

Assignment No: 01

Install Android Studio and Run Hello World App.

Task 1: Install Android Studio

Android Studio provides a complete integrated development environment (IDE) including an advanced code editor and a set of app templates. In addition, it contains tools for development, debugging, testing, and performance that make it faster and easier to develop apps. You can test your apps with a large range of preconfigured emulators or on your own mobile device, build production apps, and publish on the Google Play store.

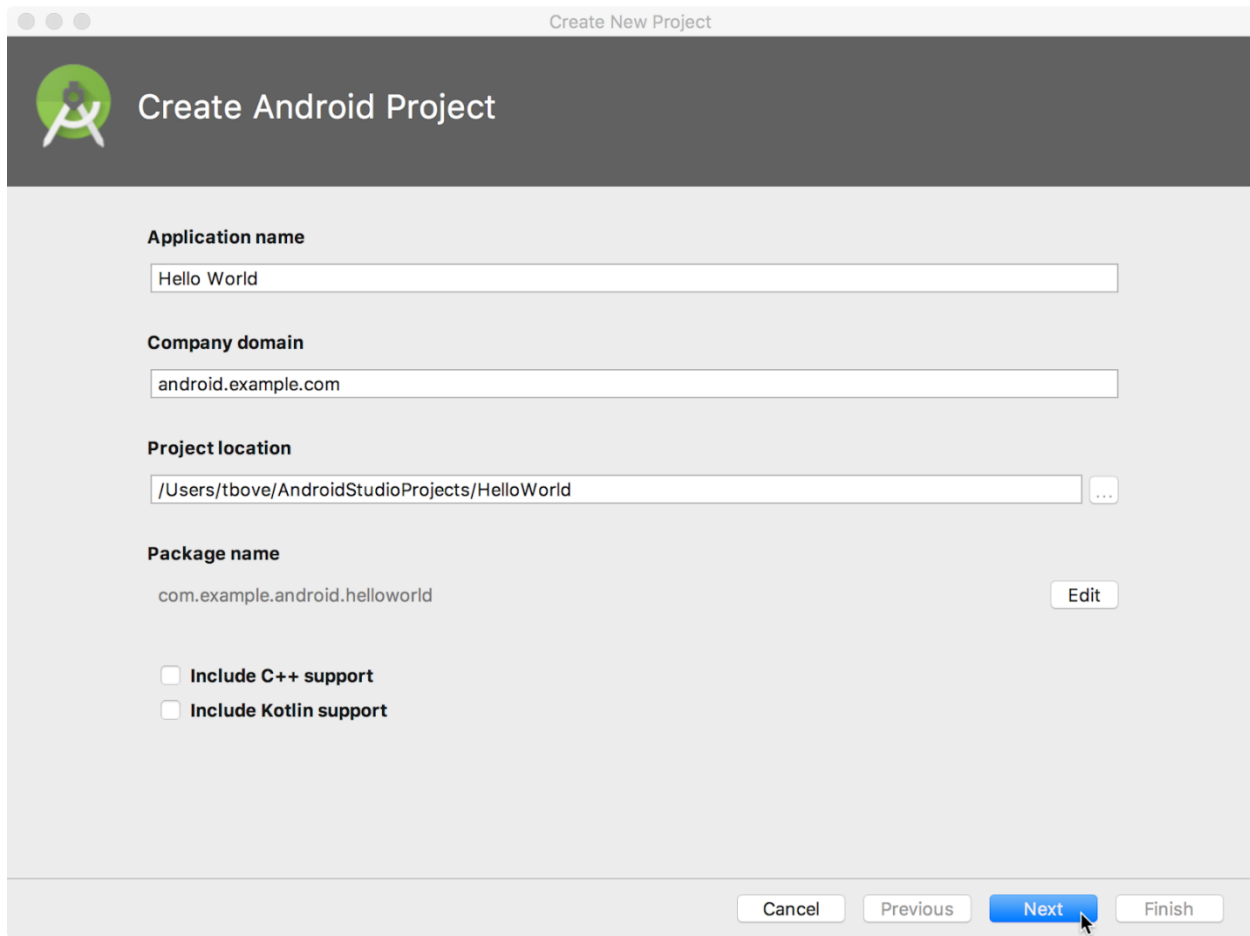
1. Navigate to the Android developers site and follow the instructions to download and install Android Studio.
2. Accept the default configurations for all steps, and ensure that all components are selected for installation.
3. After finishing the install, the Setup Wizard will download and install some additional components including the Android SDK. Be patient, this might take some time depending on your Internet speed, and some of the steps may seem redundant.
4. When the download completes, Android Studio will start, and you are ready to create your first project.

