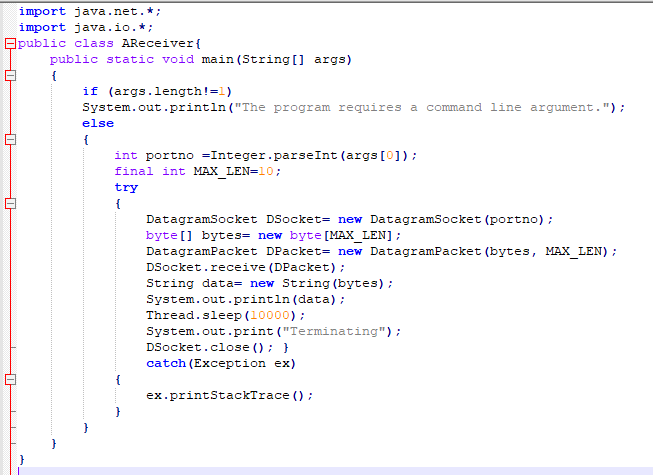
**LAB # 1**

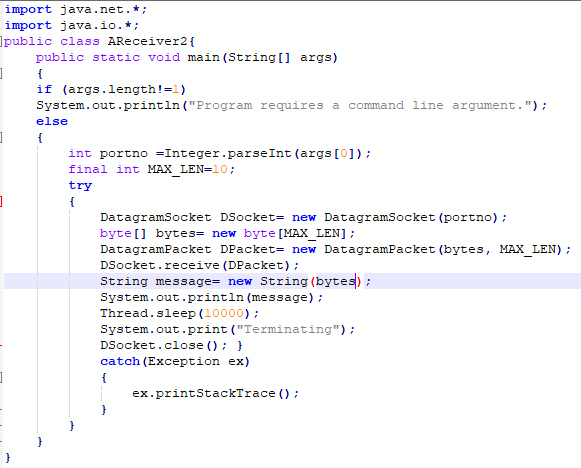
**OBJECT : To learn creation of datagram sockets.**

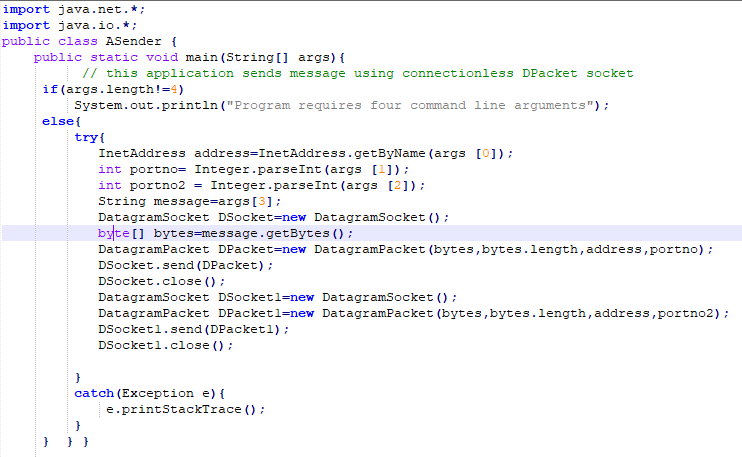
**Lab Tasks:**

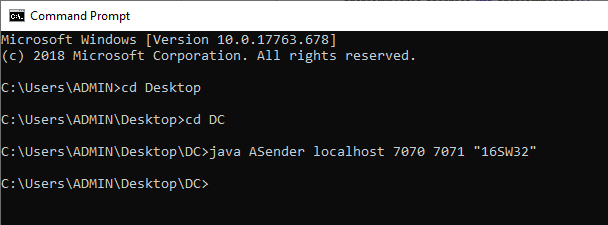
1. **Modify the sample code so that the sender uses the same socket to send the same message to two different receivers. Start the two receivers first, then the sender. Does each receiver receive the message? Capture the code and output. Describe the outcome.**

