

Example # 01 (TCP, Simple Client/Server communication)

s.java

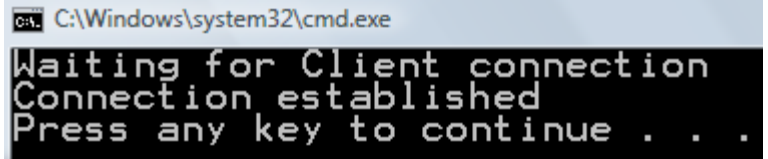
```
import java.io.*; import java.net.*;

public class s{
    public static void main(String args[]) throws IOException{
        ServerSocket s=new ServerSocket(1000);
        System.out.println("Waiting for Client connection");
        Socket c=s.accept();
        System.out.println("Connection established");

        ObjectOutputStream oos=new ObjectOutputStream(c.getOutputStream());
        String msg="Message from Server machine";

        oos.writeObject(msg);

        c.close();
    }
}
```

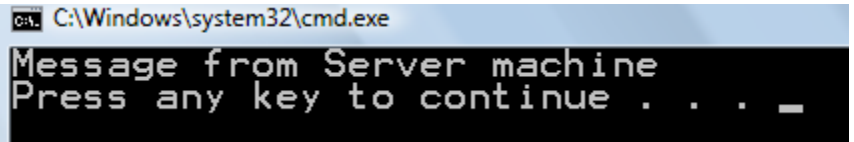


c.java

```
import java.io.*; import java.net.*;

public class c{
    public static void main(String args[]) throws IOException, ClassNotFoundException{
        Socket c=new Socket("127.0.0.1",1000);

        ObjectInputStream ois=new ObjectInputStream(c.getInputStream());
        String m=(String) ois.readObject();
        System.out.println(m);
        ois.close();
        c.close();
    }
}
```



Example # 02 (TCP, Simple Date & Time Client/Server)*SimpleTimeServer.java*

```

import java.net.*; import java.util.*; import java.io.*;

public class SimpleTimeServer
{
    int i_Socket_Number = 8001;
    public SimpleTimeServer ()
    {
        Socket clientSocket = null;
        try
        {
            ServerSocket serverSocket = new ServerSocket(i_Socket_Number);
            while (serverSocket != null)
            {
                try
                {
                    Scket = serverSocket.accept();
                    System.out.println("Connection Established");
                    PrintStream ps = new PrintStream(Scket.getOutputStream());
                    Date nowDate = new Date();
                    ps.write( nowDate.toString().getBytes() );
                    Scket.close();
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                    serverSocket = null;
                }
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    public static void main(String args[])
    {
        new SimpleTimeServer();
    }
}

```

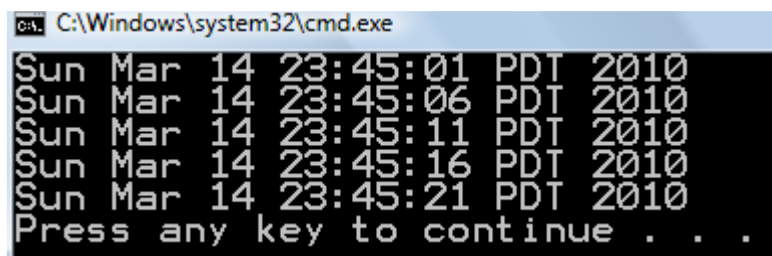
SimpleClient.java

```

import java.net.*; import java.io.*;

public class SimpleClient
{
    String s_Host = "localhost";
    int i_Port = 8001;
    public SimpleClient()
    {
        for (int i = 0 ; i < 5 ; i++)
        {
            try
            {
                Socket socket = new Socket(s_Host, i_Port);
                if (socket != null)
                {
                    InputStream is = socket.getInputStream();
                    BufferedReader bis=new BufferedReader(new InputStreamReader(is));
                    System.out.println(bis.readLine());
                    socket.close();
                }
            }
            catch (IOException ioe)
            {
                ioe.printStackTrace();
            }
            try
            {
                Thread.sleep(5000);
            }
            catch (InterruptedException ie)
            {
            }
        }
    }
    public static void main(String args[])
    {
        new SimpleClient();
    }
}

```



```

C:\Windows\system32\cmd.exe
Sun Mar 14 23:45:01 PDT 2010
Sun Mar 14 23:45:06 PDT 2010
Sun Mar 14 23:45:11 PDT 2010
Sun Mar 14 23:45:16 PDT 2010
Sun Mar 14 23:45:21 PDT 2010
Press any key to continue . . .

