

A SS OCIATE-ANDROID-DEVELOPER^{Q&As}

Google Developers Certification - Associate Android Developer (Kotlin
and Java Exam)

**Pass Google ASSOCIATE-ANDROID-DEVELOPER
Exam with 100% Guarantee**

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass2lead.com/associate-android-developer.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Google
Official Exam Center

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers



QUESTION 1

For example, we have a file in our raw folder `app/src/main/res/raw/sample_tears.json`. To get an `InputStream` for reading it, from our `Context` `context`, we can do this:

- A. `val input = context!!.openRawResource(R.raw.sample_tears)`
- B. `val input = context!!.getRawResource(R.raw.sample_tears)`
- C. `val input = context!!.resources.openRawResource(R.raw.sample_tears)`

Correct Answer: C

QUESTION 2

By default, the notification's text content is truncated to fit one line. If you want your notification to be longer, for example, to create a larger text area, you can do it in this way:

- A. `var builder = NotificationCompat.Builder(this, CHANNEL_ID).setContentText("Much longer text that cannot fit one line...").setStyle(NotificationCompat.BigTextStyle().bigText("Much longer text that cannot fit one line..."))`
...
- B. `var builder = NotificationCompat.Builder(this, CHANNEL_ID).setContentText("Much longer text that cannot fit one line...").setLongText("Much longer text that cannot fit one line...")`
...
- C. `var builder = NotificationCompat.Builder(this, CHANNEL_ID).setContentText("Much longer text that cannot fit one line...").setTheme(android.R.style.Theme_LongText); ...`

Correct Answer: A

Reference: <https://developer.android.com/training/notify-user/build-notification>

QUESTION 3

`SharedPreferences.Editor` is an interface used for modifying values in a `SharedPreferences` object. To mark in the editor that a preference value should be removed, which will be done in the actual preferences once `commit()` or `apply()` is called, what method in `SharedPreferences.Editor` should we use?

- A. `delete(String key)`
- B. `clear()`
- C. `remove(String key)`

D. removeAll()

Correct Answer: B

clear() method marks in the editor to remove ALL values from the preferences. Once commit is called, the only remaining preferences will be any that you have defined in this editor. And no delete and removeAll method exists in SharedPreferences.Editor