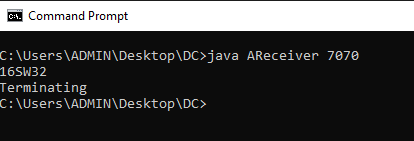
**Program:**

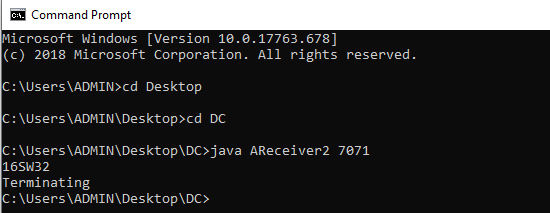












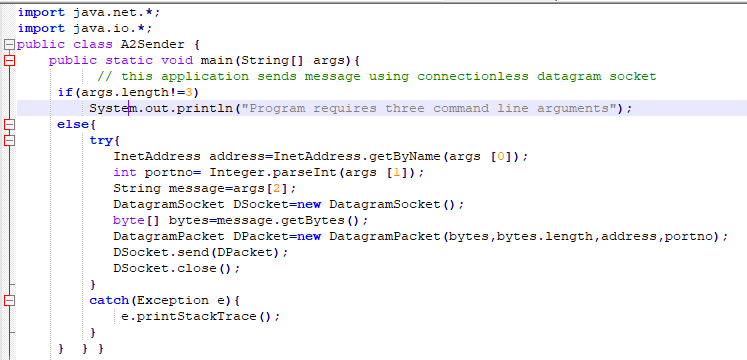
**2. Modify the sample code so that the receiver loops five times to repeatedly receive then display your bio data (name, roll num etc.) received. Recompile. Then**

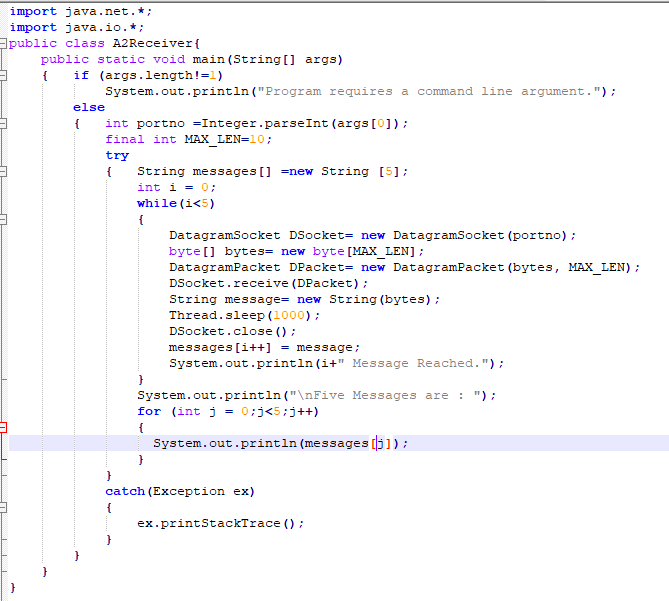
**i. start the receiver**

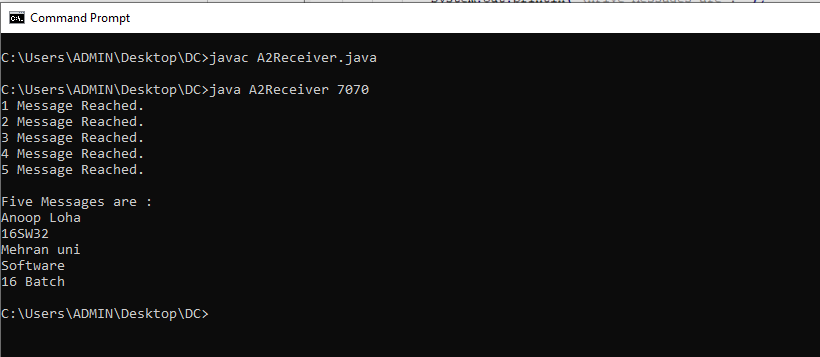
**ii. Execute the sender, sending your bio data, and**

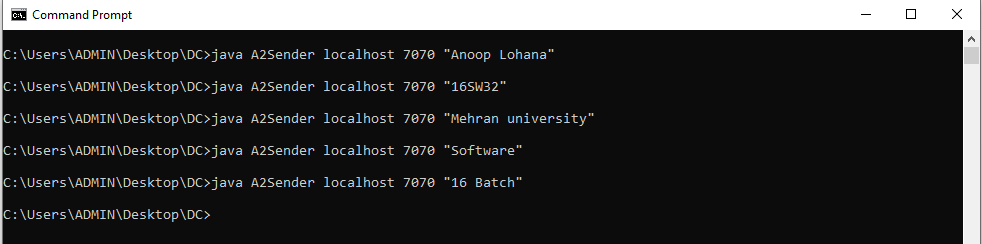
**iii. In another window, start another instance of the sender, sending your friend’s bio data. Does the receiver receive both the messages? Capture the code and output.**

