

100% Money Back
Guarantee

Vendor:Google

Exam Code:ASSOCIATE-ANDROID-DEVELOPER

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

Version:Demo

QUESTION 1

As an example. In an Activity we have our TimerViewModel object (extended ViewModel), named mTimerViewModel. mTimerViewModel.timer 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!!.timer.value.toString().observe (Observer { aLong -> callAnyChangeUIMethodHere(aLong!!) })`
- B. `mTimerViewModel!!.timer.observe (this, Observer { aLong -> callAnyChangeUIMethodHere(aLong!!) })`
- C. `mTimerViewModel.observe (Observer { aLong -> callAnyChangeUIMethodHere(aLong!!) })`

Correct Answer: B

QUESTION 2

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.

Correct Answer: AC

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>

QUESTION 3

RecyclerView is a subclass of ViewGroup and is a more resource-efficient way to display scrollable lists. Instead of creating a View for each item that may or may not be visible on the screen, RecyclerView:

- A. creates a single list item and reuses it for visible content.
- B. creates an unlimited number of list items and never reuses them
- C. creates a limited number of list items and reuses them for visible content.

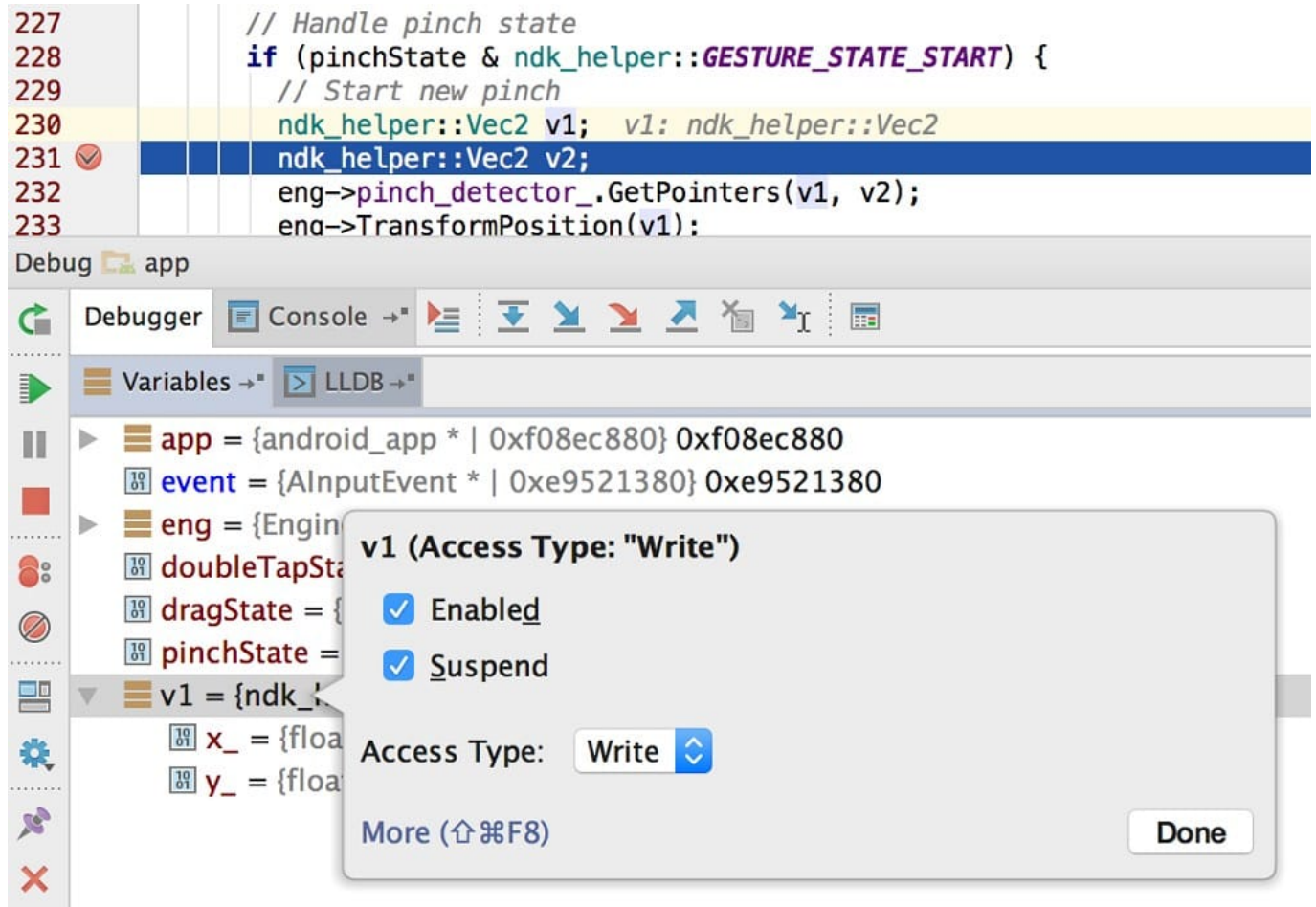
D. creates a single list item and never reuses it

