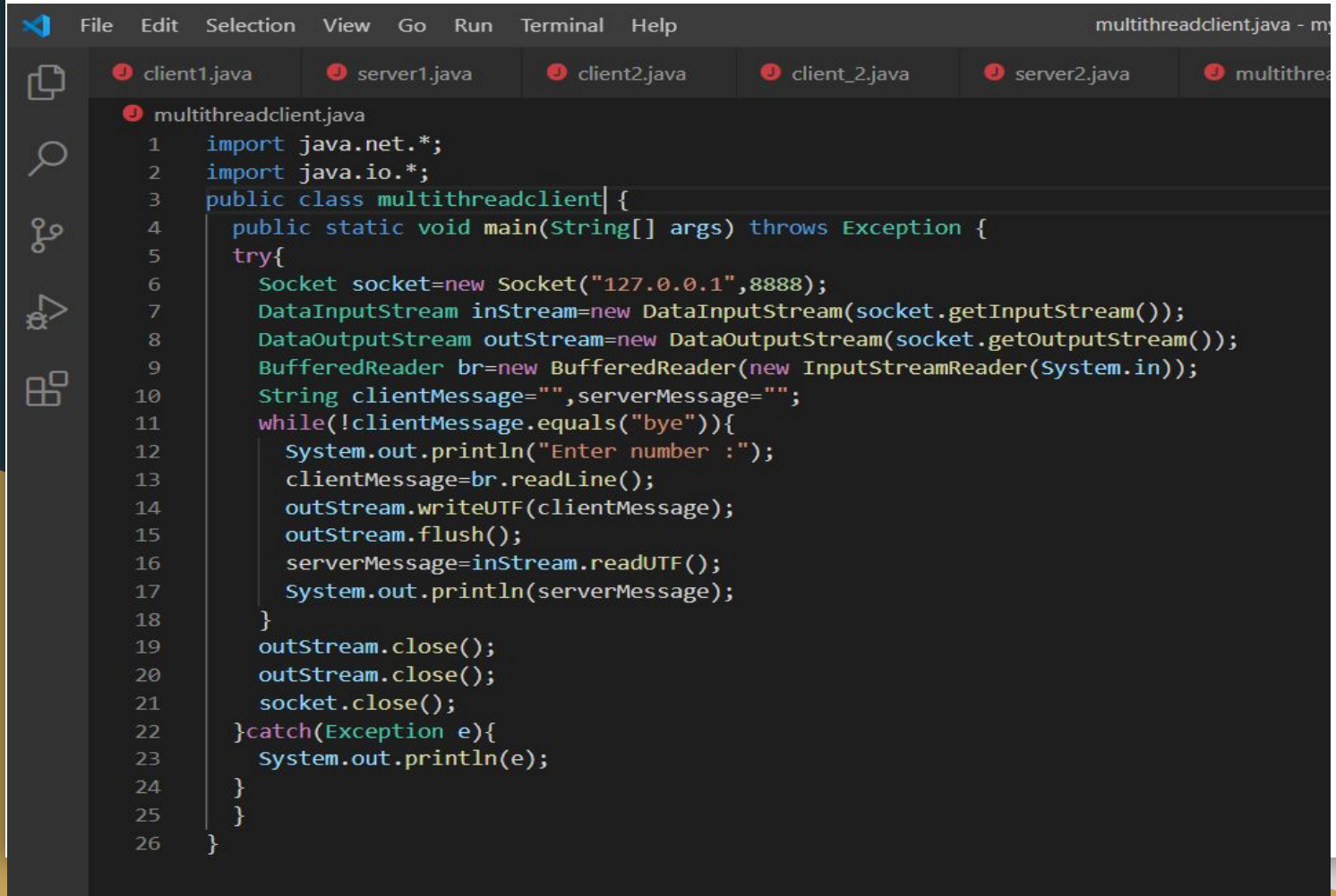
4: SERVER


File Edit Selection View Go Run Terminal Help
multithreadserver.java - myjava - Visual Studio Code
client1.java
server1.java
client2.java
client_2.java
server2.java
multithreadserver.java
multithreadclient.java
threadserver.java
multithreadserver.java
1 import java.net.*;
2 import java.io.*;
3 class ServerClientThread extends Thread {
4 Socket serverClient;
5 int clientNo;
6 int squre;
7 ServerClientThread(Socket inSocket,int counter){
8 serverClient = inSocket;
9 clientNo=counter;
10 }
11 public void run(){
12 try{
13 DataInputStream inStream = new DataInputStream(serverClient.getInputStream());
14 DataOutputStream outStream = new DataOutputStream(serverClient.getOutputStream());
15 String clientMessage="", serverMessage="";
16 while(!clientMessage.equals("bye")){
17 clientMessage=inStream.readUTF();
18 System.out.println("From Client-" +clientNo+ ": Number is :"+clientMessage);
19 serverMessage="From Server to Client-" + clientNo + " Client has entered :: " + clientMessage ;
20 outStream.writeUTF(serverMessage);
21 outStream.flush();
22 }
23 inStream.close();
24 outStream.close();
25 serverClient.close();
26 }catch(Exception ex){
27 System.out.println(ex);
28 }finally{
29 System.out.println("Client -" + clientNo + " exit!! ");
30 }
31 }
32 }
33

multithreadserver.java

```
26        }catch(Exception ex){
27          System.out.println(ex);
28        }finally{
29          System.out.println("Client -" + clientNo + " exit!! ");
30        }
31      }
32    }
33
34    public class multithreadserver {
35      public static void main(String[] args) throws Exception {
36        try{
37          ServerSocket server=new ServerSocket(8888);
38          int counter=0;
39          System.out.println("Server Started ....");
40          while(true){
41            counter++;
42            Socket serverClient=server.accept();
43            System.out.println(" >> " + "Client No:" + counter + " started!");
44            ServerClientThread sct = new ServerClientThread(serverClient,counter);
45            sct.start();
46          }
47        }catch(Exception e){
48          System.out.println(e);
49        }
50      }
51    }
```

CLIENT


File Edit Selection View Go Run Terminal Help
multithreadclient.java - my
client1.java
server1.java
client2.java
client_2.java
server2.java
multithrea
multithreadclient.java
1 import java.net.*;
2 import java.io.*;
3 public class multithreadclient {
4 public static void main(String[] args) throws Exception {
5 try{
6 Socket socket=new Socket("127.0.0.1",8888);
