1. **Agenda**: We will see what is JMS?
2. **JSM:** Java Messaging Service.
3. **Application**: Using JMS, two applications can interact with each other.
4. **Real Life Example**: Two persons can communicate with each other using **mobile text service that is basically SMS**.
5. In the following, we have 4 java applications.
   1. App-1 is sending messages and others are receiving.  
      So, App-1 is producer and others are consumers.
   2. But App-1 can’t send messages directly to other apps. We need some **intermediates.** They are basically **message brokers**.
   3. So, in this course, we’re going to use RabbitMQ message broker.
   4. Now, here the messages from producer App-1 are routed through Exchange and Queue and then they goes to the **consumers**.  
      Exchange and Queue are part of RabbitMQ Message Broker.
6. There can be n number of consumers attached to Queue.
7. **Scenario to use JMS**: Suppose, you have e-commerce app site. When a user places an order,
   1. You want to send e-mail for the confirmation of the order and
   2. You want to set the shipping details with 3rd party courier.

If you do this within App-1, it would slow down the process and performance of the app.   
**Approach**: After placing an order, you can build a message and you can pass it to the consumers an then you can write the logic in the consumers to send out an email to the user for confirmation of the order and also you can write the logic to save the shipping details with the 3rd courier.

1. 