# [**Android fundamentals 01.1: Android Studio and Hello World**](https://developer.android.com/codelabs/android-training-hello-world?index=..%2F..%2Fandroid-training)

**Question 1**

What is the name of the layout file for the main activity?

* MainActivity.java
* AndroidManifest.xml
* **activity\_main.xml**
* build.gradle

**Question 2**

What is the name of the string resource that specifies the application's name?

* app\_name
* xmlns:app
* **android:name**
* applicationId

**Question 3**

Which tool do you use to create a new emulator?

* Android Device Monitor
* **Device Manager**
* SDK Manager
* Theme Editor

**Question 4**

Assume that your app includes this logging statement:

Log.i("MainActivity", "MainActivity layout is complete");

You see the statement "MainActivity layout is complete" in the **Logcat** pane if the Log level menu is set to which of the following? (Hint: multiple answers are OK.)

* Verbose
* Debug
* **Info**
* Warn
* Error
* Assert

# [**Android fundamentals 01.2 Part B: The layout editor**](https://developer.android.com/codelabs/android-training-layout-editor-part-b?index=..%2F..%2Fandroid-training)

**Question 1**

Which two layout constraint attributes on the **Zero**Button position it vertically equal distance between the other two Button elements? (Pick 2 answers.)

* app:layout\_constraintBottom\_toTopOf="@+id/button\_count"
* **android:layout\_marginBottom="8dp"**
* android:layout\_marginStart="16dp"
* app:layout\_constraintTop\_toBottomOf="@+id/button\_toast"
* **android:layout\_marginTop="8dp"**

**Question 2**

Which layout constraint attribute on the **Zero**Button positions it horizontally in alignment with the other two Button elements?

* **app:layout\_constraintLeft\_toLeftOf="parent**"
* app:layout\_constraintBottom\_toTopOf="@+id/button\_count"
* android:layout\_marginBottom="8dp"
* app:layout\_constraintTop\_toBottomOf="@+id/button\_toast"

**Question 3**

What is the correct signature for a method used with the android:onClick XML attribute?

* public void callMethod()
* **public void callMethod(View view)**
* private void callMethod(View view)
* public boolean callMethod(View view)

**Question 4**

The click handler for the **Count**Button starts with the following method signature:

public void countUp(View view)

Which of the following techniques is more efficient to use within this handler to change the Button element's background color? Choose one:

* Use findViewById to find the **Count**Button. Assign the result to a View variable, and then use [setBackgroundColor()](https://developer.android.com/reference/android/view/View.html#setBackgroundColor(int)).
* **Use the view parameter that is passed to the click handler with**[**setBackgroundColor()**](https://developer.android.com/reference/android/view/View.html#setBackgroundColor(int))**: view.setBackgroundColor()**

# [**Android fundamentals 01.3: Text and scrolling views**](https://developer.android.com/codelabs/android-training-text-and-scrolling-views?index=..%2F..%2Fandroid-training)

**Question 1**

How many views can you use within a ScrollView? Choose one:

* One view only
* One view or one view group
* **As many as you need**

**Question 2**

Which XML attribute do you use in a LinearLayout to show views side by side? Choose one:

* **android:orientation="horizontal"**
* android:orientation="vertical"
* android:layout\_width="wrap\_content"

**Question 3**

Which XML attribute do you use to define the width of the LinearLayout inside the scrolling view? Choose one:

* android:layout\_width="wrap\_content"
* **android:layout\_width="match\_parent"**
* android:layout\_width="200dp"

# [**Android fundamentals 01.4: Learn to help yourself**](https://developer.android.com/codelabs/android-training-available-resources?index=..%2F..%2Fandroid-training)

**Question 1**

Within an Android Studio project, what menu command can you use to open the list of sample apps? Choose one:

* File > Open
* **File > New > Import Sample**
* File > New > Import Module
* File > New > Import Project

**Question 2**

Which buttons does the Basic Activity template provide as part of the UI? Choose two:

* Navigation buttons
* Options menu overflow button
* Floating action button
* **Button class button with the text "Button"**

**Question 3**

Which source of documentation is the official documentation for Android developers? Choose one:

* stackoverflow.com
* officialandroid.blogspot.com
* **developer.android.com**
* github.com

# [**Android fundamentals 02.1: Activities and intents**](https://developer.android.com/codelabs/android-training-create-an-activity?index=..%2F..%2Fandroid-training)

**Question 1**

What changes are made when you add a second Activity to your app by choosing **File > New > Activity** and an Activity template? Choose one:

* The second Activity is added as a Java class. You still need to add the XML layout file.
* The second Activity XML layout file is created and a Java class added. You still need to define the class signature.
* **The second Activity is added as a Java class, the XML layout file is created, and the AndroidManifest.xml file is changed to declare a second Activity.**
* The second Activity XML layout file is created, and the AndroidManifest.xml file is changed to declare a second Activity.

**Question 2**

What happens if you remove the android:parentActivityName and the <meta-data> elements from the second Activity declaration in the AndroidManifest.xml file? Choose one:

* The second Activity no longer appears when you try to start it with an explicit Intent.
* The second Activity XML layout file is deleted.
* **The Back button no longer works in the second Activity to send the user back to the main Activity.**
* The Up button in the app bar no longer appears in the second Activity to send the user back to the parent Activity.

**Question 3**

Which constructor method do you use to create a new explicit Intent? Choose one:

* **new Intent()**
* new Intent(Context context, Class<?> class)
* new Intent(String action, Uri uri)
* new Intent(String action)

**Question 4**

In the HelloToast app homework, how do you add the current value of the count to the Intent? Choose one:

* As the Intent data
* As the Intent TEXT\_REQUEST
* As an Intent action
* **As an Intent extra**

**Question 5**

In the HelloToast app homework, how do you display the current count in the second "Hello" Activity? Choose one:

* Get the Intent that the Activity was launched with.
* Get the current count value out of the Intent.
* Update the TextView for the count.
* **All of the above.**

# [**Android fundamentals 02.2: Activity lifecycle and state**](https://developer.android.com/codelabs/android-training-activity-lifecycle-and-state?index=..%2F..%2Fandroid-training)

**Question 1**

If you run the homework app before implementing onSaveInstanceState(), what happens if you rotate the device? Choose one:

* The EditText no longer contains the text you entered, but the counter is preserved.
* **The counter is reset to 0, and the EditText no longer contains the text you entered.**
* The counter is reset to 0, but the contents of the EditText is preserved.
* The counter and the contents of the EditText are preserved.

**Question 2**

What Activity lifecycle methods are called when a device-configuration change (such as rotation) occurs? Choose one:

* Android immediately shuts down your Activity by calling onStop(). Your code must restart the Activity.
* Android shuts down your Activity by calling onPause(), onStop(), and onDestroy(). Your code must restart the Activity.
* **Android shuts down your Activity by calling onPause(), onStop(), and onDestroy(), and then starts it over again, calling onCreate(), onStart(), and onResume().**
* Android immediately calls onResume().

**Question 3**

When in the Activity lifecycle is onSaveInstanceState() called? Choose one:

* **onSaveInstanceState() is called before the onStop() method.**
* onSaveInstanceState() is called before the onResume() method.
* onSaveInstanceState() is called before the onCreate() method.
* onSaveInstanceState() is called before the onDestroy() method.

**Question 4**

Which Activity lifecycle methods are best to use for saving data before the Activity is finished or destroyed? Choose one:

* **onPause() or onStop()**
* onResume() or onCreate()
* onDestroy()
* onStart() or onRestart()

# [**Android fundamentals 03.1: The debugger**](https://developer.android.com/codelabs/android-training-using-debugger?index=..%2F..%2Fandroid-training)

**Question 1**

Run the SimpleCalc app without the debugger. Leave one or both of the EditText elements empty, and try any calculation. Why did the error occur?

* **java.lang.NumberFormatException: empty String**
* W/OpenGLRenderer: Failed to choose config with EGL\_SWAP\_BEHAVIOR\_PRESERVED
* The app may be doing too much work on its main thread.
* The code cache capacity was increased to 128KB.

**Question 2**

Which function do you perform in the Debug pane in order to execute the current line where the breakpoint is, and then stop at the next line in the code? Choose one:

* **Step Into**
* Step Over
* Step Out
* Resume

**Question 3**

Which function do you perform in the Debug pane in order to jump to the execution of a method call from the current line where the breakpoint is? Choose one:

* Step Into
* **Step Over**
* Step Out
* Resume

# [**Android fundamentals 03.2: Unit tests**](https://developer.android.com/codelabs/android-training-unit-tests?index=..%2F..%2Fandroid-training)

**Question 1**

Which statement best describes a local unit test? Choose one:

* Tests that run on an Android-powered device or emulator and have access to the Android framework.
* Tests that enable you to write automated UI test methods.
* **Tests that are compiled and run entirely on your local machine with the Java Virtual Machine (JVM).**

**Question 2**

Source sets are collections of related code. In which source set are you likely to find unit tests? Choose one:

* app/res
* com.example.android.SimpleCalcTest
* **com.example.android.SimpleCalcTest (test)**
* com.example.android.SimpleCalcTest (androidTest)

**Question 3**

Which annotation is used to mark a method as an actual test? Choose one:

* @RunWith(JUnit4.class)
* @SmallTest
* @Before
* **@Test**

# [**Android fundamentals 03.3: Support libraries**](https://developer.android.com/codelabs/android-training-support-libraries?index=..%2F..%2Fandroid-training)

**Question 1**

Which class appears when you *first***Step Into** the ContextCompat.getColor() method? Choose one:

* **MainActivity**
* ContextCompat
* AppCompatActivity
* Context

**Question 2**

In the class that appears, which statement is executed if the build version is API version 23 or newer? Choose one:

* return context.getColor(id);
* **return context.getResources().getColor(id);**
* throw new IllegalArgumentException("permission is null");
* return mResources == null ? super.getResources() : mResources;

**Question 3**

If you change the ContextCompat.getColor() method back to the getColor() method, what will happen when you run the app? Choose one:

* If your minSdkVersion is 15, the word getColor is underlined in red in the code editor. Hover your pointer over it, and Android Studio reports "Call requires API 23 (current min is 15)".
* The app will run without error on emulators and devices using API 23 or newer.
* The app will crash when the user taps **Change Color** if the emulator or device is using API 17.
* **All of the above.**

# [**Android fundamentals 04.1: Clickable images**](https://developer.android.com/codelabs/android-training-clickable-images?index=..%2F..%2Fandroid-training)

Question 1

How do you add images to an Android Studio project? Choose one:

* Drag each image to the layout editor.
* Copy the image files into your project's drawable folder.
* Drag an ImageButton to the layout editor.
* **Choose New > Image Asset and then choose the image file.**

Question 2

How do you make an ImageView clickable like a simple Button? Choose one:

* Add the android:contentDescription attribute to the ImageView in the layout and use it to call the click handler in the Activity.
* Add the android:src attribute to the ImageView in the layout and use it to call the click handler in the Activity.
* **Add the android:onClick attribute to the ImageView in the layout and use it to call the click handler in the Activity.**
* Add the android:id attribute to the ImageView in the layout and use it to call the click handler in the Activity.

Question 3

Which rule applies to a click handler called from the attribute in the layout? Choose one:

* The click handler method must include the event listener View.OnClickListener, which is an interface in the View class .
* The click handler method must be public, return void, and define a View as its only parameter.
* **The click handler must customize the View.OnClickListener class and override its click handler to perform some action.**
* The click handler method must be private and return a View.

# [**Android fundamentals 04.2: Input controls**](https://developer.android.com/codelabs/android-training-input-controls?index=..%2F..%2Fandroid-training)

Question 1

What's the most important difference between checkboxes and a RadioGroup of radio buttons? Choose one:

* The only difference is in how they appear: checkboxes show a checkmark when selected, while circular "radio" buttons appear filled when selected.
* CheckBox elements in the layout can use the android:onClick attribute to call a handler when selected.
* **The major difference is that checkboxes enable multiple selections, while a RadioGroup allows only one selection.**

Question 2

Which layout group lets you align a set of CheckBox elements vertically? Choose one:

* RelativeLayout
* **LinearLayout**
* ScrollView

Question 3

Which of the following is the method of the [Checkable](https://developer.android.com/reference/android/widget/Checkable.html) interface to check the state of a radio button (that is, whether it has been selected or not)?