Task 2: Create the Hello World app

In this task, you will create an app that displays "Hello World" to verify that Android studio is correctly installed, and to learn the basics of developing with Android Studio.

2.1 Create the app project

1. Open Android Studio if it is not already opened.
2. In the main **Welcome to Android Studio** window, click **Start a new Android Studio project**.
3. In the **Create Android Project** window, enter **Hello World** for the **Application name**.



Create New Project

Create Android Project

Application name
Hello World

Company domain
android.example.com

Project location
/Users/tbove/AndroidStudioProjects/HelloWorld

Package name
com.example.android.helloworld

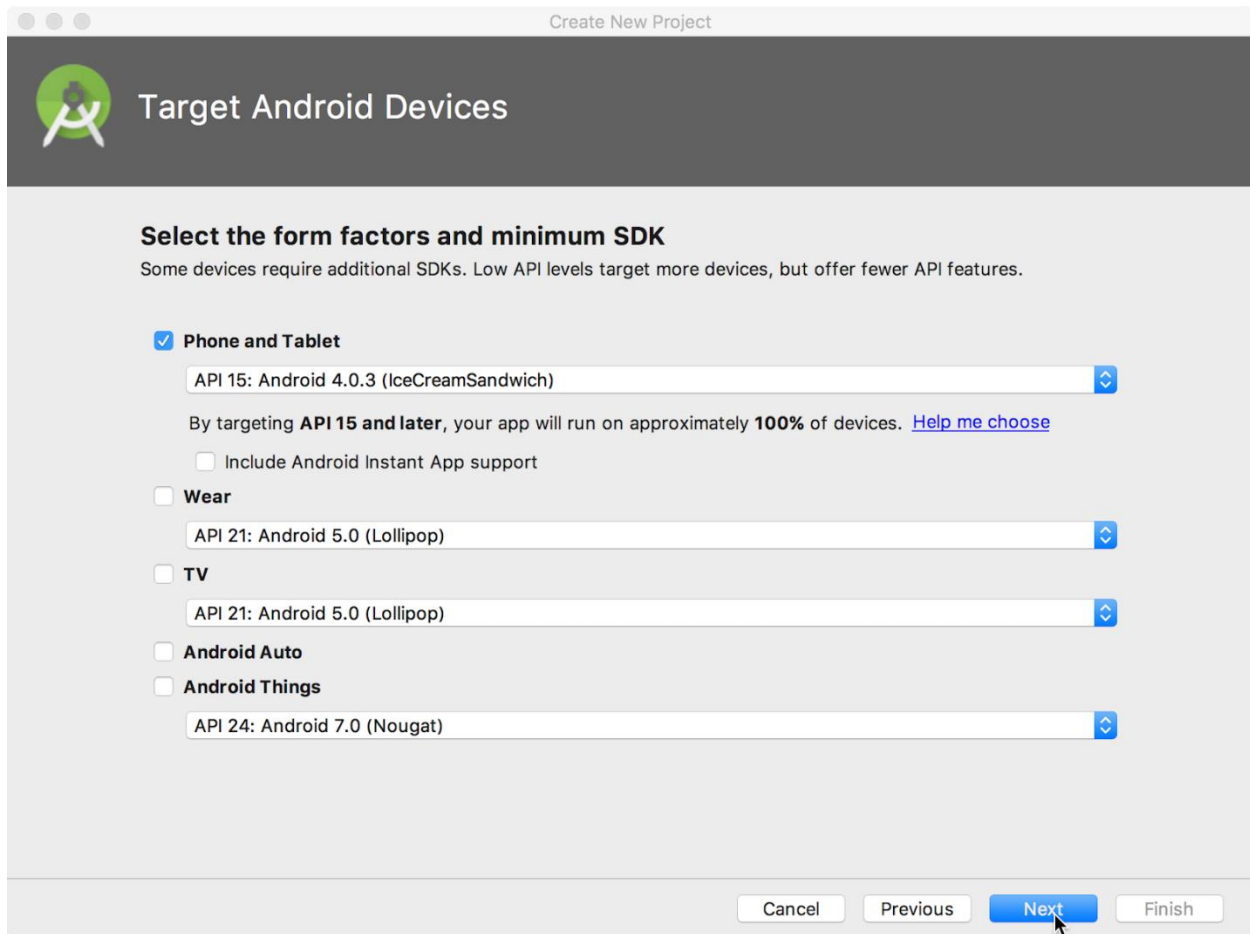
☐ Include C++ support
☐ Include Kotlin support

