Experiment No: 4

```
import javax.jms.*;
import org.apache.activemq.ActiveMQConnectionFactory;
class Sender implements Runnable {
   public void run() {
      ConnectionFactory connectionFactory = new
ActiveMQConnectionFactory("tcp://localhost:61616");
      try {
         Connection connection = connectionFactory.createConnection();
         connection.start();

         Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
         Topic topic = session.createTopic("MyTopic");

         MessageProducer producer = session.createProducer(topic);
         TextMessage message = session.createTextMessage("Hello, Subscribers!");

         producer.send(message);
         System.out.println("Message sent: " + message.getText());

         producer.close();
         session.close();
         connection.close();
      } catch (JMSException e) {
         e.printStackTrace();
      }
   }
}
```

```
class Subscriber implements Runnable {
    public void run() {
        ConnectionFactory connectionFactory = new
ActiveMQConnectionFactory("tcp://localhost:61616");
        try {
            Connection connection = connectionFactory.createConnection();
            connection.start();

            Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
            Topic topic = session.createTopic("MyTopic");

            MessageConsumer consumer = session.createConsumer(topic);
            consumer.setMessageListener(new MessageListener() {
                @Override
                public void onMessage(Message message) {
                    if (message instanceof TextMessage) {
                        TextMessage textMessage = (TextMessage) message;
                        try {
                            System.out.println("Received message: " + textMessage.getText());
                        } catch (JMSException e) {
                            e.printStackTrace();
                        }
                    }
                }
            });
            System.out.println("Waiting for messages...");
            Thread.sleep(10000);
            consumer.close();
            session.close();
            connection.close();
        } catch (JMSException | InterruptedException e) {
```

```
            e.printStackTrace();
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Runnable sender = new Sender();
        Runnable subscriber = new Subscriber();
        Thread t1 = new Thread(sender);
        Thread t2 = new Thread(subscriber);

        t1.start();
        t2.start();
        try {
            t1.join();
            t2.join();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
Message sent: Hello, Subscribers!
Waiting for messages...
Received message: Hello, Subscribers!
```