



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.geekcert.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



VCE & PDF

GeekCert.com

<https://www.geekcert.com/associate-android-developer.html>
2024 Latest geekcert ASSOCIATE-ANDROID-DEVELOPER PDF and VCE
dumps Download

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



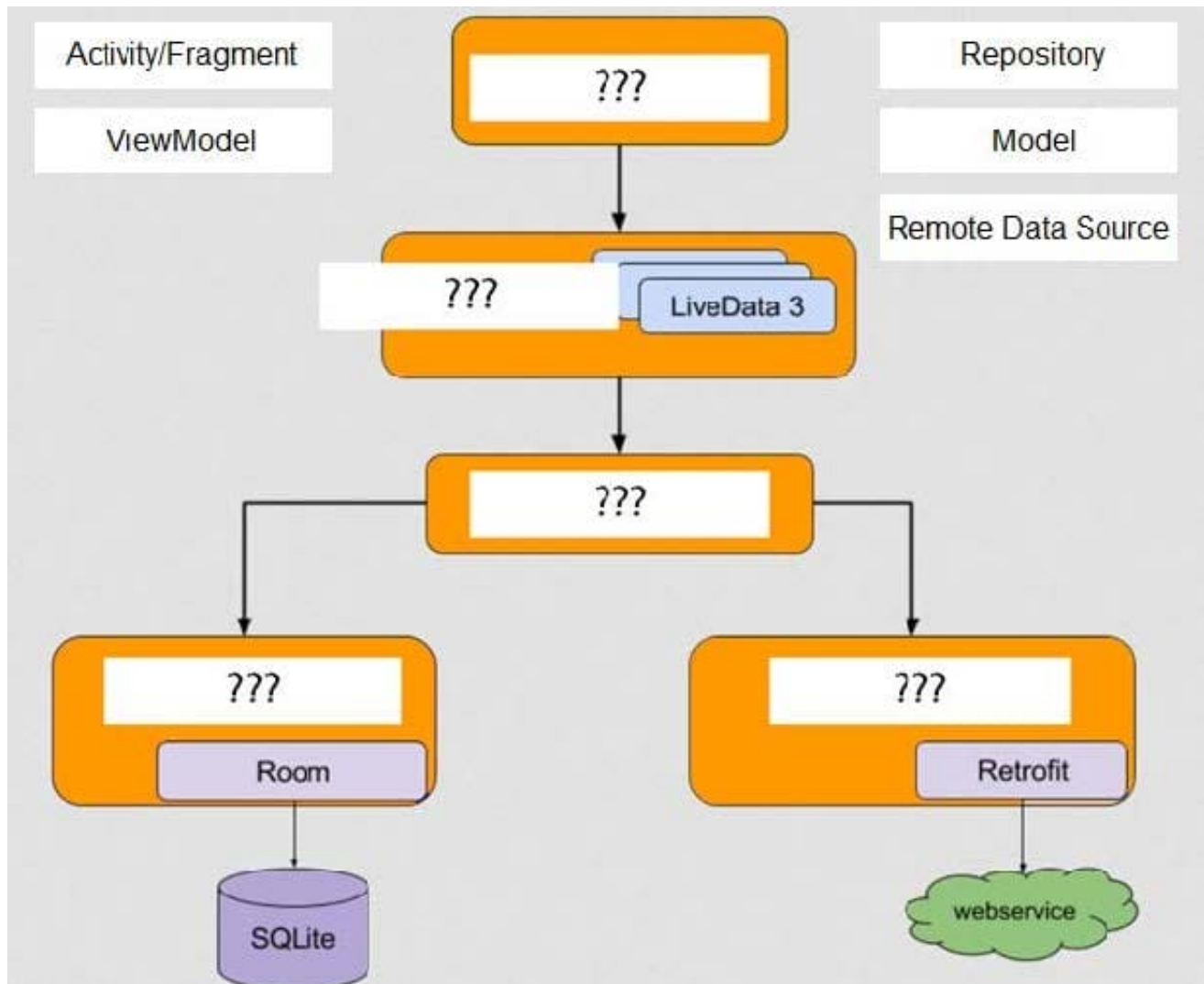


QUESTION 1

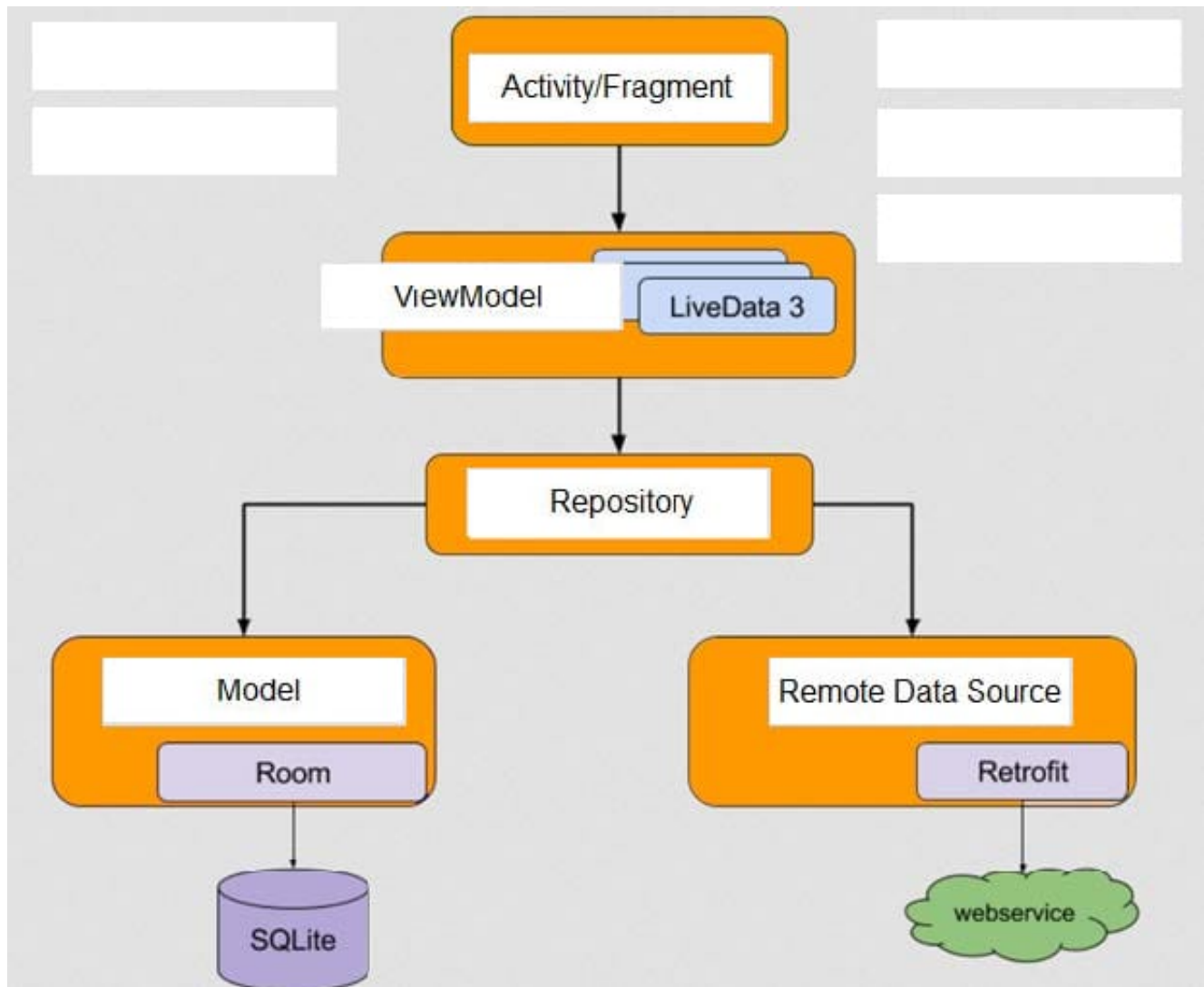
DRAG DROP

With recommended app architecture. Fill the the following diagram, which shows how all the modules usually should interact with one another after designing the app (drag modules to correct places).

Select and Place:



Correct Answer:



QUESTION 2

Choose the most correct statement.

- A. Android is a closed source, Linux-based software stack created for a wide array of devices and form factors.
- B. Android is a closed source, Windows-based software stack created for a wide array of devices and form factors.
- C. Android is an open source, Linux-based software stack created for a wide array of devices and form factors.
- D. Android is an open source software stack created for a highly limited array of devices and form factors.

Correct Answer: C

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



QUESTION 3

In general, you should send an AccessibilityEvent whenever the content of your custom view changes. For example, if you are implementing a custom slider bar that allows a user to select a numeric value by pressing the left or right arrows, your custom view should emit an event of type TYPE_VIEW_TEXT_CHANGED whenever the slider value changes. Which one of the following sample codes demonstrates the use of the sendAccessibilityEvent() method to report this event.

A. @Override

```
public boolean dispatchPopulateAccessibilityEvent(AccessibilityEvent event) {  
  