Cancel Previous Next Finish

4. Verify that the default **Project location** is where you want to store your Hello World app and other Android Studio projects, or change it to your preferred directory.
5. Accept the default **android.example.com** for **Company Domain**, or create a unique company domain.

If you are not planning to publish your app, you can accept the default. Be aware that changing the package name of your app later is extra work.


6. Leave unchecked the options to **Include C++ support** and **Include Kotlin support**, and click **Next**.
7. On the **Target Android Devices** screen, **Phone and Tablet** should be selected. Ensure that **API 15: Android 4.0.3 IceCreamSandwich** is set as the Minimum SDK; if it is not, use the popup menu to set it.





Create New Project

Target Android Devices


Select the form factors and minimum SDK
Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ **Phone and Tablet**
API 15: Android 4.0.3 (IceCreamSandwich) 
By targeting **API 15 and later**, your app will run on approximately **100%** of devices. [Help me choose](#)
☐ Include Android Instant App support

☐ **Wear**
API 21: Android 5.0 (Lollipop) 

☐ **TV**
API 21: Android 5.0 (Lollipop) 

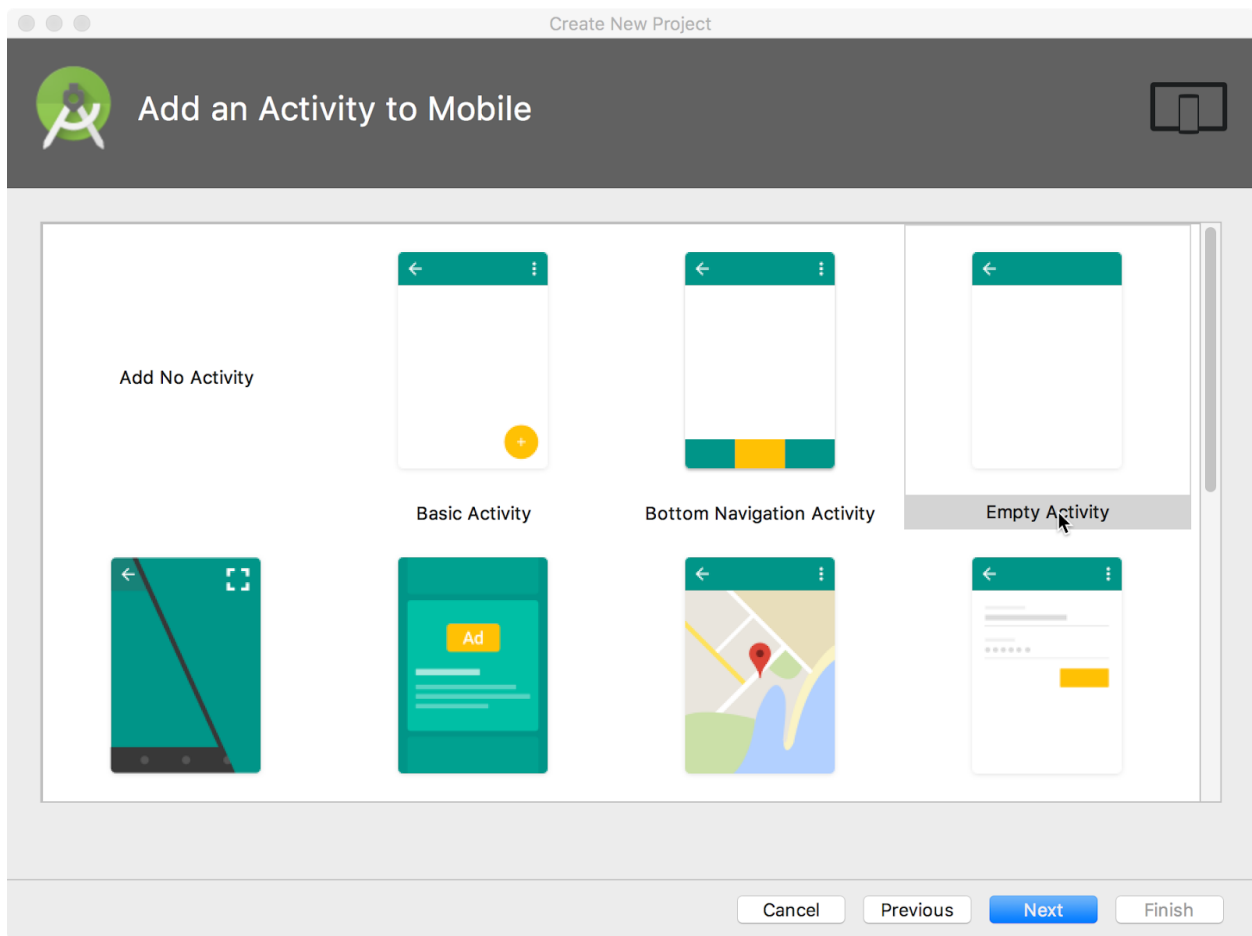
☐ **Android Auto**

☐ **Android Things**
API 24: Android 7.0 (Nougat) 

