PRACTICAL-4

AIM: Implement Concurrent TCP Server programming in which more than one client can connect and communicate with Server for sending the string and server returns the reverse of string to each of client.

PROGRAM:

- **Server:**

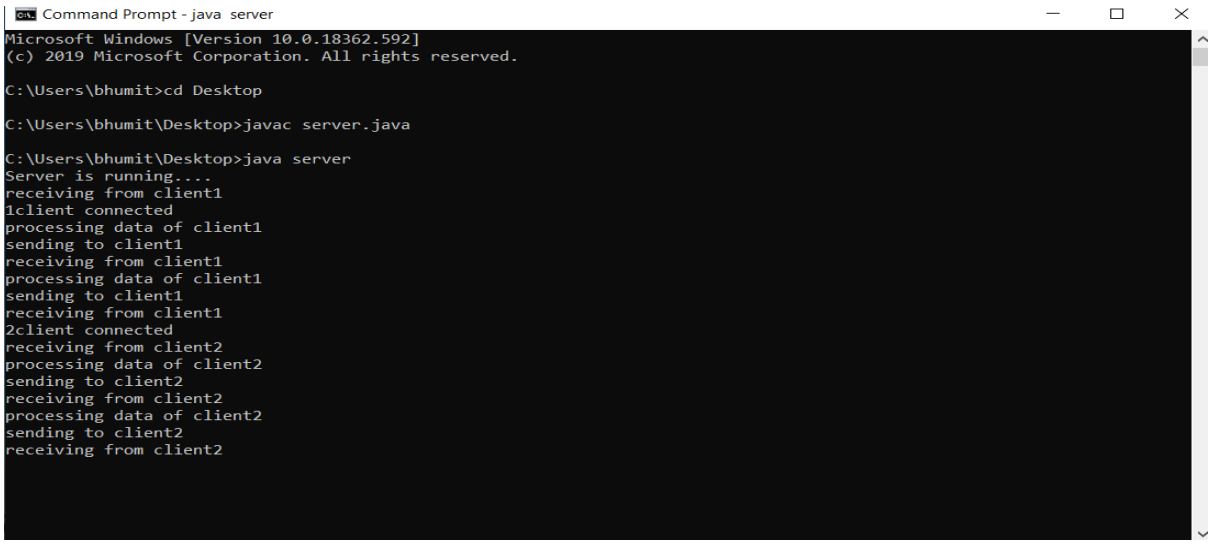
```
 import java.net.*;
import java.io.*;
public class server
{
  public static void main(String[] args) throws Exception
   {
       int count=1;
       System.out.println("Server is running....");
       ServerSocket ss=new ServerSocket(7878);
       while(true)
       {
        new RevThread(ss.accept(),count).start();
        System.out.println(count+"client connected");
        count++;
       }
   }
}

class RevThread extends Thread

{
   Socket s=null;
   int n;
   public RevThread(Socket socket,int count)
    {
      s=socket;
      n=count;
    }
  public void run()
   {
    try
     {
         while(true)
          {
             System.out.println("receiving from client"+n);
             DataInputStream din=new DataInputStream(s.getInputStream());
             String str=din.readUTF();
             System.out.println("processing data of client"+n);
             StringBuffer rev=new StringBuffer();
             rev=rev.append(str);
```

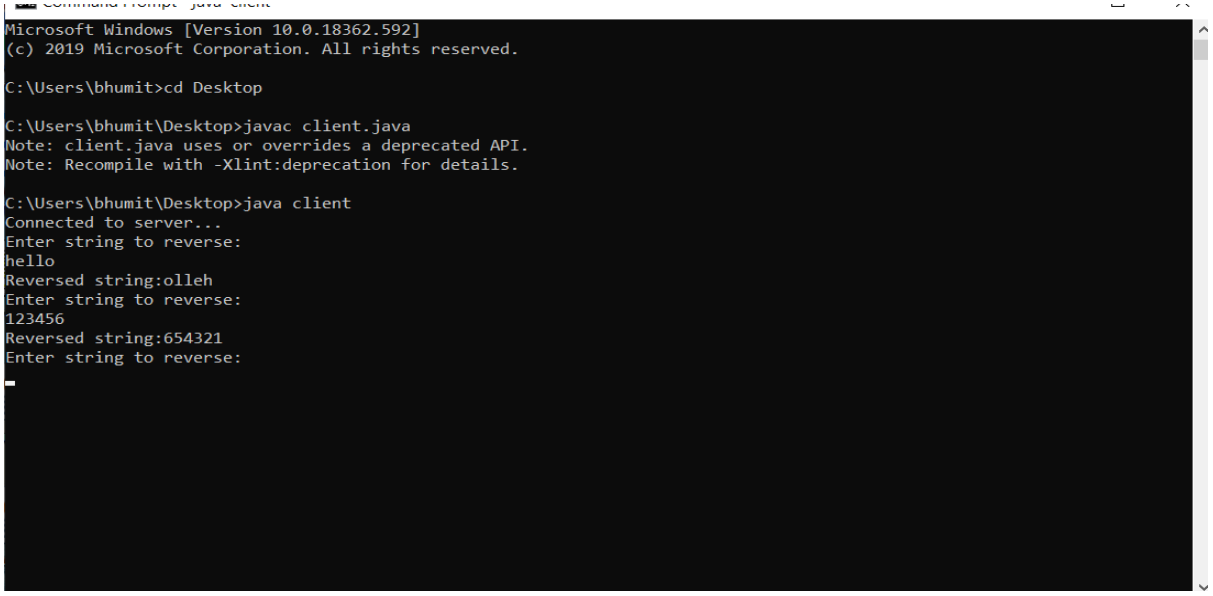
```
                rev=rev.reverse();
                String revStr=new String(rev);
                System.out.println("sending to client"+n);
                DataOutputStream dout=new DataOutputStream(s.getOutputStream());
                dout.writeUTF(revStr);
            }  }
        catch(IOException e)
        {
            System.out.println(e);
        }  } }
```