**Program :**



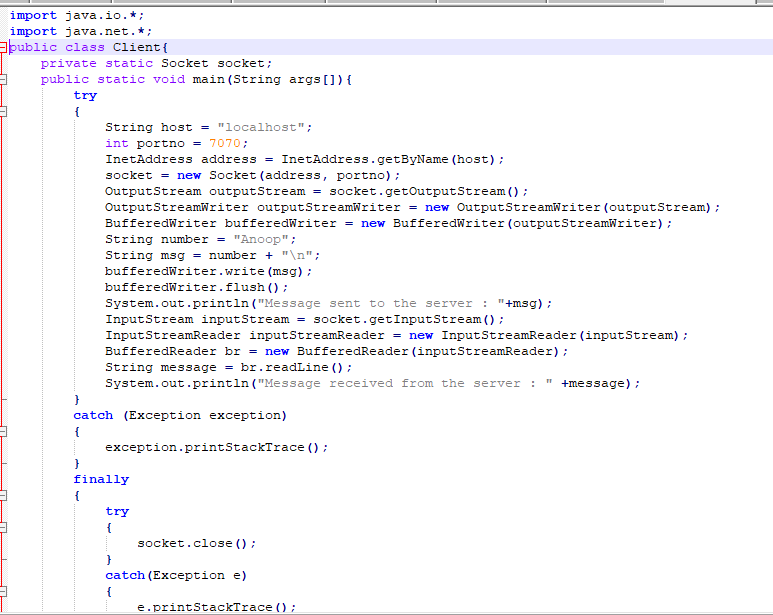




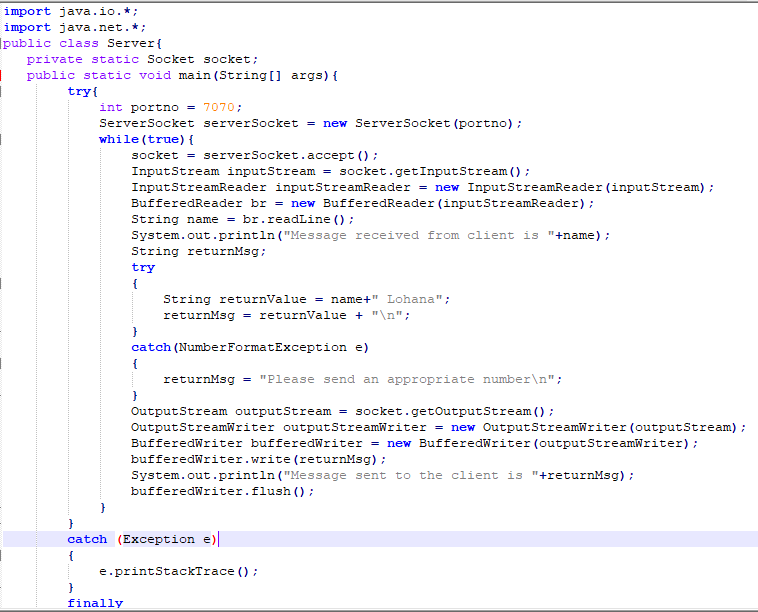


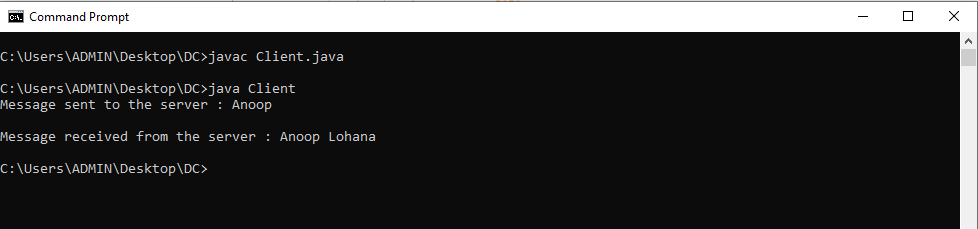
**3-Modify the sample code to cater to a two way communication i.e. Sender sends a message to the Receiver, the Receiver receives the message and sends a reply to the Sender in return.**