```

Example # 03 (TCP, Multithreaded sever)*Srvr.java*

```

import java.net.*;
import java.io.*;

public class Srvr {

    public static void main(String argv[]) {

        try {
            ServerSocket sSoc = new ServerSocket(2001);
            while(true) {
                Socket inSoc = sSoc.accept();
                SThread ST = new SThread(inSoc);
                ST.start();
            }
        } catch (Exception e) {
            System.out.println("Oh Dear! " + e.toString());
        } } /*end of method*/    }/*end of class*/

class SThread extends Thread {

    Socket threadSoc;

    SThread(Socket inSoc) {
        threadSoc = inSoc;
    }

    public void run() {
        try {
            PrintStream ps = new PrintStream(threadSoc.getOutputStream());
            for (int i=0; i < 100; i++) {
                ps.println(i);
            }
        } catch (Exception e) {
            System.out.println("Whoops! " +
                e.toString());
        }

        try {
            threadSoc.close();
        } catch (Exception e) {
            System.out.println("Oh no! " +
                e.toString());
        } } /*end of run method*/    }/*end of class*/

```

Clint.java

```
import java.net.*;
import java.io.*;

public class Clint {

    public static void main(String argv[]) {

        Socket Soc;
        BufferedReader in;
        String message;
        try {
            Soc = new Socket("127.0.0.1",2001);
            in = new BufferedReader(new InputStreamReader(Soc.getInputStream()));
            for (int i = 0; i < 100; i++) {
                message = in.readLine();
                System.out.println(message); }

        }
        catch (Exception e) {
            System.out.println("Died... " +
                e.toString());
        }

    }

}
```

Output:

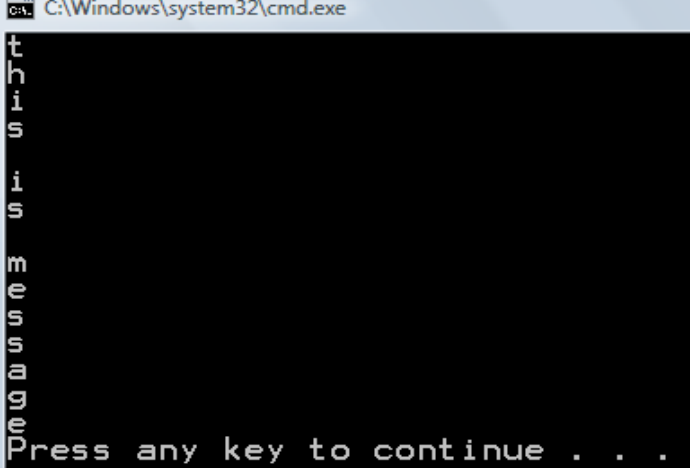
Will print 0-99 digits on client side.

Example # 04 (UDP, Simple sever and client application)***server1.java***

```
import java.io.*; import java.net.*;

public class server1{
    public static void main(String args[]) throws IOException{
        DatagramSocket s=new DatagramSocket(1000);
        byte b[]=new byte[100];

        DatagramPacket p=new DatagramPacket(b,b.length);
        s.receive(p);
        b=p.getData();
        int len=p.getLength();
        for(int i=0;i<len;i++){
            System.out.print((char) b[i]);
        }
    }
}
```



```
C:\Windows\system32\cmd.exe
t
h
i
s

i
s

m
e
s
s
a
g
e
Press any key to continue . . .
```

client1.java

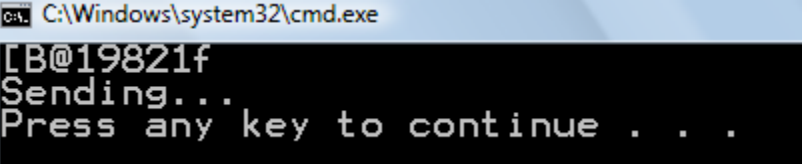
```
import java.io.*;
import java.net.*;

public class client1{
    public static void main(String args[]) throws IOException{
        DatagramSocket s=new DatagramSocket();
        String msg="this is message";

        byte b[]=msg.getBytes();

        System.out.println(msg.getBytes());

        DatagramPacket p=new DatagramPacket(b,b.length,InetAddress.getByName("127.0.0.1"),1000);
        System.out.println("Sending...");
        s.send(p);
    }
}
```



```
C:\Windows\system32\cmd.exe
[B@19821f
Sending...
Press any key to continue . . .
```