    boolean completed = super.dispatchPopulateAccessibilityEvent(event);  
  
    CharSequence text = getText();  
  
    if (!TextUtils.isEmpty(text)) {  
  
        event.getText().add(text);  
  
        return true;  
    }  
  
    return completed;  
}
```

B. @Overridepublic boolean onKeyUp (int keyCode, KeyEvent event) {

```
if (keyCode == KeyEvent.KEYCODE_DPAD_LEFT) {  
  
    currentValue--;  
  
    sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_TEXT_CHANGED);  
  
    return true;  
}  
  
...  
}
```

C. @Overridepublic boolean onKeyUp (int keyCode, KeyEvent event) {

```
if (keyCode == KeyEvent.KEYCODE_ENTER) {  
  
    currentValue--;  
  
    sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_CONTEXT_CLICKED);  
  
    return true;  
}  
  
...
```



}

Correct Answer: B

Reference: <https://developer.android.com/guide/topics/ui/accessibility/custom-views>

QUESTION 4

What happens when you create a DAO method and annotate it with @Insert?

Example:

@Dao

public interface MyDao {

@Insert(onConflict = OnConflictStrategy.REPLACE)

public void insertUsers(User... users);}

- A. Room generates an implementation that inserts all parameters into the database in a single transaction.
- B. Room modifies a set of entities, given as parameters, in the database. It uses a query that matches against the primary key of each entity.
- C. Room removes a set of entities, given as parameters, from the database. It uses the primary keys to find the entities to delete.

Correct Answer: A

QUESTION 5

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)
```

```
}
```

```
});
```

Correct Answer: B

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

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

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