7 DataInputStream inStream=new DataInputStream(socket.getInputStream());
8 DataOutputStream outStream=new DataOutputStream(socket.getOutputStream());
9 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
10 String clientMessage="",serverMessage="";
11 while(!clientMessage.equals("bye")){
12 System.out.println("Enter number :");
13 clientMessage=br.readLine();
14 outStream.writeUTF(clientMessage);
15 outStream.flush();
16 serverMessage=inStream.readUTF();
17 System.out.println(serverMessage);
18 }
19 outStream.close();
20 outStream.close();
21 socket.close();
22 }catch(Exception e){
23 System.out.println(e);
24 }
25 }
26 }

OUTPU

Command Prompt

```
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java multithreadclient
Enter number :
123
From Server to Client-1 Client has entered ::   123
Enter number :
34
From Server to Client-1 Client has entered ::   34
Enter number :
167
From Server to Client-1 Client has entered ::   167
Enter number :
bye
From Server to Client-1 Client has entered ::   bye

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>
```

Command Prompt - java multithreadserver

```
C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java multithreadserver
Server Started ....
 >> Client No:1 started!
From Client-1: Number is :123
From Client-1: Number is :34
From Client-1: Number is :167
 >> Client No:2 started!
From Client-2: Number is :890
From Client-2: Number is :1000
From Client-1: Number is :bye
Client -1 exit!!
From Client-2: Number is :12345
From Client-2: Number is :66
From Client-2: Number is :bye
Client -2 exit!!
 >> Client No:3 started!
 >> Client No:4 started!
java.io.EOFException
Client -3 exit!!
 >> Client No:5 started!
java.io.EOFException
Client -4 exit!!
 >> Client No:6 started!
java.io.EOFException
Client -5 exit!!
java.io.EOFException
Client -6 exit!!
 >> Client No:7 started!
 >> Client No:8 started!
```

Command Prompt

```
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\sasi>cd C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava\

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>javac multithreadclient.java

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>java multithreadclient
Enter number :
890
From Server to Client-2 Client has entered ::   890
Enter number :
1000
From Server to Client-2 Client has entered ::   1000
Enter number :
12345
From Server to Client-2 Client has entered ::   12345
Enter number :
66
From Server to Client-2 Client has entered ::   66
Enter number :
bye
From Server to Client-2 Client has entered ::   bye

C:\Users\sasi\java\Java\jdk1.8.0_241\bin\myjava>
```

LOCALHOST ERROR