* getId()
* **isChecked()**
* onRadioButtonClicked()
* onClick()

# [**Android fundamentals 04.3: Menus and pickers**](https://developer.android.com/codelabs/android-training-menus-and-pickers?index=..%2F..%2Fandroid-training)

Question 1

What is the name of the file in which you create options menu items? Choose one:

* menu.java
* **menu\_main.xml**
* activity\_main.xml
* content\_main.xml

Question 2

Which method is called when an options menu item is clicked? Choose one:

* onOptionsItemSelected(MenuItem item)
* **onClick(View view)**
* onContextItemSelected()
* onClickShowAlert()

Question 3

Which of the following statements sets the title for an alert dialog? Choose one:

* myAlertBuilder.setMessage("Alert");
* myAlertBuilder.setPositiveButton("Alert");
* **myAlertBuilder.setTitle("Alert");**
* AlertDialog.Builder myAlertBuilder = new AlertDialog.Builder("Alert");

Question 4

Where do you create a DialogFragment for a date picker? Choose one:

* In the onCreate() method in the hosting Activity.
* In the onCreateContextMenu() method in Fragment.
* In the onCreateView() method in the extension of DialogFragment.
* **In the onCreateDialog() method in the extension of DialogFragment.**

# [**Android fundamentals 04.4: User navigation**](https://developer.android.com/codelabs/android-training-provide-user-navigation?index=..%2F..%2Fandroid-training)

### ****Question 1****

Which template provides an Activity with an options menu and the [v7 appcompat](https://developer.android.com/tools/support-library/features.html#v7-appcompat) support library [Toolbar](https://developer.android.com/reference/android/support/v7/widget/Toolbar.html) as the app bar? Choose one:

* Empty Activity template
* Basic Activity template
* **Navigation Drawer Activity template**
* Bottom Navigation Activity

### ****Question 2****

Which dependency do you need in order to use a [TabLayout](https://developer.android.com/reference/android/support/design/widget/TabLayout.html)? Choose one:

* **com.android.support:design**
* com.android.support.constraint:constraint-layout
* junit:junit:4.12
* com.android.support.test:runner

### ****Question 3****

Where do you define each child Activity and parent Activity to provide **Up** navigation? Choose one:

* To provide the **Up** button for a child screen Activity, declare the parent Activity in the child Activity section of the activity\_main.xml file.
* To provide the **Up** button for a child screen Activity, declare the parent Activity in the "main" XML layout file for the child screen Activity.
* **To provide the Up button for a child screen Activity, declare the parent Activity in the child Activity section of the AndroidManifest.xml file.**
* To provide the **Up** button for a child screen Activity, declare the parent Activity in the parent Activity section of the AndroidManifest.xml file.

# **Android fundamentals 04.5: Recyclerview**

**Question 1**

Which of the following statements about a RecyclerView is *false*? Choose one.

* A RecyclerView is a more resource-efficient way to display scrollable lists.
* You need to provide a layout for just one item of the list.
* All list items look the same.
* **You don't need a layout manager with a RecyclerView to handle the hierarchy and layout of View elements.**

**Question 2**

Which of the following is the primary component you need to provide to an adapter a View item and its position within a RecyclerView? Choose one.

* RecyclerView
* **RecyclerView.Adapter**
* RecyclerView.ViewHolder
* AppCompatActivity

**Question 3**

Which interface do you need to implement in order to listen and respond to user clicks in a RecyclerView? Choose one.

* **View.onClickListener**
* RecyclerView.Adapter
* RecyclerView.ViewHolder
* View.OnKeyListener

# [**Android fundamentals 05.1: Drawables, styles, and themes**](https://developer.android.com/codelabs/android-training-drawables-styles-and-themes?index=..%2F..%2Fandroid-training)

**Question 1**

Which type of Drawable do you use to create a Button with a background that stretches properly to accommodate the text or image inside the Button so that it looks correct for different screen sizes and orientations? Choose one:

* LevelListDrawable
* TransitionDrawable
* StateListDrawable
* **NinePatchDrawable**

**Question 2**

Which type of Drawable do you use to create a Button that shows one background when it is pressed and a different background when it is hovered over? Choose one:

* LevelListDrawable
* TransitionDrawable
* **StateListDrawable**
* NinePatchDrawable

**Question 3**

Suppose you want to create an app that has a white background, dark text, and a dark action bar. Which base style does your application style inherit from? Choose one:

* Theme.AppCompat.Light
* Theme.AppCompat.Dark.NoActionBar
* **Theme.AppCompat.Light.DarkActionBar**
* Theme.AppCompat.NoActionBar
* Theme.NoActionBar

# [**Android fundamentals 05.2: Cards and colors**](https://developer.android.com/codelabs/android-training-cards-and-colors?index=..%2F..%2Fandroid-training)

### ****Question 1****

Which color attribute in your style defines the color of the status bar above the app bar? Choose one:

* colorPrimary
* **colorPrimaryDark**
* colorAccent
* colorAccentDark

### ****Question 2****

Which support library does the FloatingActionButton belong to? Choose one:

* v4 Support Library
* **v7 Support Library**
* Design Support Library
* Custom Button Support Library

### ****Question 3****

In the [Material Design color palette](https://material.google.com/style/color.html#color-color-palette), which shade of a color should you use as the primary color for your brand in your app? Choose one:

* Any color shade that starts with A.
* Any color shade labeled 200.
* **Any color shade labeled 500.**
* Any color shade labeled 900.

# [**Android fundamentals 05.3: Adaptive layouts**](https://developer.android.com/codelabs/android-training-adaptive-layouts?index=..%2F..%2Fandroid-training)

**Question 1**

Which resource qualifier is used most frequently to select for tablets? Choose one:

* Orientation
* Screen width
* Screen height
* **Smallest screen width**

**Question 2**

Which folder would hold the strings.xml file for translation into French for Canada? Choose one:

* res/values-fr-rFR/
* res/values-ca-rFR/
* **res/values-fr-rCA/**
* res/values-en-rFR/

**Question 3**

Which folder is for XML files that contain strings, integers, and colors? Choose one:

* res/layout
* res/mipmap
* res/raw
* **res/values**

# [**Android fundamentals 07.1: AsyncTask**](https://developer.android.com/codelabs/android-training-create-asynctask?index=..%2F..%2Fandroid-training)

**Question 1**

For a ProgressBar:

1. How do you determine the range of values that a ProgressBar can show?
2. How do you change how much of the progress bar is filled in?

**Question 2**

If an AsyncTask is defined as follows:

 private class DownloadFilesTask extends AsyncTask<URL, Integer, Long>

1. What is the type of the value that is passed to doInBackground() in the AsyncTask?

**-> URL**

1. What is the type of the value that is passed to the callback that reports the progress of the task?

**-> Integer**

1. What is the type of the value that is passed to the callback that is executed when the task completes?

-**> Long**

**Question 3**

To report progress of the work executed by an AsyncTask, what callback method do you *implement*, and what method do you *call* in your AsyncTask subclass?

* Implement publishProgress(). Call publishProgress().
* Implement publishProgress(). Call onProgressUpdate().
* **Implement onProgressUpdate(). Call publishProgress().**
* Implement onProgressUpdate(). Call onProgressUpdate().

# [**Android fundamentals 07.2: AsyncTask and AsyncTaskLoader**](https://developer.android.com/codelabs/android-training-asynctask-asynctaskloader?index=..%2F..%2Fandroid-training)

**Question 1**

What permissions does your app need to connect to the internet?

* android.permission.CONNECTIVITY
* **android.permission.INTERNET**
* It doesn't need any special permissions, because all Android apps are allowed to connect to the internet.

**Question 2**

How does your app check that internet connectivity is available?

In the manifest:

* **request ACCESS\_NETWORK\_STATE permission**
* request ALL\_NETWORK\_STATE permission
* request NETWORK\_CONNECT permission

In the code:

* Wrap the code to connect to the internet in a try/catch block, and catch NO\_NETWORK errors.
* **Use ConnectivityManager to check for an active network before connecting to the network.**
* Present a dialog to the user reminding them to make sure that internet connectivity is available before they attempt to connect to the internet.

**Question 3**

Where do you implement the loader callback method that's triggered when the loader finishes executing its task?

* **In the AsyncTaskLoader subclass. The AsyncTaskLoader must implement LoaderManager.LoaderCallbacks.**
* In the Activity that displays the results of the task. The Activity must implement LoaderManager.LoaderCallbacks.
* In a Utility class that extends Object and implements LoaderManager.LoaderCallbacks.

