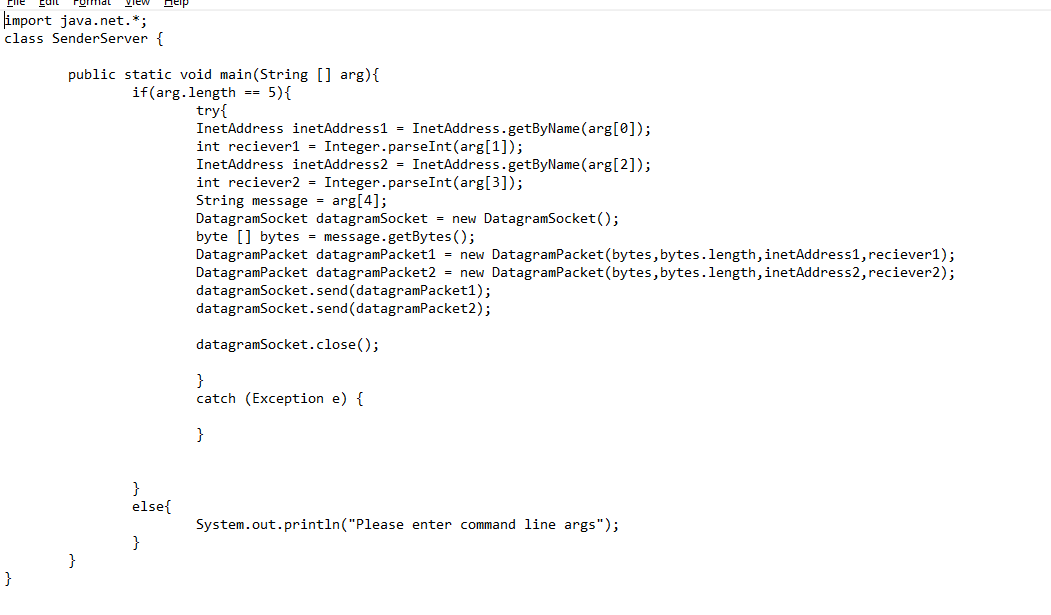
**LAB NO # 01**

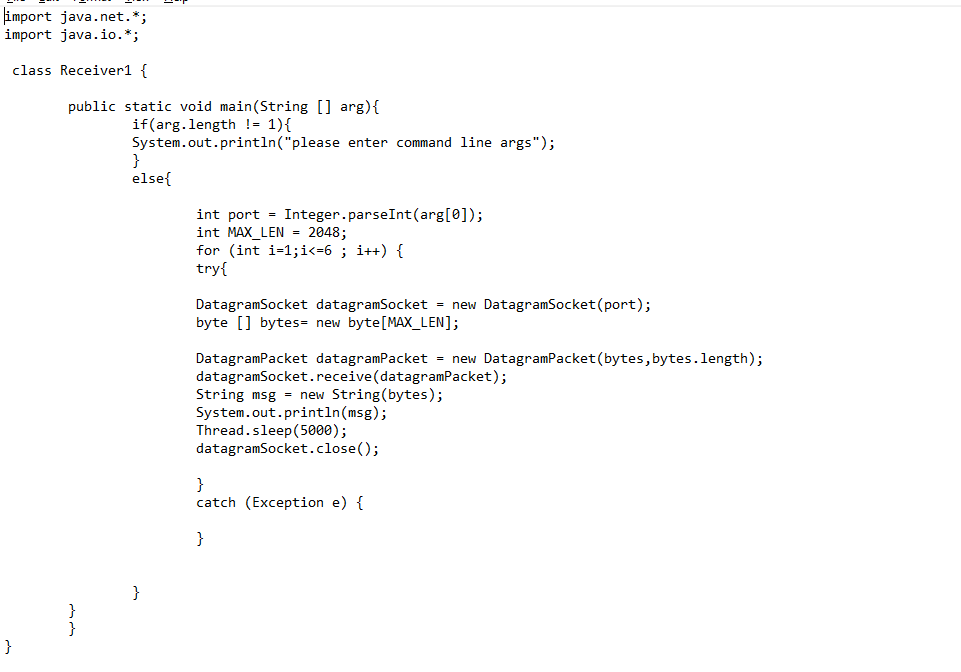
**OBJECT: Creating applications using Datagram sockets.**

**Task 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.

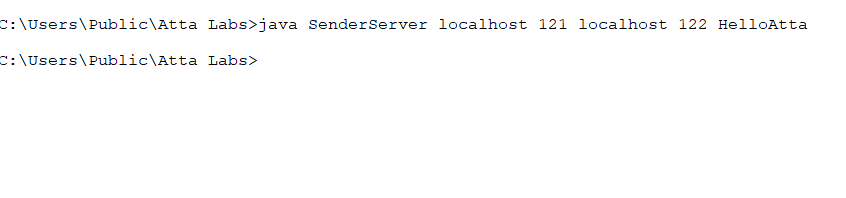
Server:

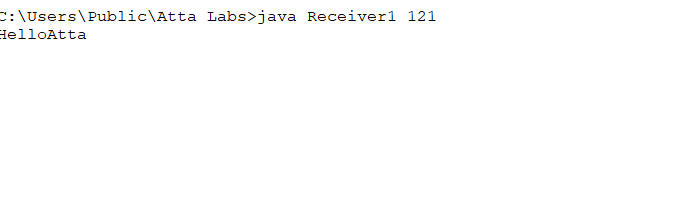
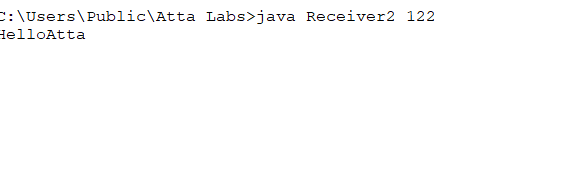


Client:



Output:



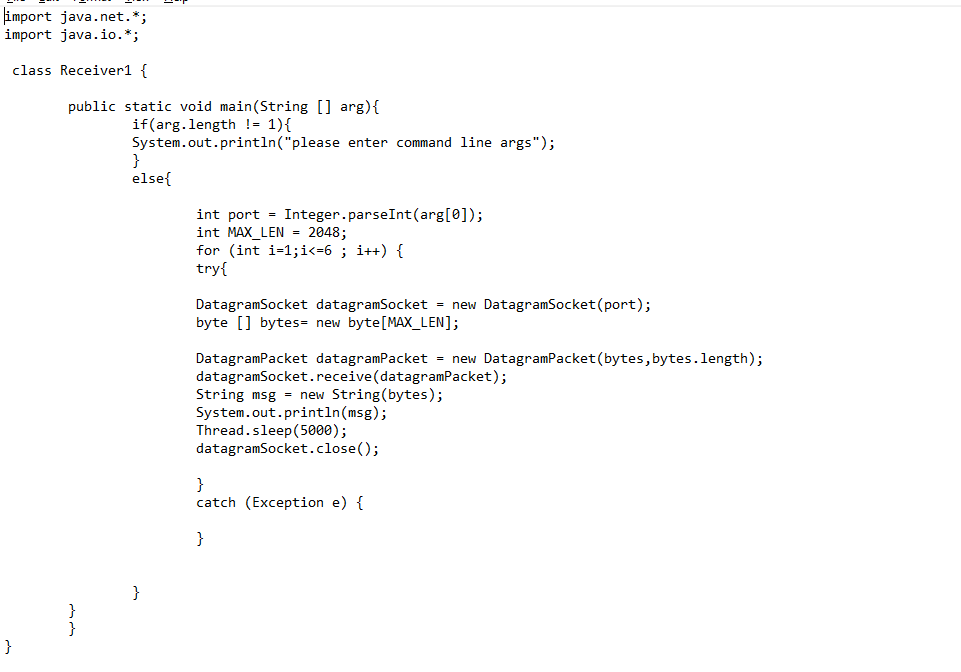


Task 2:Modify the sample code so that the receiver loops five times to repeatedly receive then display your bio data (name, roll num etc.) received. Recompile. Then  
i. start the receiver  
ii. Execute the sender, sending your bio data, and  
iii. In another window, start another instance of the sender, sending your friend’s bio data. Does the receiver receive both the messages? Capture the code and output.

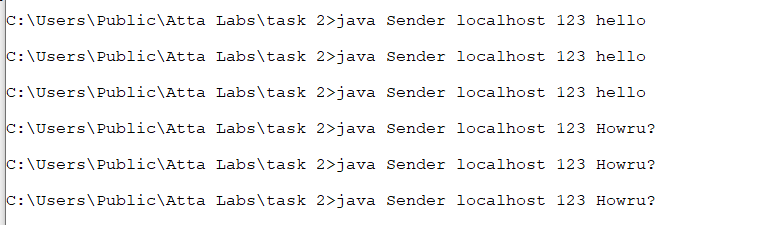
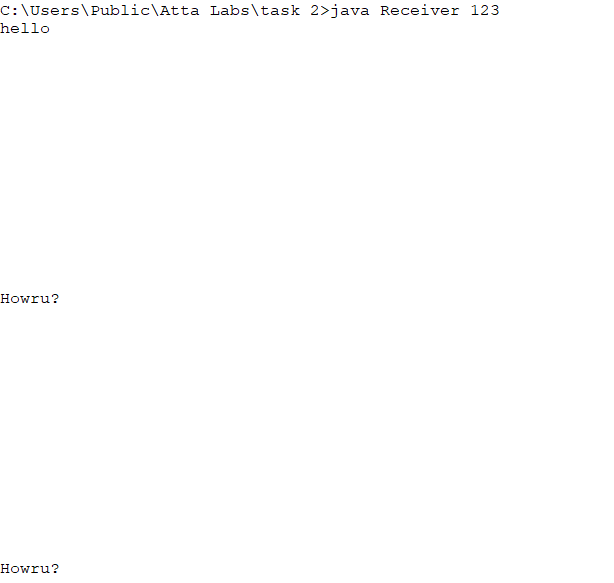
Sender.java



Receiver.java



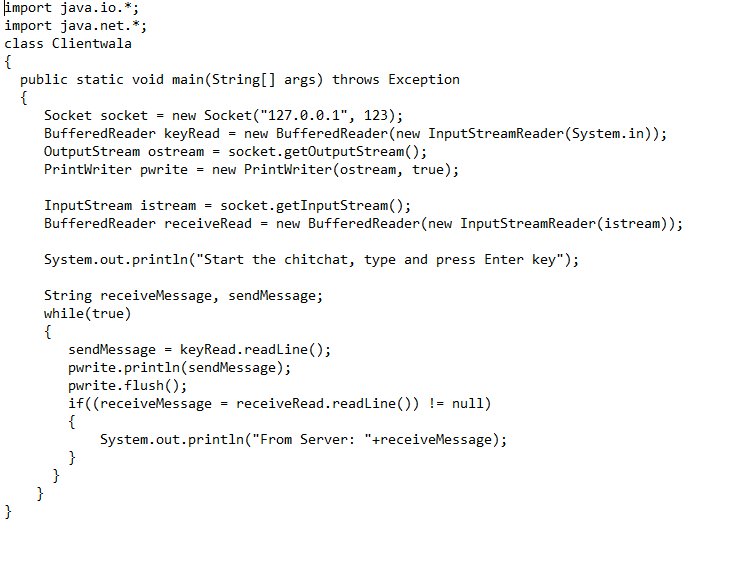
Output:



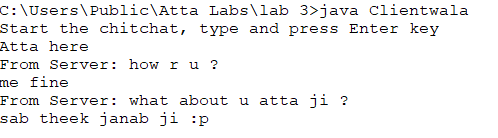
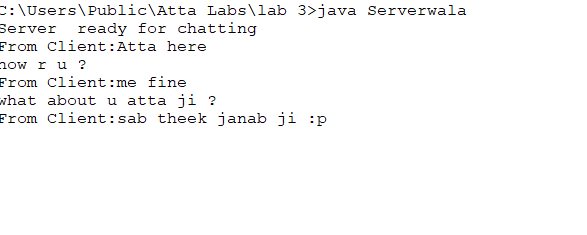
Task 3:Implement an encoding client-server application. The client sends a sentence to the server using datagram packet, the server accepts it and then encodes the sentence using any simple encoding algorithm (for e.g. letters reversal etc.) and sends the encoded sentence back to the client using a datagram packet.

Server.java

’Client.java



Output:

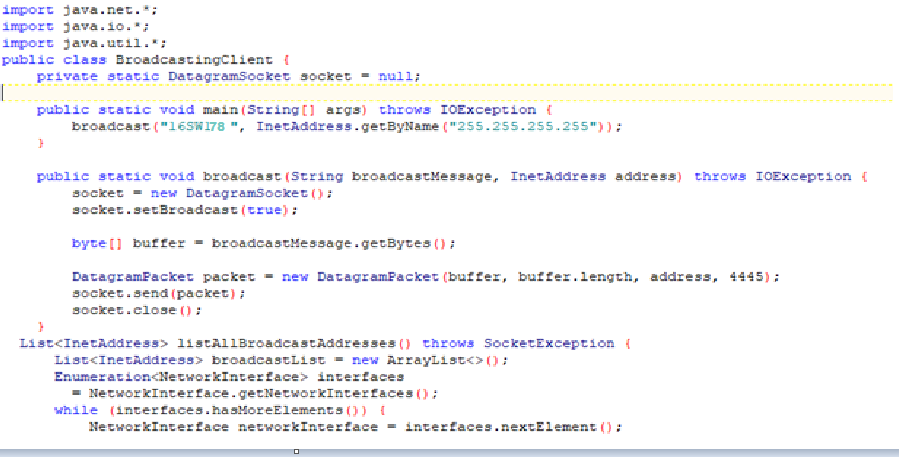


**TASK:4 (Broadcast and Multicast)**

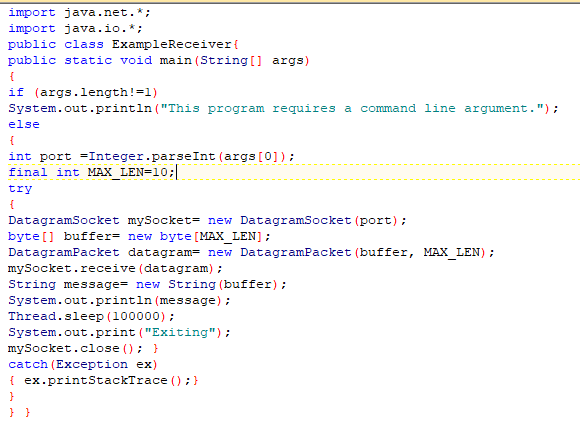
Implement two simple programs using Java datagram sockets, which broadcasts and multicast your roll number to all or selected network nodes respectively.

