

# DUMPSBOSS.

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

Google Associate-Android-Developer

Version Demo

Total Demo Questions: 10

Total Premium Questions: 128

Buy Premium PDF

<https://dumpsboss.co>

[support@dumpsboss.co](mailto:support@dumpsboss.co)

support@dumpsboss.co  
dumpsboss.co

## Topic Break Down

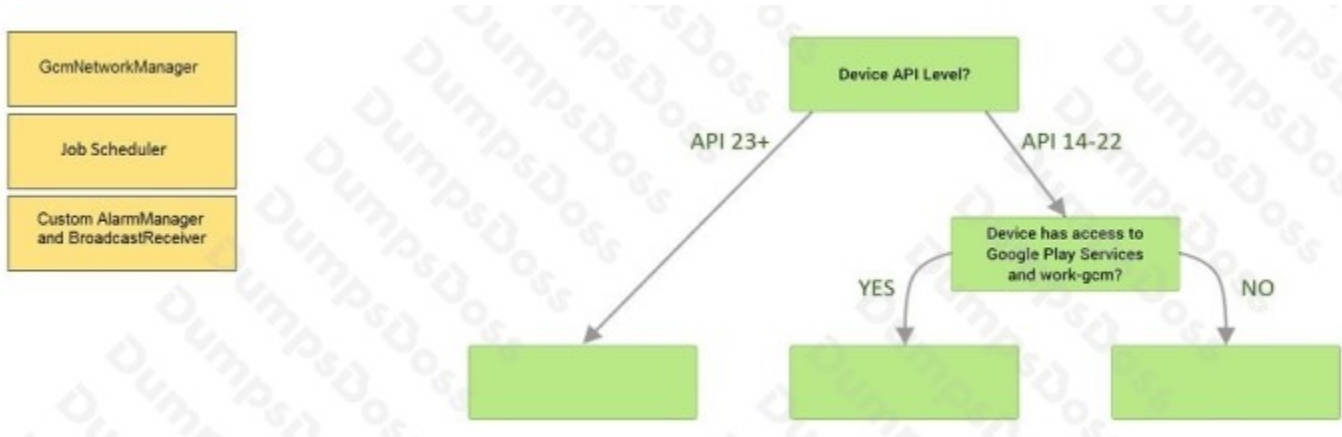
Topic	No. of Questions
Topic 1, KOTLIN only	64
Topic 2, JAVA only	64
<b>Total</b>	<b>128</b>

## QUESTION NO: 1 - (DRAG DROP)

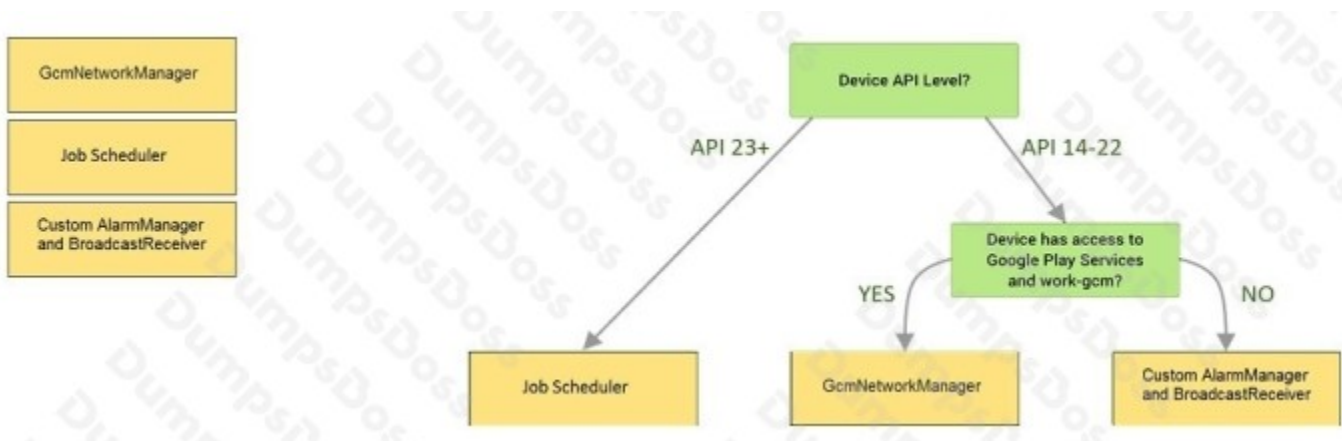
DRAG DROP

Under the hood WorkManager uses an underlying job dispatching service based on the following criteria. You need to move services to the correct places.