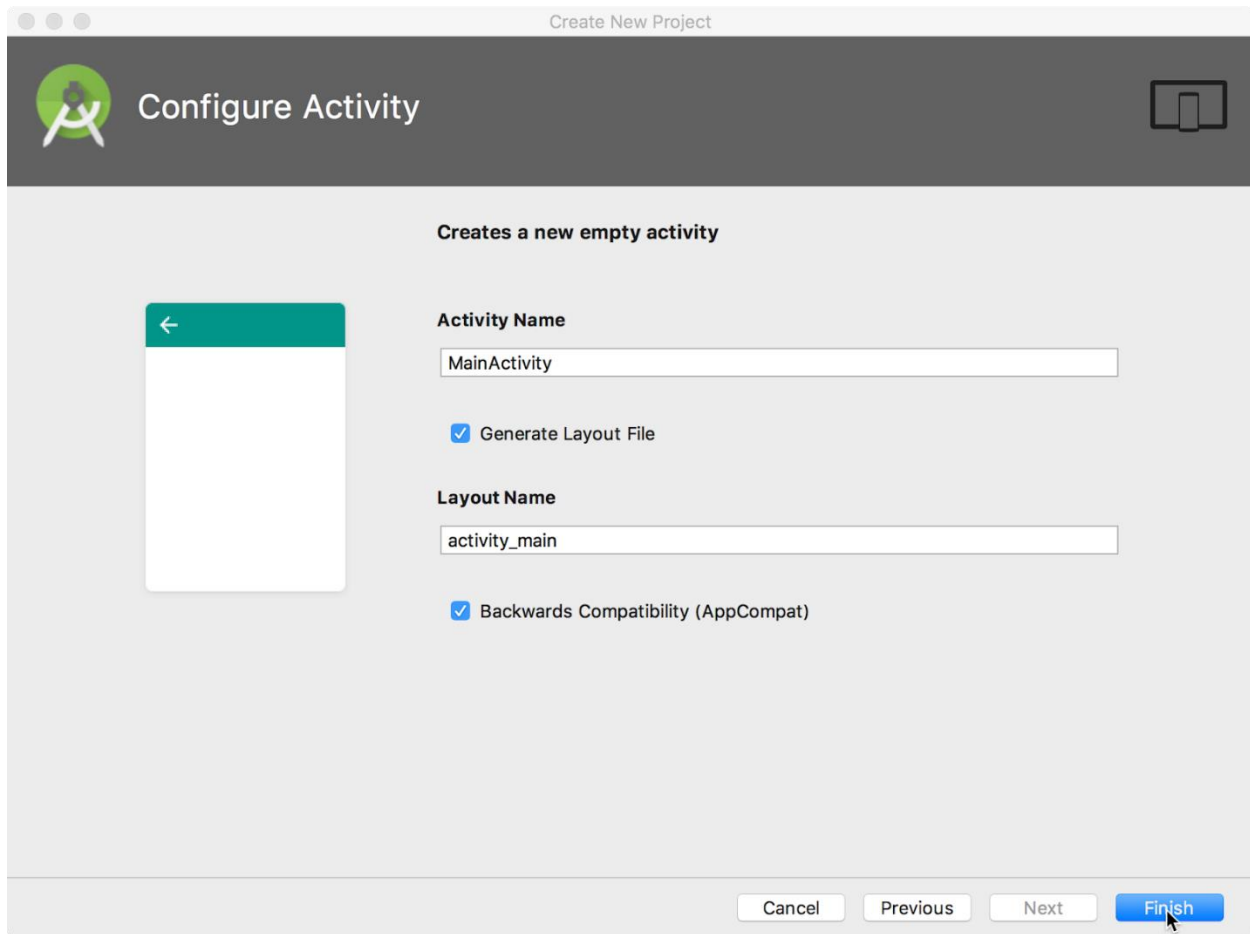
Cancel Previous **Next** Finish

These are the settings used by the examples in the lessons for this course. As of this writing, these settings make your Hello World app compatible with 97% of Android devices active on the Google Play Store.

- Leave unchecked the **Include Instant App support** and all other options. Then click **Next**. If your project requires additional components for your chosen target SDK, Android Studio will install them automatically.
- The **Add an Activity** window appears. An **Activity** is a single, focused thing that the user can do. It is a crucial component of any Android app. An Activity typically has a layout associated with it that defines how UI elements appear on a screen. Android Studio provides Activity templates to help you get started. For the Hello World project, choose **Empty Activity** as shown below, and click **Next**.



10. The **Configure Activity** screen appears (which differs depending on which template you chose in the previous step). By default, the empty Activity provided by the template is named MainActivity. You can change this if you want, but this lesson uses MainActivity.



11. Make sure that the **Generate Layout file** option is checked. The layout name by default is `activity_main`. You can change this if you want, but this lesson uses `activity_main`.
12. Make sure that the **Backwards Compatibility (App Compat)** option is checked. This ensures that your app will be backwards-compatible with previous versions of Android.
13. Click **Finish**.