**Question 4**

When the user rotates the device, how do AsyncTask and AsyncTaskLoader behave differently if they are in the process of running a task in the background?

* A running AsyncTask becomes disconnected from the activity, but keeps running. A running AsyncTaskLoader becomes disconnected from the activity and stops running, preserving system resources.
* A running AsyncTask becomes disconnected from the activity and stops running, preserving system resources. A running AsyncTaskLoader automatically restarts execution of its task from the beginning. The activity displays the results.
* **A running AsyncTask becomes disconnected from the activity, but keeps running. A running AsyncTaskLoader automatically reconnects to the activity after the device rotation. The activity displays the results.**

**Question 5**

How do you initialize an AsyncTaskLoader to perform steps, such as initializing variables, that must be done before the loader starts performing its background task?

* **In onCreateLoader() in the activity, create an instance of the AsyncTaskLoader subclass. In the loader's constructor, perform initialization tasks.**
* In onCreateLoader() in the activity, create an instance of the AsyncTaskLoader subclass. In the loader's init() method, perform initialization tasks.
* In the Activity, implement initLoader() to initialize the loader.
* Perform initialization tasks for the loader at the start of loadInBackgroud() in the Loader.

**Question 6**

What methods must an AsyncTaskLoader implement?

**-> onCreateLoader()**

# [Android fundamentals 07.3: Broadcast receivers](https://developer.android.com/codelabs/android-training-broadcast-receivers?index=..%2F..%2Fandroid-training)

### **Question 1**

What is a system broadcast?

* A message that your app sends and receives when an event of interest occurs in the app.
* A message that is sent from an app to a different component of the same app.
* **A message that the Android system sends when a system event occurs.**
* A message that the Android system receives when an event of interest occurs in your app.

### **Question 2**

Which pair of methods do you use to register and unregister your broadcast receiver dynamically?

* registerBroadcast() and unRegisterBroadcast().
* registerComponentCallbacks() and unRegisterComponentCallbacks().
* registerBroadcastReceiver() and unRegisterBroadcastReceiver().
* **registerReceiver() and unRegisterReceiver().**

### **Question 3**

Which of the following are true?

* Broadcast receivers can't see or capture the intents used to start an activity.
* Using a broadcast intent, you can't find or start an activity.
* You can use a broadcast intent to start an activity.
* **You can receive the intent used to start activity in your broadcast receiver.**

### **Question 4**

Which class is used to mitigate the security risks of broadcast receivers when the broadcasts are not cross-application (that is, when broadcasts are sent and received by the same app)?

* SecureBroadcast
* **LocalBroadcastManager**
* OrderedBroadcast
* SecureBroadcastManager

# [Android fundamentals 08.1: Notifications](https://developer.android.com/codelabs/android-training-notifications?index=..%2F..%2Fandroid-training)

### **Question 1**

Select all that are true for notification channels:

* You use notification channels to display notifications to the user in the device status bar.
* **You use notification channels to group multiple notifications so that the user can control the notifications' behavior.**
* Notification channels are available in older devices, those running Android 7.0 Nougat (API 24) and lower.
* Notification channels are not yet available in the Android Support Library package.

### **Question 2**

Which API do you use to show a notification in the notification drawer and in the device's status bar?

* Notification.notify()
* **NotificationManager.notify()**
* NotificationCompact.notify()
* NotificationCompat.Builder.notify()

### **Question 3**

Which component is *not* needed when you add a notification action?

* Icon that represents the action
* Title that describes the action
* **Click listener for the action button click event.**
* PendingIntent that's sent when the user taps the action button.

### **Question 4**

Which API do you use to add an action button to a notification?

* NotificationCompat.addActionButton()
* **NotificationCompat.Builder.addAction()**
* Notification.Builder.addActionButton()
* NotificationManager.addAction()

# [Android fundamentals 08.2: The alarm manager](https://developer.android.com/codelabs/android-training-alarm-manager?index=..%2F..%2Fandroid-training)

### **Question 1**

In which API level did inexact timing become the default for AlarmManager? (All set() methods use inexact timing, unless explicitly stated.)

* API level 16
* API level 18
* **API level 19**
* API level 17

# [Android fundamentals 08.3: JobScheduler](https://developer.android.com/codelabs/android-training-job-scheduler?index=..%2F..%2Fandroid-training)

### **Question 1**

What class do you use if you want features like the ones provided by JobScheduler, but you want the features to work for devices running API level 20 and lower?