Select and Place:



ANSWER:



Explanation:

Videos:

- Working with WorkManager, from the 2018 Android Dev Summit
- WorkManager: Beyond the basics, from the 2019 Android Dev Summit

Reference: <https://developer.android.com/reference/androidx/work/WorkManager?hl=en>

## QUESTION NO: 2

If constant LENGTH\_INDEFINITE is used as a parameter for the setDuration method in Snackbar, what will happen?

- A. The Snackbar will be displayed for a short period of time.
- B. The Snackbar will be displayed for a long period of time.
- C. The Snackbar will be displayed for a very long period of time.
- D. The Snackbar will be displayed from the time that is shown until either it is dismissed, or another Snackbar is shown.
- E. The constant LENGTH\_INDEFINITE is impossible parameter for the setDuration method in Snackbar

**ANSWER: D**

### Explanation:

Reference:

[https://developer.android.com/reference/com/google/android/material/snackbar/BaseTransientBottomBar#LENGTH\\_INDEFINITE](https://developer.android.com/reference/com/google/android/material/snackbar/BaseTransientBottomBar#LENGTH_INDEFINITE) <https://developer.android.com/guide/topics/ui/notifiers/toasts> <https://developer.android.com/training/snackbar/action>

## QUESTION NO: 3

Relative positioning is one of the basic building blocks of creating layouts in ConstraintLayout. Constraints allow you to position a given widget relative to another one. What constraints do not exist?

- A. layout\_constraintBottom\_toBottomOf
- B. layout\_constraintBaseline\_toBaselineOf
- C. layout\_constraintBaseline\_toStartOf
- D. layout\_constraintStart\_toEndOf

**ANSWER: C**

### Explanation:

Reference:

<https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout>

## QUESTION NO: 4

The Layout Inspector in Android Studio allows you to compare your app layout with design mockups, display a magnified or 3D view of your app, and examine details of its layout at runtime. When this is especially useful?

- A. when your layout is built entirely in XML rather than runtime and the layout is behaving expectedly.

B. when your layout is built at runtime rather than entirely in XML and the layout is behaving unexpectedly.

**ANSWER: B**

## QUESTION NO: 5

If you want to access a specific UI component in an app, use the `UiSelector` class. This class represents a query for specific elements in the currently displayed UI. What is correct about it?

(Choose two.)

- A. If more than one matching element is found, the first matching element in the layout hierarchy is returned as the target `UiObject`.
- B. If no matching UI element is found, an `IOException` is thrown.
- C. If more than one matching element is found, the last matching element in the layout hierarchy is returned as the target `UiObject`.
- D. If no matching UI element is found, a `UiAutomatorObjectNotFoundException` is thrown.

**ANSWER: A D**

## QUESTION NO: 6

As an example. In an Activity we have our `TimerViewModel` object (extended `ViewModel`), named `mTimerViewModel`. `mTimerViewModel.getTimer()` method returns a `LiveData` value. What can be a correct way to set an observer to change UI in case if data was changed?

- A. 

```
mTimerViewModel.getTimer().getValue().toString().observe(new Observer() { @Override
public void onChanged(Long aLong) { callAnyChangeUIMethodHere(aLong)
}
});
```
- B. 

```
mTimerViewModel.getTimer().observe(this, new Observer() {
@Override
public void onChanged(Long aLong) { callAnyChangeUIMethodHere(aLong)
}
});
```
- C. 

```
mTimerViewModel.observe(new Observer() { @Override
public void onChanged(Long aLong) { callAnyChangeUIMethodHere(aLong)
}
});
```

**ANSWER: B**

## QUESTION NO: 7

If you added to your build.gradle file a room.schemaLocation:

```
android { defaultConfig { javaCompileOptions { annotationProcessorOptions {
arguments = ["room.schemaLocation": "$projectDir/schemas".toString()]
}
}
}
}
```

Then, you build your app or module.