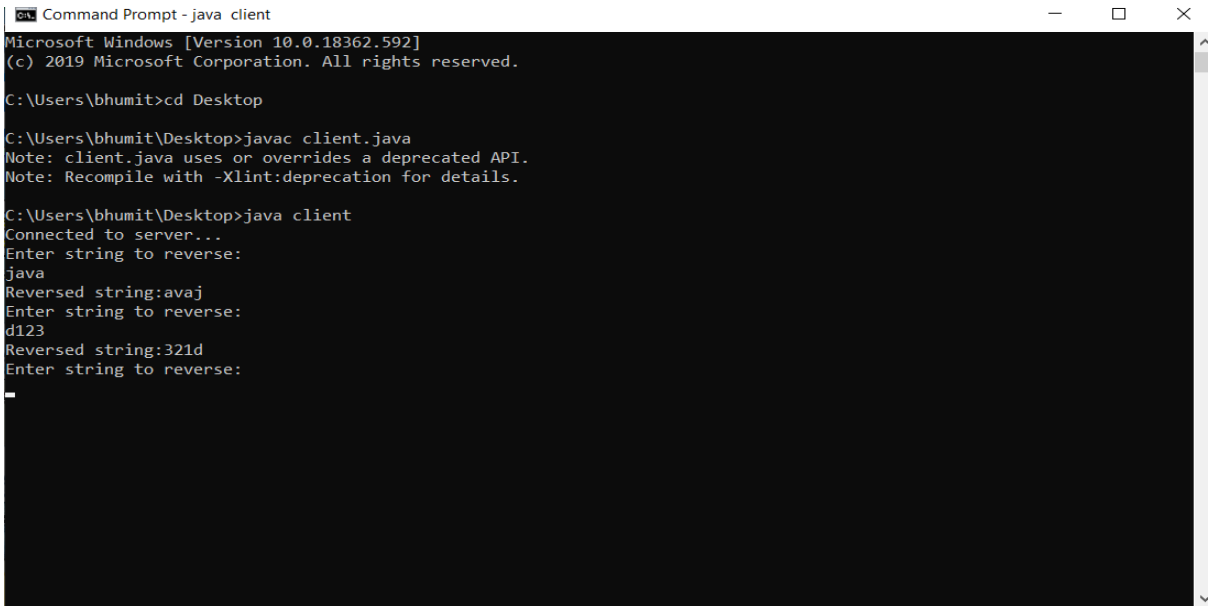
- **Client:**

```
import java.net.*;
import java.io.*;
public class client
{
   public static void main(String[] args) throws Exception
    {
      Socket s=new Socket("127.0.0.1",7878);
      if(s.isConnected())
       {
         System.out.println("Connected to server...");
       }
      while(true)
       {
         System.out.println("Enter string to reverse:");
         DataInputStream in=new DataInputStream(System.in);
         String str=in.readLine();
         DataOutputStream dout=new DataOutputStream(s.getOutputStream());
         dout.writeUTF(str);
         DataInputStream din=new DataInputStream(s.getInputStream());
         String rev=din.readUTF();
         System.out.println("Reversed string:"+rev);
       } } }
```

OUTPUT:


Command Prompt - java server
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\bhumit>cd Desktop
C:\Users\bhumit\Desktop>javac server.java
C:\Users\bhumit\Desktop>java server
Server is running....
receiving from client1
1client connected
processing data of client1
sending to client1
receiving from client1
processing data of client1
sending to client1
receiving from client1
2client connected
receiving from client2
processing data of client2
sending to client2
receiving from client2
processing data of client2
sending to client2
receiving from client2


Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\bhumit>cd Desktop
C:\Users\bhumit\Desktop>javac client.java
Note: client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
C:\Users\bhumit\Desktop>java client
Connected to server...
Enter string to reverse:
hello
Reversed string:olleh
Enter string to reverse:
123456
Reversed string:654321
Enter string to reverse:


Command Prompt - java client
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\bhumit>cd Desktop
C:\Users\bhumit\Desktop>javac client.java
Note: client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
C:\Users\bhumit\Desktop>java client
Connected to server...
Enter string to reverse:
java
Reversed string:avaj
Enter string to reverse:
d123
Reversed string:321d
Enter string to reverse: