Datagram socket for client/server to display the messages on client side, typed at the server side.

Sender.java

```
import java.io.*;
import java.net.*;
public class Sender
{
    public static void main(String[] args) throws IOException
  {
      InetAddress addr = InetAddress.getByName(args[0]);
      byte[] buf = args[1].getBytes();
      DatagramPacket packet = new DatagramPacket(buf, buf.length, addr, 4444);
      DatagramSocket socket = new DatagramSocket();
      socket.send(packet);
  }
}
```

Receiver.java

```
import java.io.*;
import java.net.*;
public class Receiver
{
   public static void main(String[] args) throws IOException
  {
    DatagramSocket socket = new DatagramSocket(4444);
    byte[] buf = new byte[256];
    DatagramPacket packet = new DatagramPacket(buf, buf.length);
    System.out.println("Waiting ...");
    socket.receive(packet);
    String s = new String(packet.getData(), 0, packet.getLength());
    System.out.println(packet.getAddress().getHostName() + ": " + s);
 }
}
```

Datagram socket for client/server to display the messages on client side, typed at the server side.

- Compile the program.
- Start the receiver by running “java Receiver”.
- Assuming that the receiver is running on a host with IP address 127.0.0.1
  Start the sender by running:

java Sender 127.0.0.1 “My String”

- The receiver program should now display the string “My String”.
- Repeat this exercise, with the difference, that you run the sender and receiver on two different hosts.

Output:

```
krishna@ubuntu:~$ javac Sender.java
krishna@ubuntu:~$ java Sender 127.0.0.1 "Hello Ubuntu"
krishna@ubuntu:~$
```

```
krishna@ubuntu:~$ javac Receiver.java
krishna@ubuntu:~$ java Receiver
Waiting ...
localhost: Hello Ubuntu
krishna@ubuntu:~$
```