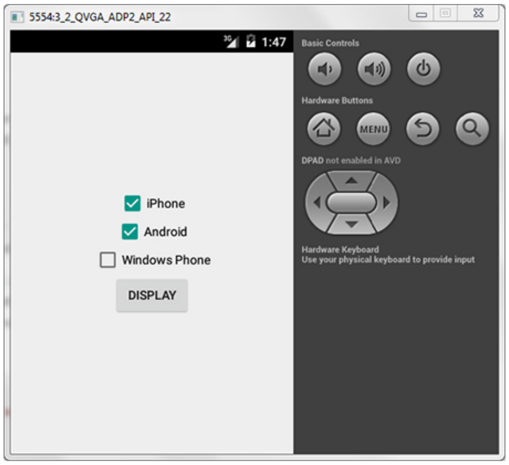
Slide 9.9: Radio buttons: Java source code
Slide 9.11: Checkboxes: layout file
Home

PRINT VERSION

Checkboxes

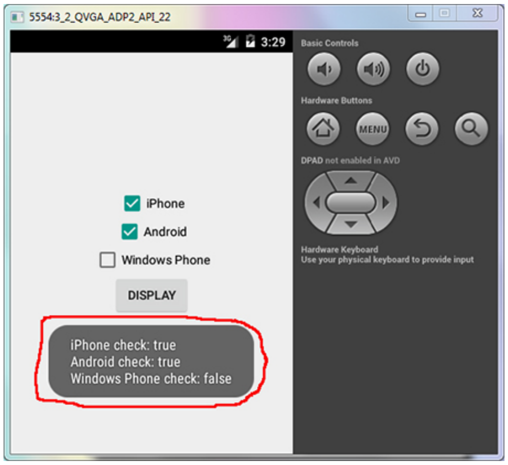


☑ iPhone
☑ Android
☐ Windows Phone
DISPLAY

Selecting iPhone & Android and

pushing DISPLAY

⇒



☑ iPhone
☑ Android
☐ Windows Phone
DISPLAY

iPhone check: true
Android check: true
Windows Phone check: false

Checkboxes allow the user to select one or more options from a set. To create each checkbox option, create a `Checkbox` in your layout. Because a set of checkbox options allows the user to select multiple items, each checkbox is managed separately and you must register a click listener for each one.

Responding to Click Events

However, this example uses another approach. When the user selects a checkbox, the `CheckBox` object receives an on-click event. To define the click event handler for a checkbox, add the `android:onClick` attribute to the `CheckBox` element in your XML layout. The value for this attribute must be the name of the method you want to call in response to a click event. The activity hosting the layout must then implement the corresponding method. However, this example uses another approach.

MyApplication/app/src/main/AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android = "http://schemas.android.com/apk/res/android"
  package       = "com.example.wenchen.myapplication" >
  <application
```

```
    android:icon  = "@mipmap/ic_launcher"
    android:label = "Checkboxes"
    android:theme = "@style/AppTheme" >
    <activity
      android:name  = ".MainActivity"
      android:label = "@string/app_name" >
      <intent-filter>
        <action   android:name = "android.intent.action.MAIN" />
        <category android:name = "android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

Slide 9.9: Radio buttons: Java source code
Slide 9.11: Checkboxes: layout file
Home

PRINT VERSION