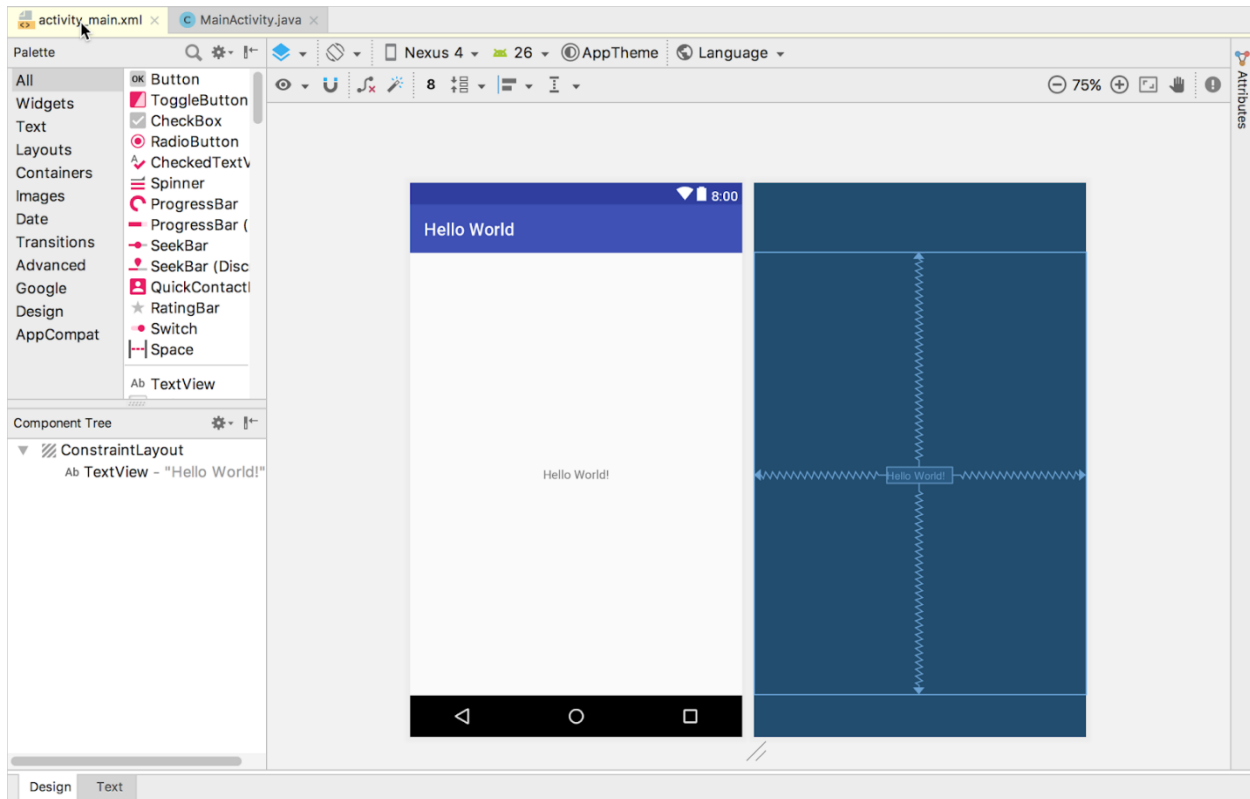
Android Studio creates a folder for your projects, and builds the project with Gradle (this may take a few moments).

Tip: See the [Configure your build](#) developer page for detailed information.