Correct Answer: C

Reference: <https://developer.android.com/guide/topics/ui/layout/recyclerview>

QUESTION 4

```
227 // Handle pinch state
228 if (pinchState & ndk_helper::GESTURE_STATE_START) {
229     // Start new pinch
230     ndk_helper::Vec2 v1; v1: ndk_helper::Vec2
231     ndk_helper::Vec2 v2;
232     eng->pinch_detector_.GetPointers(v1, v2);
233     eng->TransformPosition(v1);
```



The screenshot shows the Android Studio Debugger window. The top pane displays the source code for a pinch state handler. The bottom pane shows the LLDB Variables window. A context menu is open over the variable 'v1', showing options to set its access type to 'Write' and checkboxes for 'Enabled' and 'Suspend'.

What is illustrated in the picture?

- A. Logcat window with filter settings
- B. Debugging native code using LLDB
- C. The Variables and Watches panes in the Debugger window
- D. The Breakpoints window lists all the current breakpoints and includes behavior settings for each
- E. Adding a watchpoint to a variable in memory

Correct Answer: E

QUESTION 5

With a room database. When performing queries, you'll often want your app's UI to update automatically when the data changes. Can you use a return value of type LiveData in your query method description to achieve this?

A. Yes

B. No

Correct Answer: A

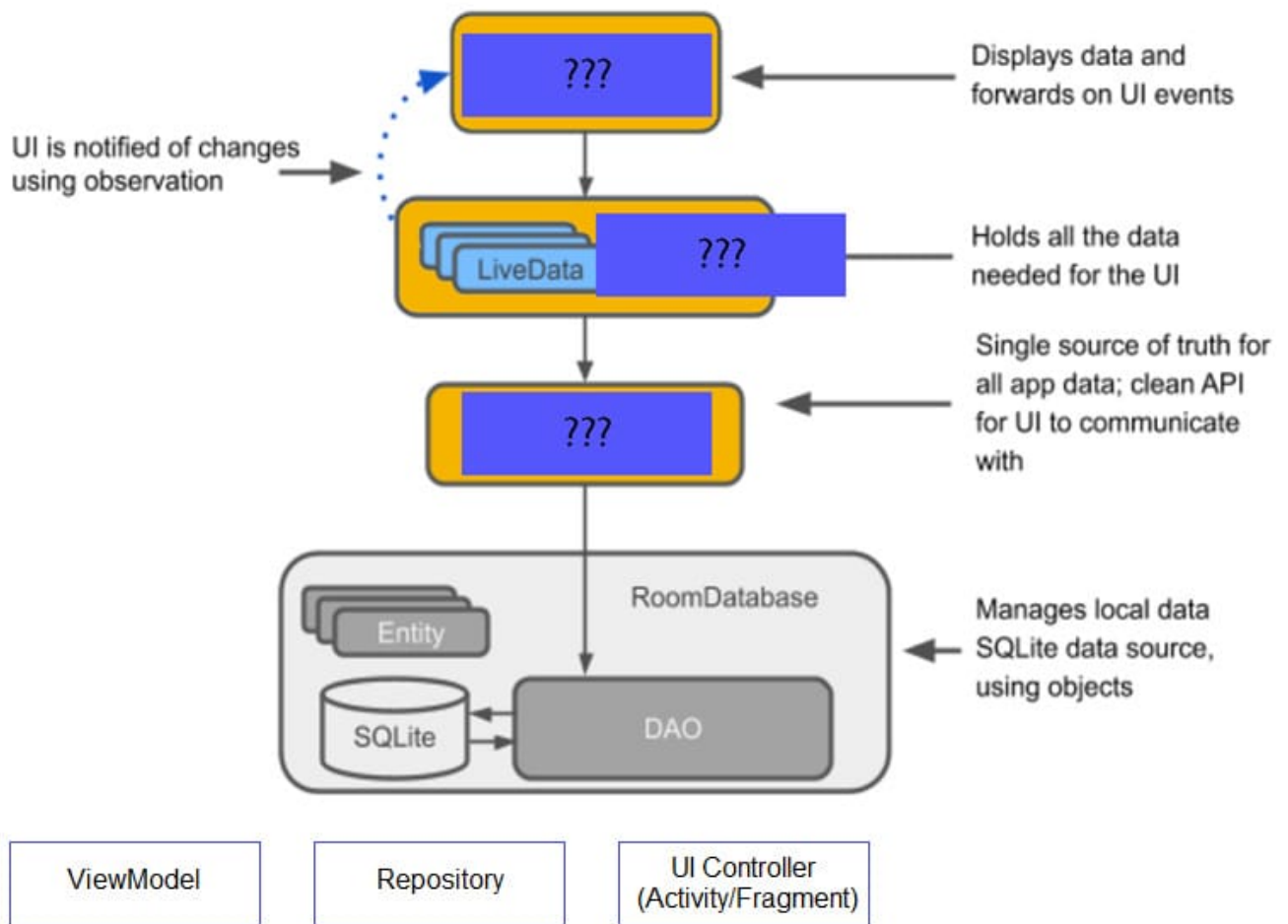
Room generates all necessary code to update the LiveData when the database is updated.

QUESTION 6

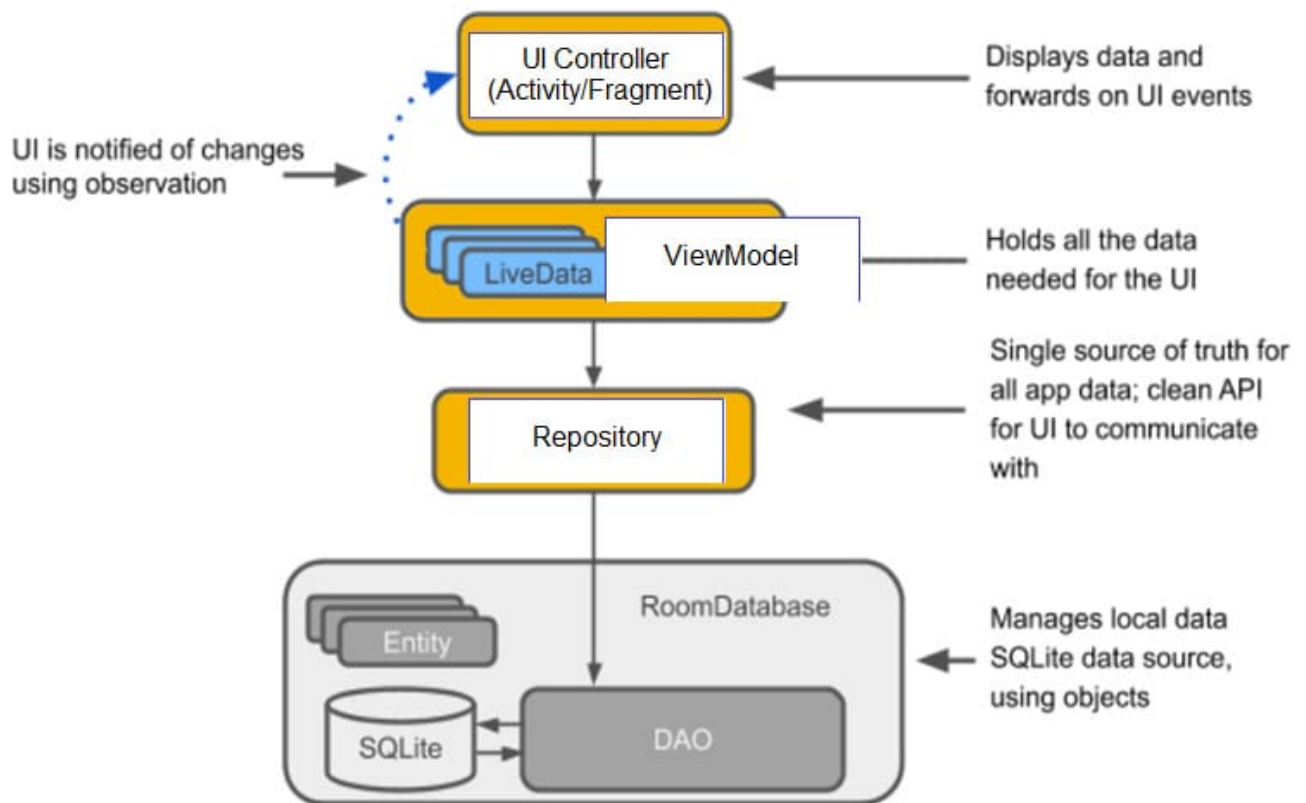
DRAG DROP

The diagram below shows a basic form of the recommended architecture for apps that use Architecture Components. The architecture consists of a UI controller, a ViewModel that serves LiveData, a Repository, and a Room database. Drag modules to correct places.

Select and Place:



Correct Answer:



QUESTION 7

When using an ImageView, ImageButton, CheckBox, or other View that conveys information graphically. What attribute to use to provide a content label for that View?

- A. android:contentDescription
- B. android:hint
- C. android:labelFor

Correct Answer: A

Reference: <https://support.google.com/accessibility/android/answer/7158690?hl=en>

QUESTION 8

Enable debugging on your device: If you are using the emulator, this is enabled by default. But for a connected device,

you need to

- A. enable transfer data from the device in usb connection options.
- B. enable debugging in the device developer options.
- C. enable connection in bluetooth options.

Correct Answer: B

QUESTION 9

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 fun dispatchPopulateAccessibilityEvent(event: AccessibilityEvent): Boolean {return super.dispatchPopulateAccessibilityEvent(event).let { completed ->

```
if (text?.isNotEmpty() == true) {
```

```
event.text.add(text)
```

```
true
```

```
} else {
```

```
completed
```

```
}
```

```
}
```

```
}
```

B. override fun onKeyUp(keyCode: Int, event: KeyEvent): Boolean {return when(keyCode) {

```
KeyEvent.KEYCODE_DPAD_LEFT -> {
```

```
currentValue.sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_TEXT_CHANGED)
```

```
true
```

```
}
```

```
...
```

```
}
```

```
}
```

C. override fun onKeyUp(keyCode: Int, event: KeyEvent): Boolean {return when(keyCode) {

```
KeyEvent.KEYCODE_ENTER -> {  
    currentValue.sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_CONTEXT_CLICKED)  
    true  
}  
...  
}  
}
```

Correct Answer: B

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

QUESTION 10

