PassQueen
a.k.a dumpsqueen.com

Android Application Development v8

Android AND-801

Version Demo

Total Demo Questions: 6

Total Premium Questions: 62

Buy Premium PDF

https://passqueen.com

support@passqueen.com

QUESTION NO: 1

The following Android code locates Toronto CN tower on Google map using on the latitude and longitude coordinates. Which of the following choices is the correct choice to show your Google map in the zoom-in state?

```
override fun onMapReady(googleMap: GoogleMap) {
 GMap=googleMap
 val cnTower =LatLng(43.642633,-79.3878268)
 GMap.moveCamera(CameraUpdateFactory.newLatLng(cnTower))
 GMap.addMarker(MarkerOptions().position(cnTower).title("Toronto CN Tower"))

}
```

A. Replace `gMap.moveCamera(CameraUpdateFactory.newLatLng(cnTower));` with `gMap.moveCamera(CameraUpdateFactory.newLatLngZoom(cnTower, 16f));`

B. We can only show a position on Google map and we don't have permission to modify the zoom level

C. Change the coordinates value to get the zoom level what we need

D. Add the following code: GoogleMapZoom(60)%

ANSWER: A

Explanation:

Replace `gMap.moveCamera(CameraUpdateFactory.newLatLng(cnTower));` with `gMap.moveCamera(CameraUpdateFactory.newLatLngZoom(cnTower, 16f));` because `CameraUpdateFactory.newLatLngZoom()` creates a camera update that sets both the map target location and its zoom level. The `cnTower LatLng` value continues to identify the CN Tower's latitude and longitude, while `16f` supplies a floating-point zoom level appropriate for a close, zoomed-in view of a landmark. Applying that update through `moveCamera()` immediately positions the Google Map camera at the specified coordinates and zoom. Camera zoom is controlled independently from the latitude and longitude values, so it should be explicitly included in the camera update when the required initial view must be zoomed in. The Maps SDK for Android documents `newLatLngZoom(LatLng, float)` specifically for changing the camera to a geographic location at a specified zoom level. See the CameraUpdateFactory.newLatLngZoom reference and the Maps SDK for Android camera and map views guide.

QUESTION NO: 2

Android Studio is the official IDE for Android application development.

A. False

B. True

ANSWER: B

Explanation:

True is correct. Android Studio is Google's official integrated development environment for Android app development. It is built on JetBrains IntelliJ IDEA and provides the core tools needed to create, build, test, debug, profile, and package Android applications. Its Android-specific capabilities include project templates, a visual layout editor, Gradle-based build integration, Logcat, the Android Emulator and Device Manager, code inspection and refactoring tools, APK/App Bundle analysis, and integrations for testing and performance profiling.

Google develops, maintains, and distributes Android Studio as the primary supported desktop development environment for Android. It is available for Windows, macOS, Linux, and ChromeOS, and it is updated alongside Android platform tooling so

developers can work with current SDKs, build tools, emulators, and Android Gradle Plugin versions. While Android development can involve other editors or IDEs in some workflows, Android Studio is explicitly identified by the Android Developers documentation as the official IDE. See the Android Studio overview and the official introduction to Android Studio.

QUESTION NO: 3

Which of the below choices is the best answer to fill the empty space in the following sentence? If you want to give all text widgets in an app the same format such as font color, font size, and font family, you should add the style attribute to the TextView XML tags in layout file and set the attribute value to the same style name. You should define that style by adding a new style tag "style name-style_name" inside.....

A. AadroiddManifest-xml file

B. activity_main.xml file

C. stytes.xml file

D. MainActivity.java or MainActivity.kt file

ANSWER: C

Explanation:

The correct location is the `styles.xml` resource file, conventionally located at `res/values/styles.xml`. Android resource XML files in the `res/values/` directory are intended for reusable values such as colors, dimensions, strings, themes, and styles. A style is declared with a `<style>` element and a name, then contains one or more `<item>` elements that define view attributes. For example, a style can set `android:textColor`, `android:textSize`, and `android:fontFamily` once. Each `TextView` can then reference the shared definition with `style="@style/style_name"`, ensuring that text formatting remains consistent throughout the application and can be changed centrally later. Android's resource system compiles these named values and makes them available to layouts, themes, and code. Although modern Android projects may also organize styles in other appropriately named XML files under `res/values`, `styles.xml` is the standard answer for defining a reusable view style. See the Android documentation on styles and themes and app resources.

QUESTION NO: 4

The following image includes Android code for a WebView widget (ID: MyWebView) to work as a web browser for the URL: http://www.androidatc.com. When you run this code, the app does not open the www.androidatc.com web site. Which step is still missing to open this web site through this app?

```
class MainActivity : AppCompatActivity() {

 override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.activity_main)

 //Set the Web View to have a transparent border
 MyWebView.setBackgroundColor(Color.TRANSPARENT)

 //To enable JavaScript for the web browser
 MyWebView.settings.javaScriptEnabled =true

 //to load images automatically
 MyWebView.settings.loadsImagesAutomatically = true

 //Enable Scrolling
 MyWebView.scrollBarStyle=View.SCROLLBARS_INSIDE_OVERLAY

 //Shows the URL
 MyWebView.loadUrl("http://www.androidatc.com")
 }
}
```

A. Configure me activity main.xml file to enable Ihe WIFi connection or mobile data connection.

B. This app will open the web site without the need to add any additional configuration

C. Add additional code to configure FireFox web browser as default web browser for this app.

D. Configure the app's AndroidManifest file to allow this app to connect to the Internet by adding the tag.

ANSWER: D

Explanation:

Configure the app's AndroidManifest file to allow this app to connect to the Internet by adding the `<uses-permission android:name="android.permission.INTERNET" />` tag. A `WebView` renders web content within the application, but its call to `loadUrl()` still requires the app process to have permission to access the network. The Internet permission is a normal permission, so it is declared in the manifest and does not require a runtime permission request from the user. It should be placed as a direct child of the root `<manifest>` element, typically before the `<application>` element. For example: `<uses-permission android:name="android.permission.INTERNET" />`. Once declared, the WebView can make the network request required to load the site, assuming the device has a usable network connection. Android's documentation identifies `INTERNET` as the permission that allows applications to open network sockets, which is the capability needed for browser-style WebView navigation. See the Android INTERNET permission reference and the Android WebView documentation.

QUESTION NO: 5

Fill in the blank space in the following sentence with the correct choice: The..... to arranges widgets in positions relative each other.

A. RelativeLayout

B. Table Layout

C. Linear Layout

D. ConstraintLayout

ANSWER: A

Explanation:

RelativeLayout is correct because it is the Android view group specifically designed to position child views relative to one another or relative to the parent container. A widget can be placed below, above, to the start or end of another widget, aligned with another widget's edge, or centered within the parent. These relationships are declared through layout attributes on each child view, such as `layout_below`, `layout_toEndOf`, and `layout_alignParentTop`. This directly matches the sentence's description of arranging widgets in positions relative to each other.

For example, a label can be positioned above an input field, while a button can be placed below that input field, without requiring a strictly horizontal or vertical sequence. The Android API reference defines `RelativeLayout` as a layout where the positions of children can be described relative to each other or to the parent. See the official RelativeLayout API reference. Android's layout guide also explains how view groups define the positioning of their child views: Declare a layout.

QUESTION NO: 6

What does the following code snippet do?

```
package com.androidatc.lesson_07_android

class PlaceListAdapter(private val list:ArrayList<Place>,private val
context: Context):RecyclerView.Adapter<PlaceListAdapter.ViewHolder>(){

 override fun onBindViewHolder(holder: ?, position: Int) {}

 override fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): {}

 override fun getItemCount(): Int {}

 class ViewHolder(itemView:View): RecyclerView.ViewHolder(itemView) {
 }
}
```

A. Creates a FrameView.

B. Creates a Cradview.

C. Creates a RecyclerView.

D. Creates a ListView-

ANSWER: C

Explanation:

The code creates a RecyclerView, which is Android's flexible view container for displaying a collection of data items efficiently. A RecyclerView is typically declared or instantiated as a `RecyclerView`, then connected to a layout manager and an adapter. The layout manager determines how individual item views are arranged—for example, in a vertical list, horizontal list, or grid—while the adapter supplies the data and creates/binds the item views that appear on screen. This separation lets the component recycle item views that scroll off screen instead of repeatedly creating new views, which supports efficient rendering of potentially large or changing data sets.

In Android’s current guidance, RecyclerView is provided by the AndroidX RecyclerView library and is the standard component for presenting dynamic collections. Its architecture also supports optional item decoration, animations, and efficient updates when the underlying data changes. The Android Developers guide describes the required collaboration between a RecyclerView, its adapter, and its layout manager: Create dynamic lists with RecyclerView. The API reference further defines RecyclerView as a flexible container for a large data set that provides a limited window into that data: RecyclerView API reference.