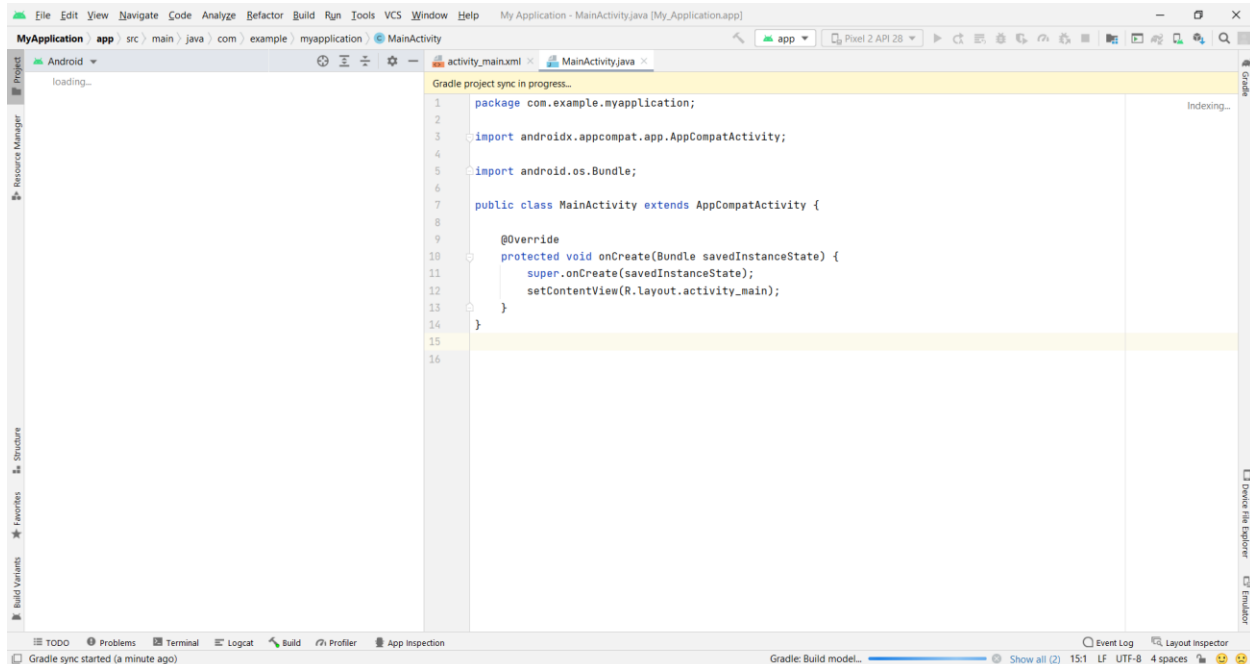
You may also see a "Tip of the day" message with keyboard shortcuts and other useful tips. Click **Close** to close the message.

The Android Studio editor appears. Follow these steps:

1. Click the **activity_main.xml** tab to see the layout editor.
2. Click the layout editor **Design** tab, if not already selected, to show a graphical rendition of the layout as shown below.



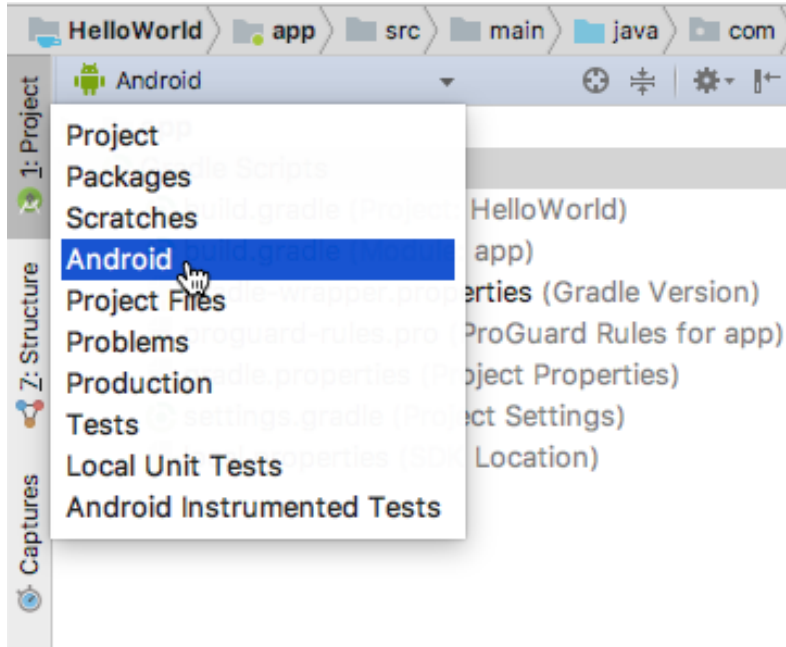
3. Click the **MainActivity.java** tab to see the code editor as shown below.



2.2 Explore the Project > Android pane

In this practical, you will explore how the project is organized in Android Studio.

1. If not already selected, click the **Project** tab in the vertical tab column on the left side of the Android Studio window. The Project pane appears.
2. To view the project in the standard Android project hierarchy, choose **Android** from the popup menu at the top of the Project pane, as shown below.



3. Click the **Activity_Main.xml** tab to see the code editor as shown below.

