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

⚙ **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

LiveData.postValue() and LiveData.setValue() methods have some differences. So if you have a following code executed in the main thread:

liveData.postValue("a"); liveData.setValue("b");

What will be the correct statement?

A. The value "b" would be set at first and later the main thread would override it with the value "a".

B. The value "a" would be set at first and later the main thread would override it with the value "b".

C. The value "b" would be set at first and would not be overridden with the value "a".

D. The value "a" would be set at first and would not be overridden with the value "b".

Correct Answer: B

QUESTION 3

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 4

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 5

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

ASSOCIATE-ANDROID-
DEVELOPER VCE Dumps

ASSOCIATE-ANDROID-
DEVELOPER Practice Test

ASSOCIATE-ANDROID-
DEVELOPER Exam
Questions