* **JobSchedulerCompat**
* workManager
* AlarmManager

# [Android fundamentals 09.1: Shared preferences](https://developer.android.com/codelabs/android-training-shared-preferences?index=..%2F..%2Fandroid-training)

### **Question 1**

In which lifecycle method do you save the app state to shared preferences?

**-> OnPause()**

### **Question 2**

In which lifecycle method do you restore the app state?

**-> OnCreate()**

### **Question 3**

Can you think of a case where it makes sense to have both shared preferences and instance state?

# [Android fundamentals 09.2: App settings](https://developer.android.com/codelabs/android-training-adding-settings-to-app?index=..%2F..%2Fandroid-training)

### **Question 1**

In which file of the DroidCafeWithSettings project do you define the array of entries and the array of values for the ListPreference? Choose one:

* pref\_general.xml
* **strings.xml**
* menu\_main.xml
* activity\_main.xml
* content\_main.xml

### **Question 2**

In which file of the DroidCafeWithSettings project do you *use* the array of entries and the array of values in setting up the ListPreference, and also set the ListPreference key and default value? Choose one:

* **pref\_general.xml**
* strings.xml
* menu\_main.xml
* content\_main.xml
* SettingsActivity.java

### **Question 3**

How do you set up a settings Activity and a Fragment with a SwitchPreference for the UI, and still remain compatible with the [v7 appcompat library](https://developer.android.com/topic/libraries/support-library/features.html#v7-appcompat) for backward compatibility with older versions of Android?

* Use a settings activity that extends Activity, a fragment that extends PreferenceFragment, and a SwitchPreference for the UI.
* Change MainActivity to extend Activity.
* **Use a settings activity that extends AppCompatActivity, a fragment that extends PreferenceFragmentCompat, and a SwitchPreferenceCompat for the UI.**
* You can't use a fragment with a SwitchPreference and remain compatible with the [v7 appcompat library](https://developer.android.com/topic/libraries/support-library/features.html#v7-appcompat).

# [Android fundamentals 10.1 Part A: Room, LiveData, and ViewModel](https://developer.android.com/codelabs/android-training-livedata-viewmodel?index=..%2F..%2Fandroid-training)

### **Question 1**

What are the advantages of using a Room database?

* **Creates and manages an Android SQLite database for you.**
* **Eliminates a lot of boilerplate code.**
* Helps you manage multiple backends.
* **Using a DAO, provides a mechanism for mapping Java methods to database queries.**

### **Question 2**

Which of the following are reasons for using a ViewModel?

* **Cleanly separates the UI from the backend.**
* **Often used with LiveData for changeable data that the UI will use or display.**
* Prevents your data from being lost when the app crashes.
* **Acts as a communication center between the Repository and the UI.**
* **ViewModel instances survive device configuration changes.**

### **Question 3**

What is the DAO?

* **Short for "data access object."**
* A library for managing database queries.
* **An annotated interface that maps Java methods to SQLite queries.**
* **A class whose methods run always in the background, not on the main thread.**
* **A class that the compiler checks for SQL errors, then uses to generate queries from the annotations.**

### **Question 4**

What are features of LiveData?

* **When LiveData is used with Room, data updates automatically if all the intermediate levels also return LiveData (DAO, ViewModel, Repository).**
* **Uses the observer pattern and notifies its observers when its data has changed.**
* **Automatically updates the UI when it changes.**
* Is lifecycle aware.

# [Android fundamentals 10.1 Part B: Deleting data from a Room database](https://developer.android.com/codelabs/android-training-room-delete-data?index=..%2F..%2Fandroid-training)

### Question 1

Android Architecture Components provide some convenience annotations for DAOs. Which of the following are available? Select as many as apply.

* **@Dao**
* **@Insert**
* **@Delete**
* **@Update**
* **@Query**
* @Select

### Question 2

What are the benefits of using Architecture Components?

* **Architecture Components help you structure your app in a way that is robust and testable.**
* Architecture Components help you create better UIs.
* **Architecture Components provide a simple, flexible, and practical approach to structuring your app.**
* **If you use the provided libraries and architecture, your app is more maintainable with less boilerplate code.**