Example 9:

Implement an application that create an alert upon receiving a Message (SMS)

Broadcast Receivers:

Broadcast Receiver simply respond to broadcast messages from other applications or from the system itself. These messages are sometime called events or intents. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action.

There are following two important steps to make **BroadcastReceiver** works for the system broadcasted intents −

- Creating the Broadcast Receiver.
- Registering Broadcast Receiver

Creating the Broadcast Receiver

A broadcast receiver is implemented as a subclass of **BroadcastReceiver** class and overriding the onReceive() method where each message is received as a **Intent** object parameter.

Registering Broadcast Receiver

An application listens for specific broadcast intents by registering a broadcast receiver in AndroidManifest.xml file. Consider we are going to register MyReceiver for system generated event ACTION_BOOT_COMPLETED which is fired by the system once the Android system has completed the boot process.

Now whenever your Android device gets booted, it will be intercepted by BroadcastReceiver, MyReceiver and implemented logic inside onReceive() will be executed.

There are several system generated events defined as final static fields in the Intent class. The following table lists a few important system events.

- **android.intent.action.BATTERY_LOW**
  Indicates low battery condition on the device.

- **android.intent.action.BOOT_COMPLETED**
  This is broadcast once, after the system has finished booting.

- **android.intent.action.DATE_CHANGED**
  The date has changed.

- **android.intent.action.REBOOT**
  Have the device reboot.

In this example, we will see how to receive SMS messages

Receive SMS Permissions

We only need receive permission android.permission.RECEIVE_SMS. In case you also want to read SMS messages from the Inbox then you need android.permission.READ_SMS.

```
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

Intent Filter to receive SMS

We also need to tell Android that we want to handle incoming SMS messages. In order to do this, we will add a <receiver> to register a broadcast receiver to the manifest XML. We will also add an <intent-filter> to let Android know that we want to launch a specific class when an SMS comes in.

```
<receiver android:name="com.javarticles.android.SMSReceiver">
   <intent-filter>
      <action android:name="android.provider.Telephony.SMS_RECEIVED" />
   </intent-filter>
</receiver>
```

Broadcast Receiver

SMSReceiver is a **BroadcastReceiver**. When SMS is received, **onReceive()** will be called

```
public class SMSReceiver extends BroadcastReceiver {
   @Override
   public void onReceive(Context context, Intent intent) {
      Bundle bundle = intent.getExtras();
      Object[] pdus = (Object[]) bundle.get("pdus");
      SmsMessage[] messages = new SmsMessage[pdus.length];
         for (int i = 0; i < messages.length; i++)
         {
                  messages[i]= SmsMessage.createFromPdu((byte[]) pdus[i]);
                  String address = messages[i].getOriginatingAddress();
                  if (phoneEnrties.contains(address)) {
                  Intent newintent = new Intent(context, MainActivity.class);
                  newintent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                  newintent.putExtra("address", address);
                  newintent.putExtra("message",
                  messages[i].getDisplayMessageBody());
                  context.startActivity(newintent);
                  }
         }
   }
}
```