As a result you got a json file, with such path to it:

app/schemas/your\_app\_package/db\_package/DbClass/DB\_VERSION.json

What are the correct statements about this file? (Choose all that apply.)

- A. It's a file with Room-exported schema
- B. Main JSONObject in this file usually should contain a number "formatVersion" and a JSONObject "database"
- C. The JSONObject "database" in this file usually should contain such objects, like "entities", "views", "setupQueries", etc.

## ANSWER: A B C

### Explanation:

Exported schema file example:

```
{
"formatVersion": 1,
"database": {
"version": 1,
"identityHash": "d90c93040756d2b94a178d5555555555",
"entities": [
{
"tableName": "tea_table",
"createSql": "CREATE TABLE IF NOT EXISTS `${TABLE_NAME}` (`id` INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL, `name` TEXT, `type` TEXT,
`origin` TEXT, `steep_times` INTEGER, `Description` TEXT, `ingredients` TEXT, `cafeinLevel` TEXT, `favorite` INTEGER)",
```

```
"fields": [  
  {  
    "fieldPath": "mId",  
    "columnName": "id",  
    "affinity": "INTEGER",  
    "notNull": true  
  },  
  {  
    "fieldPath": "mName",  
    "columnName": "name",  
    "affinity": "TEXT",  
    "notNull": false  
  },  
  {  
    "fieldPath": "mType",  
    "columnName": "type",  
    "affinity": "TEXT",  
    "notNull": false  
  },  
  {  
    "fieldPath": "mOrigin",  
    "columnName": "origin",  
    "affinity": "TEXT",  
    "notNull": false  
  },  
  {  
    "fieldPath": "mSteepTimeMs",  
    "columnName": "steep_times",  
    "affinity": "INTEGER",  
    "notNull": false  
  }  
]
```

```
},
{
  "fieldPath": "mDescription",
  "columnName": "Description",
  "affinity": "TEXT",
  "notNull": false
},
{
  "fieldPath": "mIngredients",
  "columnName": "ingredients",
  "affinity": "TEXT",
  "notNull": false
},
{
  "fieldPath": "mCaffeineLevel",
  "columnName": "cafeinLevel",
  "affinity": "TEXT",
  "notNull": false
},
{
  "fieldPath": "mFavorite",
  "columnName": "favorite",
  "affinity": "INTEGER",
  "notNull": false
}
],
"primaryKey": {
  "columnNames": [
    "id"
  ]
},
],
```



**B.** It becomes read-only and you cannot modify its values.

**C.** It becomes read-only, but you cannot see its updated values before updating the data by clicking the Refresh table button at the top of the inspector window.

**ANSWER: B**

## QUESTION NO: 10

Select correct statements about Hardware Abstraction Layer (HAL). (Choose two.)

**A.** The HAL provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework.

**B.** The HAL function both as apps for users and to provide key capabilities that developers can access from their own app. For example, if your app would like to deliver an SMS message, you don't need to build that functionality yourself – you can instead invoke whichever SMS app is already installed to deliver a message to the recipient you specify

**C.** The HAL consists of multiple library modules, each of which implements an interface for a specific type of hardware component, such as the camera or bluetooth module. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component.

**D.** Using a HAL, not using a Linux kernel, allows Android to take advantage of key security features and allows device manufacturers to develop hardware drivers for a well-known kernel.

**ANSWER: A C**

### Explanation:

The system apps function both as apps for users and to provide key capabilities that developers can access from their own app. For example, if your app would like to deliver an SMS message, you don't need to build that functionality yourself — you can instead invoke whichever SMS app is already installed to deliver a message to the recipient you specify Using a Linux kernel allows Android to take advantage of key security features and allows device manufacturers to develop hardware drivers for a well-known kernel. Reference:

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