**LAB: 01**

**To Become Familiar With Socket Programming**

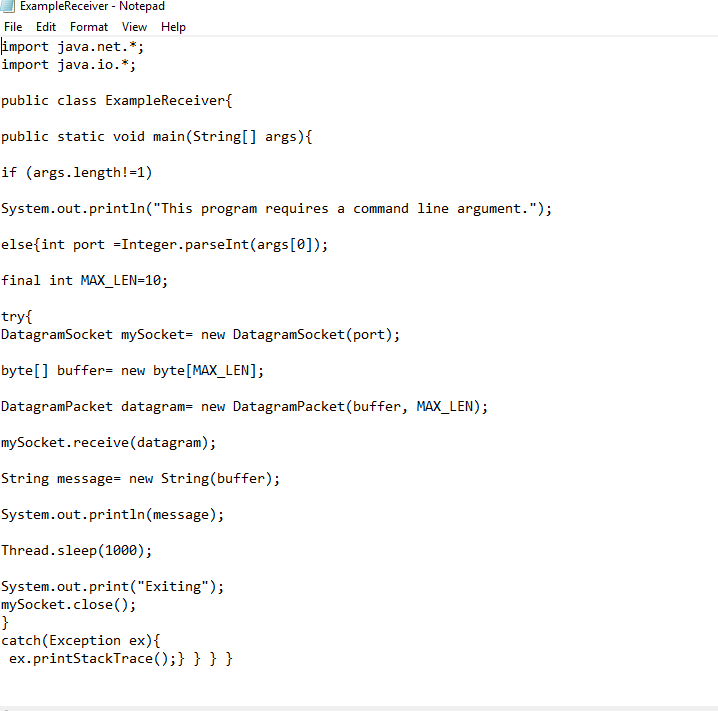
**(Client to Server).**

**Task-01: 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.**

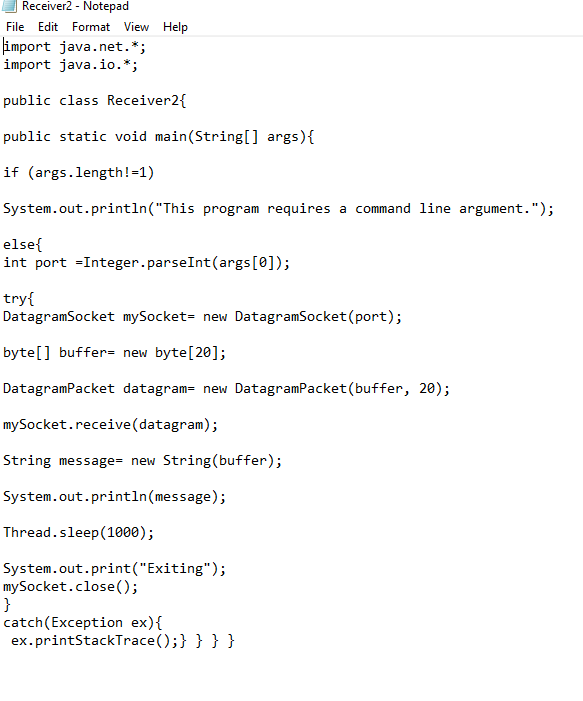
**Sender:**

****

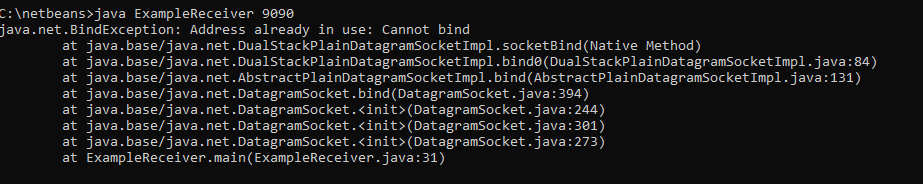
**Receiver 1:**

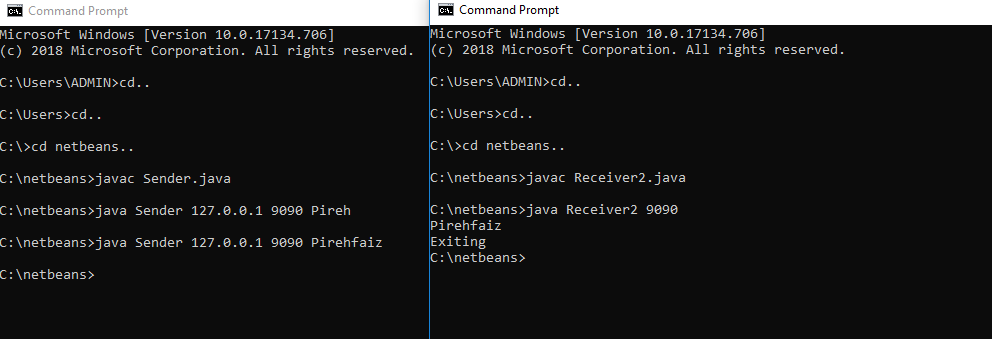
****

**Receiver 2:**

****

**Output:**

****

****

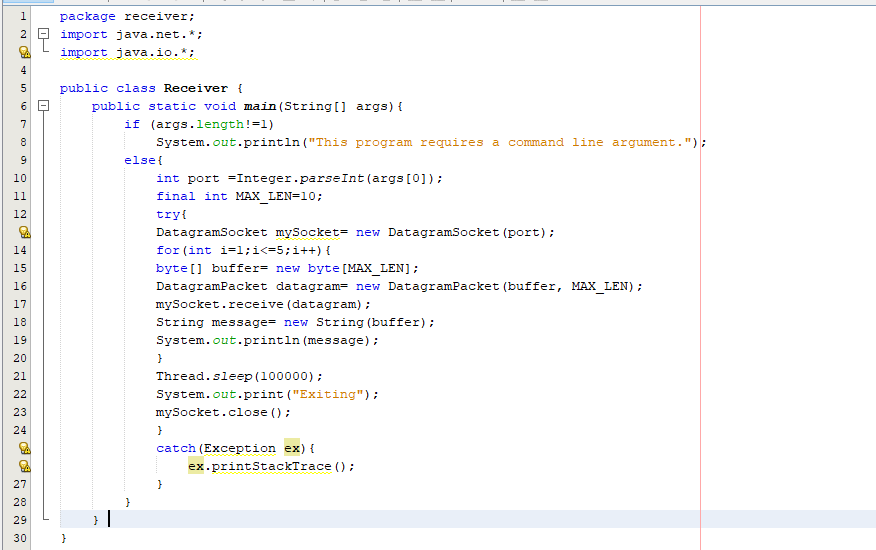
**Task-02: Modify the sample code so that the receiver loops five times to repeatedly receive then display the data received. Recompile. Then**

1. **start the receiver**
2. **Execute the sender, sending a message “message1”, and**
3. **In another window, start another instance of the sender, sending a message “message2”. Does the receiver receive both the messages? Capture the code and output.**

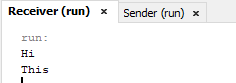
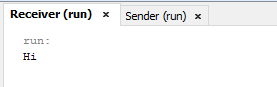
**Sender:**

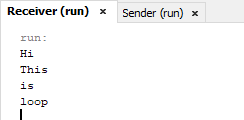
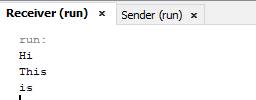


**Receiver:**

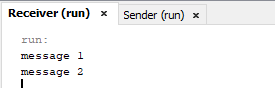
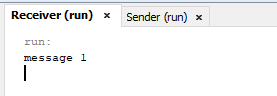


**Output:**





**After Recompiling:**



**Result:**

As we have increased the number of datagram’s to be received for each datagram sender has to send a different message as there is no connection built between the two and in this case different sender are sending the message as port number of receiver does not changes and is not confined to receive single message only.

**Task-03: 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.**

# **ChatClient:chat**

# ChatServer:

# Output:

