



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.pass4itsure.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

Pass4itSure.com

<https://www.pass4itsure.com/associate-android-developer.html>
2024 Latest pass4itsure 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

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 2

What is a correct part of an Implicit Intent for sharing data implementation?

- A. Intent sendIntent = new Intent(this, UploadService.class) sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);
- B. Intent sendIntent = new Intent(); sendIntent.setType(Intent.ACTION_SEND);
- C. Intent sendIntent = new Intent(this, UploadService.class) sendIntent.setData(Uri.parse(fileUrl));
- D. Intent sendIntent = new Intent(); sendIntent.setAction(Intent.ACTION_SEND);

Correct Answer: D

Create the text message with a string

```
Intent sendIntent = new Intent();  
  
sendIntent.setAction(Intent.ACTION_SEND);  
  
sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);  
  
sendIntent.setType("text/plain");
```

Reference:

<https://developer.android.com/guide/components/fundamentals>

QUESTION 3

An example. In our ViewModelFactory(that implements ViewModelProvider.Factory) we have an instance of our



Repository, named mRepository. Our ViewModel has such constructor:

```
class MyViewModel(private val mRepository: MyRepository) : ViewModel() ...
```

Next, in our ViewModelFactory create ViewModel method (overridden) looks like this:

```
override fun create(modelClass: Class): T { return try { //MISSED RETURN VALUE HERE" } catch (e:
InstantiationException) { throw RuntimeException("Cannot create an instance of $modelClass", e) } catch (e:
IllegalAccessException) { throw RuntimeException("Cannot create an instance of $modelClass", e) } catch (e:
NoSuchMethodException) { throw RuntimeException("Cannot create an instance of $modelClass", e) } catch (e:
InvocationTargetException) { throw RuntimeException("Cannot create an instance of $modelClass", e) } }
```

What should we write instead of “//MISSED RETURN VALUE HERE”?

- A. modelClass.getConstructor().newInstance(mRepository)
- B. modelClass.getConstructor(MyRepository::class.java).newInstance()
- C. modelClass.getConstructor(MyRepository::class.java).newInstance(mRepository)

Correct Answer: C

QUESTION 4

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 Out



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: E

QUESTION 5



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

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

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

[ASSOCIATE-ANDROID-DEVELOPER Braindumps](#)