1)Broadcast program

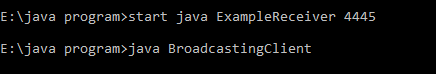
**BroadcastingClient.java**

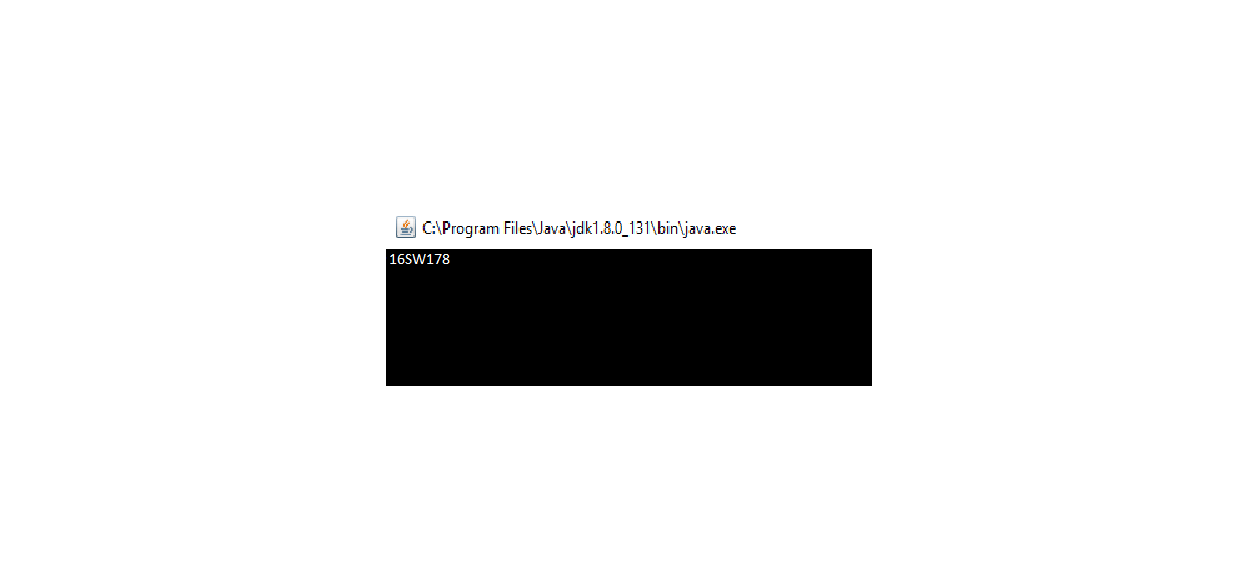


ExampleReceiver



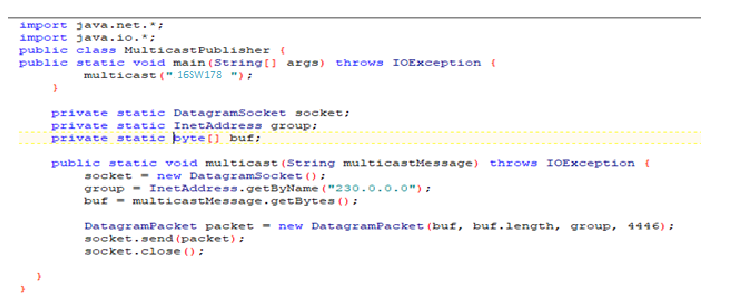
Output:



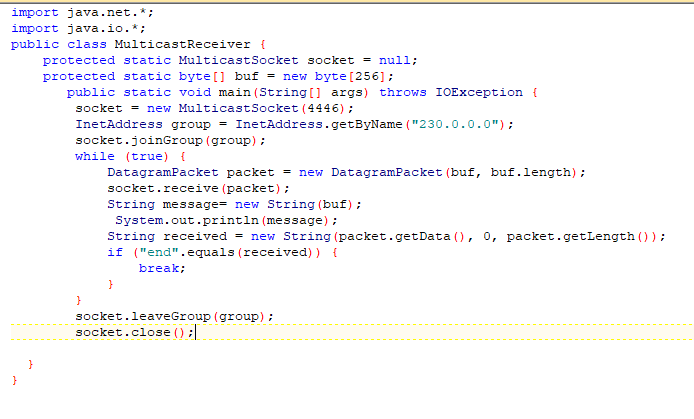


2)Multicast program

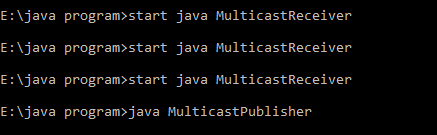
**MulticastPublisher.java**

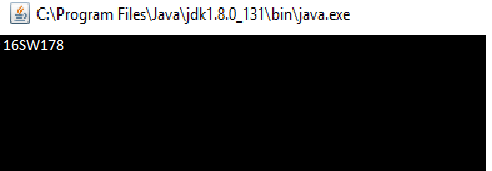
****

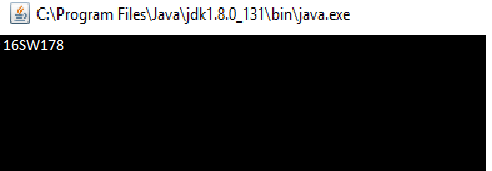
**MulticastReceiver.java**



Output:





****

**The End.**