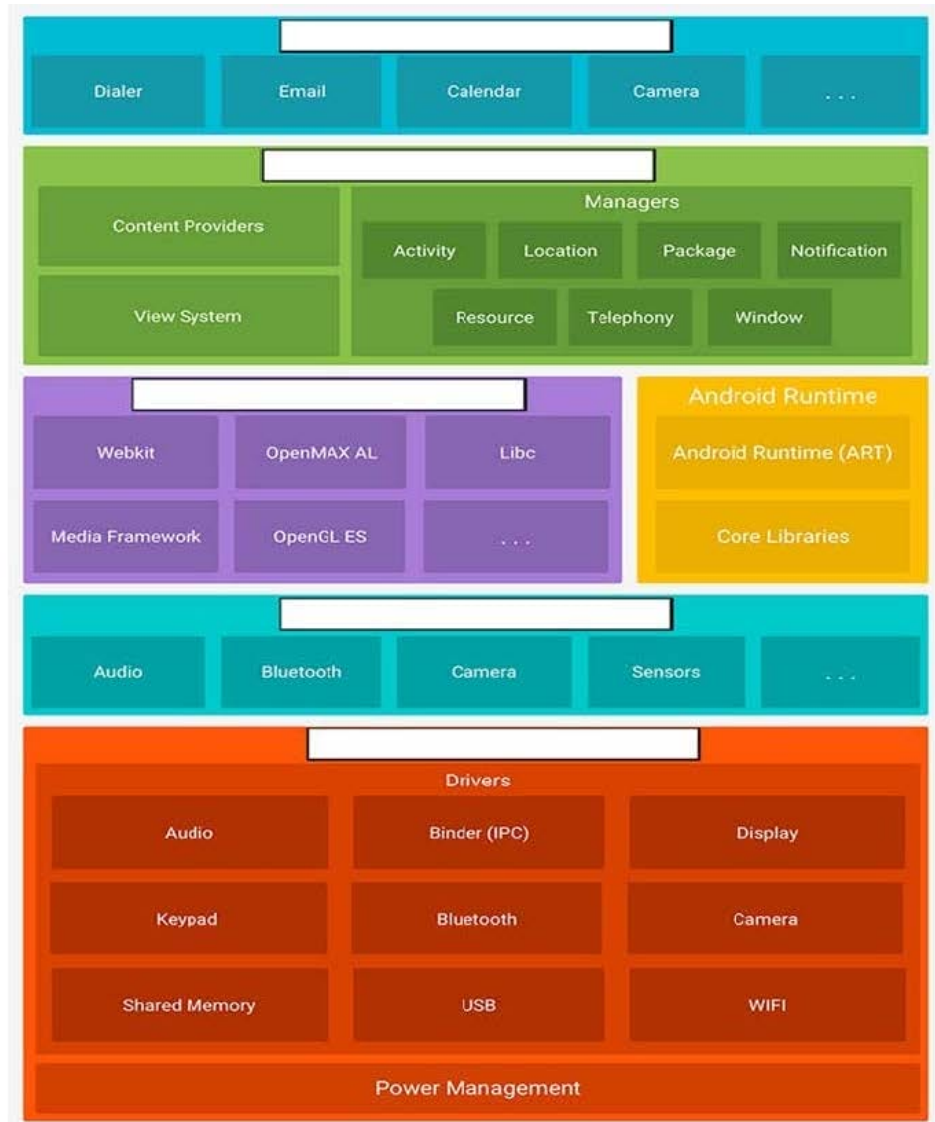
QUESTION 4

DRAG DROP

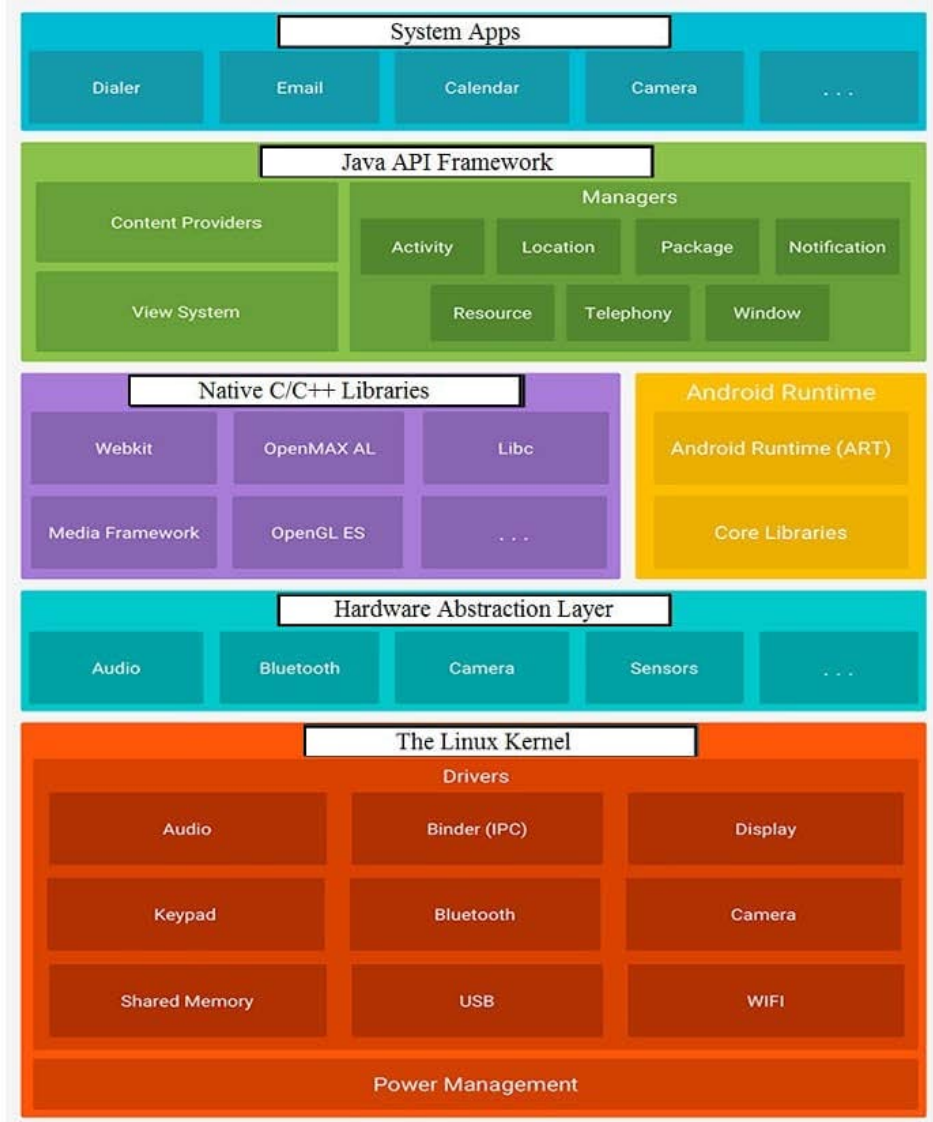
Move the major components of the Android platform to correct places in diagram.

Select and Place:

- System Apps
- Java API Framework
- Native C/C++ Libraries
- Hardware Abstraction Layer
- The Linux Kernel



Correct Answer:



Reference: <https://developer.android.com/guide/platform>

QUESTION 5

The easiest way of adding menu items (to specify the options menu for an activity) is inflating an XML file into the Menu via MenuInflater. With menu_main.xml we can do it in this way:

A. @Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return true;  
}
```

B. @Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return super.onOptionsItemSelected(item);  
}
```

C. @Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.menu.menu_main);  
}
```

Correct Answer: A

Reference:

<https://developer.android.com/guide/topics/ui/menus>

[Latest ASSOCIATE-
ANDROID-DEVELOPER
Dumps](#)

[ASSOCIATE-ANDROID-
DEVELOPER Study Guide](#)

[ASSOCIATE-ANDROID-
DEVELOPER Exam
Questions](#)