When your code execution reaches the breakpoint, Android Studio pauses execution of your app. You can then use the tools in the Debugger tab to identify the state of the app. With Step Into



you can

- A. examine the object tree for a variable, expand it in the Variables view. If the Variables view is not visible
- B. evaluate an expression at the current execution point
- C. advance to the next line in the code (without entering a method)
- D. advance to the first line inside a method call
- E. advance to the next line outside the current method
- F. continue running the app normally

Correct Answer: D

QUESTION 11

For example, our preferences.xml file was added by `addPreferencesFromResource(R.xml.preferences)`. Our preferences.xml file contains such item:

In our Fragment, we can dynamically get current notification preference value in this way:


```
A. boolean isNotificationOn = PreferenceManager.getDefaultSharedPreferences(getContext()).getBoolean(getContext().getString(R.string.pref_notification_key),getContext().getResources().getBoolean(R.bool.pref_notification_default_value)
```

```
);
```

```
B. boolean isNotificationOn = PreferenceManager.getSharedPreferences(getContext()).getBoolean(getContext().getString(R.string.pref_notification_default_value),getContext().getString(R.string.pref_notification_key)
```

```
);
```

```
C. boolean isNotificationOn = PreferenceManager.getSharedPreferences(getContext()).getBoolean(getContext().getResources().getBoolean(R.bool.pref_notification_default_value),getContext().getString(R.string.pref_notification_key)
```

```
);
```

Correct Answer: A

QUESTION 12

In our TeaViewModel class, that extends ViewModel, we have such method:

```
public LiveData getTea() {return mTea;}
```

An observer in our Activity (type of mViewModelvariable in example is TeaViewModel) is set in this way:

```
mViewModel.getTea().observe(this, this::displayTea);
```

What will be a correct displayTea method definition?

A. private void displayTea()

B. private void displayTea(Tea tea)

C. private void displayTea(LiveData)

D. private void displayTea(LiveData)

Correct Answer: B