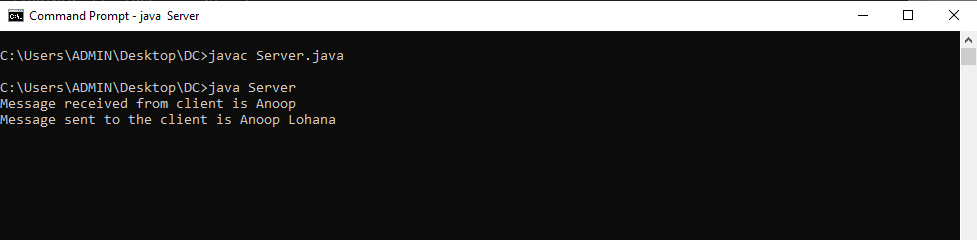
Client.java:



Server.java:



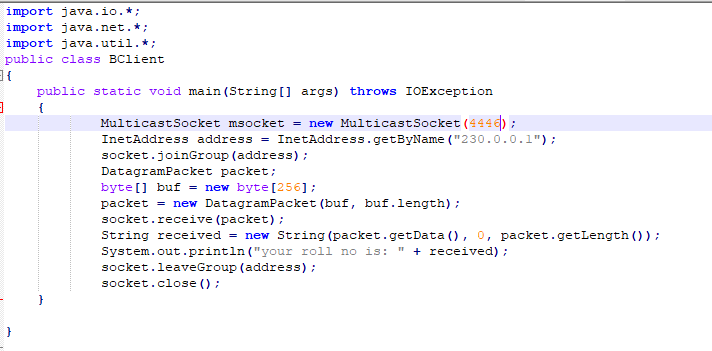




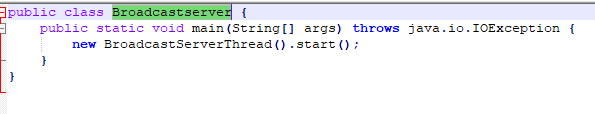
**4-Broadcasting: Broadcasting is a one-to-all type of communication, i.e. the intention is to send the datagram to all the nodes in the network. Unlike in the case of point-to-point communication, we don’t have to know the target host’s IP Address. Instead, a broadcast address is used.**

**Implement two simple programs using Java datagram sockets, which broadcasts and multicast your roll number to all or selected network nodes respectively.**

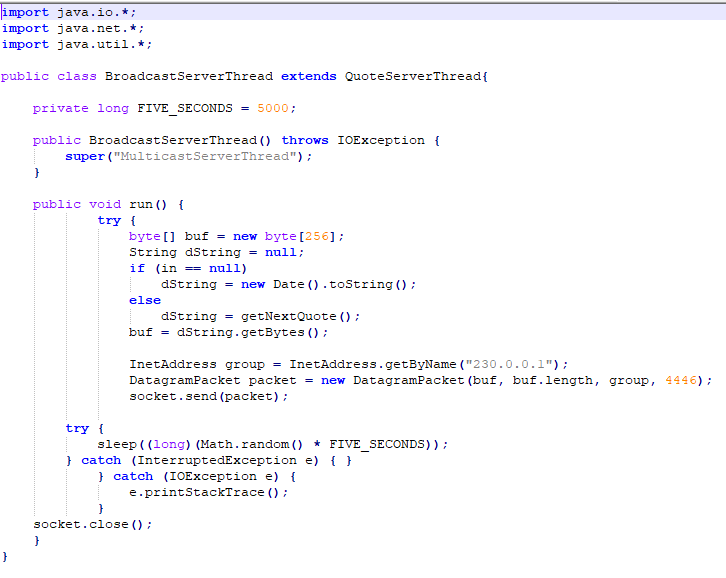
BroadcastClient.java:



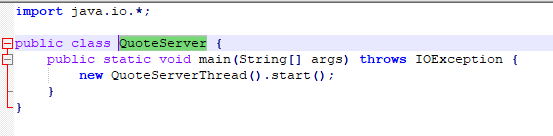
BroadcastServer.java:



BroadcastServerThread.java:



QuoteServer.java:



QuoteServerThread.java:

