Google.Associate Android Developer.vMay-2024.by.Athot.43q

Number: Associate Android Develope
Passing Score: 800
Time Limit: 120
File Version: 3.0

Certification: Associate Android Developer
Certification Full Name: Associate Android Developer

Exam A

QUESTION 1
To handle an options menu item click in an activity, we usually should override method named:

A. onKey
B. onClick
C. onOptionsItemSelected

Correct Answer: C
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/menus

QUESTION 2
Working with Custom View. To define custom attributes, we can add <declare-styleable> resources to our project. It is customary to put these resources into a file:

A. res/layout/attrs.xml
B. res/values/attrs.xml
C. res/raw/attrs.xml
D. res/xml/attrs.xml

Correct Answer: B
Section:
Explanation:
Reference: https://developer.android.com/guide/topics/ui/custom-components

QUESTION 3
In application theme style, flag windowActionBar (<item name="windowActionBar">) indicates:

A. whether the given application component is available to other applications.
B. whether action modes should overlay window content when there is not reserved space for their UI (such as an Action Bar).
C. whether this window's Action Bar should overlay application content.
D. whether this window should have an Action Bar in place of the usual title bar.

Correct Answer: D
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/look-and-feel/themes https://developer.android.com/reference/android/R.styleable.html

QUESTION 4
In application theme style, flag windowNoTitle (<item name="windowNoTitle">) indicates:

A. whether this window should have an Action Bar in place of the usual title bar.
B. whether there should be no title on this window.
C. that this window should not be displayed at all.
D. whether this is a floating window.
E. whether this Window is responsible for drawing the background for the system bars.

Correct Answer: B
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/look-and-feel/themes https://developer.android.com/reference/android/R.styleable.html

QUESTION 5
In application theme style, flag windowDrawsSystemBarBackgrounds (<item name="android:windowDrawsSystemBarBackgrounds">) indicates:

A. whether this window should have an Action Bar in place of the usual title bar.
B. whether there should be no title on this window.
C. that this window should not be displayed at all.
D. whether this is a floating window.
E. whether this Window is responsible for drawing the background for the system bars.

Correct Answer: E
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/look-and-feel/themes https://developer.android.com/reference/android/R.styleable.html

QUESTION 6
In application theme style, value statusBarColor (<item name="android:statusBarColor">) means:

A. Color of text (usually same as colorForeground).
B. Shows a thin line of the specified color between the navigation bar and the app content.
For this to take effect, the window must be drawing the system bar backgrounds with R.attr.windowDrawsSystemBarBackgrounds and the navigation bar must not have been requested to be translucent with R.attr.windowTranslucentNavigation. Corresponds to Window.setNavigationBarDividerColor(int).
C. The color for the status bar. If the color is not opaque, consider setting View.SYSTEM_UI_FLAG_LAYOUT_STABLE and View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN. For this to take effect, the window must be drawing thesystem bar backgrounds with
D. attr.windowDrawsSystemBarBackgrounds and the status bar must not have been requested to be translucent with R.attr.windowTranslucentStatus.
Corresponds to Window.setStatusBarColor(int).
E. The color for the navigation bar. If the color is not opaque, consider setting View.SYSTEM_UI_FLAG_LAYOUT_STABLE and View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION. For this to take effect, the window must be drawingthe system bar backgrounds with
F. attr.windowDrawsSystemBarBackgrounds and the navigation bar must not have been requested to be translucent with R.attr.windowTranslucentNavigation.
Corresponds to Window.setNavigationBarColor(int).

Correct Answer: C
Section:
Explanation:
Reference:

https://developer.android.com/guide/topics/ui/look-and-feel/themes https://developer.android.com/reference/android/R.styleable.html

QUESTION 7
An overridden method onCreateOptionsMenu in an Activity returns boolean value. What does this value mean?

A. You must return true for the menu to be displayed; if you return false it will not be shown.
B. You must return false for the menu to be displayed; if you return true it will not be shown.
C. You can return any value: the menu will be displayed anyway.

Correct Answer: A
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/menus

QUESTION 8
Working with Custom View. Once you define the custom attributes, you can use them in layout XML files just like built-in attributes. The only difference is that your custom attributes belong to a different namespace. Instead of belonging to the http://schemas.android.com/apk/res/android namespace, they belong to:

A. http://schemas.android.com/apk/res/[your package name]
B. http://schemas.android.com/apk/[your package name]
C. http://schemas.android.com/[your package name]

Correct Answer: A
Section:
Explanation:
Reference: https://developer.android.com/guide/topics/ui/custom-components

QUESTION 9
We have a custom view that extends android.widget.ProgressBar. Our progress bar is not touchable, focusable, etc.: it just shows progress. Style for our custom progress bar extends "Widget.AppCompat.ProgressBar.Horizontal". An item, named "progressDrawable", in our style, is a xml file . What we usually can see as a main single element in this xml file:

A. A State List (<selector> element)
B. A Layer List (<layer-list> element) with items android:id="@+id/progress" and android:id="@+id/background" inside it.
C. An <ImageView> element with android:id="@+id/progress" identifier

Correct Answer: B
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/resources/drawable-resource

QUESTION 10
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
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/layout/recyclerview

QUESTION 11
Android uses adapters (from the Adapter class) to connect data with View items in a list. There are many different kinds of adapters available, and you can also write custom adapters. To connect data with View items, the adapter needs to know about the View items. From what is extended the entity that is usually used in an adapter and describes a View item and its position within the RecyclerView?

A. RecyclerView.AdapterDataObserver
B. RecyclerView.ItemDecoration
C. RecyclerView.ViewHolder
D. RecyclerViewAccessibilityDelegate

Correct Answer: C
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/layout/recyclerview

QUESTION 12
Android Tests. You can use the childSelector() method to nest multiple UiSelector instances. For example, the following code example shows how your test might specify a search to find the first ListView in the currently displayed UI, then search within that ListView to find a UI element with the text property Apps. What is the correct sample?

A.
```
val appItem: UiObject = device.findObject(
UiSelector().className(ListView.class)
.instance(1)
.childSelector(
UiSelector().text("Apps")
)
)
```
B.
```
val appItem: UiObject = device.findObject(
UiSelector().className("android.widget.ListView")
.instance(0)
.childSelector(
UiSelector().text("Apps")
)
)
```
C.
```
val appItem: UiObject = device.findObject(
UiSelector().className("android.widget.ListView")
.instance(
UiSelector().text("Apps")
)
)
```

Correct Answer: B

Section:

QUESTION 13
The following code snippet shows an example of an Espresso test:

A. @Rule fun greeterSaysHello() {
onView(withId(R.id.name_field)).do(typeText("Steve")) onView(withId(R.id.greet_button)).do(click())
onView(withText("Hello Steve!")).check(matches(isDisplayed())) }
B. @Test fun greeterSaysHello() {
onView(withId(R.id.name_field)).perform(typeText("Steve")) onView(withId(R.id.greet_button)).perform(click())
onView(withText("Hello Steve!")).check(matches(isDisplayed())) }
C. @Test fun greeterSaysHello() {
onView(withId(R.id.name_field)).do(typeText("Steve")) onView(withId(R.id.greet_button)).do(click())
onView(withText("Hello Steve!")).compare(matches(isDisplayed())) }

Correct Answer: B
Section:

QUESTION 14
As an example. In an Activity we have our TimerViewModel object (extended ViewModel), named mTimerViewModel. mTimerViewModel.timer method returns a LiveData<Long> 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
Section:

QUESTION 15
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
Section:

QUESTION 16
In our TeaViewModel class, that extends ViewModel, we have such prorerty: val tea: LiveData<Tea>
An observer in our Activity (type of mViewModel variable in example is TeaViewModel) is set in this way:

mViewModel!!.tea.observe(this, Observer { tea: Tea? -> displayTea(tea) })
What will be a correct displayTea method definition?

A. private fun displayTea()
B. private fun displayTea(tea: Tea?)
C. private fun displayTea(tea: LiveData?<Tea>)
D. private fun displayTea(tea: LiveData?<T>)

Correct Answer: B
Section:

QUESTION 17
For example, our preferences.xml file was added by addPreferencesFromResource(R.xml.preferences). Our preferences.xml file contains such item:
<SwitchPreference android:id="@+id/notification" android:key="@string/pref_notification_key" android:title="@string/pref_notification_title" android:summary="@string/pref_notification_summary" android:defaultValue="@bool/pref_notification_default_value" app:iconSpaceReserved="false"/>
In our Fragment, we can dynamically get current notification preference value in this way:

A. val isNotificationOn = PreferenceManager.getDefaultSharedPreferences(context).getBoolean(context!!.getString(R.string.pref_notification_key), context!!.resources.getBoolean(R.bool.pref_notification_default_value))
B. val isNotificationOn = PreferenceManager.getSharedPreferences(context).getBoolean(context!!.getString(R.string.pref_notification_default_value), context!!.getString(R.string.pref_notification_key),)
C. val isNotificationOn = PreferenceManager.getSharedPreferences(context).getBoolean(context!!.resources.getBoolean (R.bool.pref_notification_default_value), context!!.getString(R.string.pref_notification_key))

Correct Answer: A
Section:

QUESTION 18
For example, our preferences.xml file was added by addPreferencesFromResource(R.xml.preferences). Our preferences.xml file contains such item:
<ListPreference android:id="@+id/order_by" android:key="@string/pref_sort_key" android:title="@string/pref_sort_title" android:summary="@string/pref_sort_summary" android:dialogTitle="@string/pref_sort_dialog_title" android:entries="@array/sort_oder" android:entryValues="@array/sort_oder_value" android:defaultValue="@string/pref_default_sort_value" app:iconSpaceReserved="false" />
In our Fragment, we can dynamically get current notification preference value in this way:

A. val sortBy = PreferenceManager.getDefaultSharedPreferences(context).getString(context!!.getString(R.string.pref_sort_key), context!!.resources.getBoolean(R.bool.pref_default_sort_value))
B. val sortBy = PreferenceManager.getSharedPreferences(context).getString(context!!.getString(R.string.pref_default_sort_value), context!!.getString (R.string.pref_sort_key),)
C. val sortBy = PreferenceManager.getSharedPreferences(context).getBoolean(context!!.resources.getBoolean(R.bool.pref_default_sort_value), context!!.getString(R.string.pref_sort_key))
D. val sortBy = PreferenceManager.getDefaultSharedPreferences(context).getString(context!!.getString(R.string.pref_sort_key), context!!.getString (R.string.pref_default_sort_value))

Correct Answer: D
Section:

QUESTION 19
For example, we have a file in our raw folder app/src/main/res/raw/sample_teas.json. To get an InputStream for reading it, from out Context context, we can do this:

A. val input = context!!.openRawResource(R.raw.sample_teas)
B. val input = context!!.getRawResource(R.raw.sample_teas)
C. val input = context!!.resources.openRawResource(R.raw.sample_teas)

Correct Answer: C
Section:

QUESTION 20
For example, we have a BufferedReader reader, associated with the json file through InputStreamReader. To get a file data we can do this:

A. var line: String?
try {
while (reader.readLine().also { line = it } != null) {
builder.append(line)
}
val json = JSONObject(builder.toString()) return json
} catch (exception: IOException) { exception.printStackTrace() } catch (exception: JSONException) { exception.printStackTrace()
}
B. var line: JSONObject ? try {
while (reader.readJSONObject ().also { line = it } != null) { builder.append(line)
}
val json = JSONObject(builder.toString()) return json
} catch (exception: IOException) { exception.printStackTrace() } catch (exception: JSONException) { exception.printStackTrace()
}
C. var line: String? try {
while (reader.readLine().also { line = it } != null) { builder.append(line)
}
val json = JSONObject(builder.toString()) return json
} catch (exception: RuntimeException) { exception.printStackTrace()
} catch (exception: ArrayIndexOutOfBoundsException) { exception.printStackTrace()
}

Correct Answer: A
Section:

QUESTION 21
Which build options in the Build menu to choose to delete all intermediate/cached build files.

A. Make Module
B. Generate Signed Bundle / APK
C. Rebuild Project
D. Clean Project
E. Make Project

Correct Answer: D
Section:
Explanation:

Reference: https://developer.android.com/studio/run

QUESTION 22
If you want get a debuggable APK that people can install without adb, in Android Studio you can:

A. Select your debug variant and click Build Bundle(s) / APK(s) > Build APK(s).
B. Click the Run button from toolbar
C. Select your debug variant and click Analyze APK.

Correct Answer: A
Section:
Explanation:
The Run button builds an APK with testOnly="true", which means the APK can only be installed via adb (which Android Studio uses). If you want a debuggable APK that people can install without adb, select your debug variant and click Build Bundle(s) / APK(s) > Build APK(s). Reference: https://developer.android.com/studio/run

QUESTION 23
To build a debug APK, you can open a command line and navigate to the root of your project directory. To initiate a debug build, invoke the assembleDebug task:
gradlew assembleDebug
This creates an APK named [module_name]-debug.apk in
[project_name]/[module_name]/build/outputs/apk/
Select correct statements about generated file. (Choose all that apply.)

A. The file is already signed with the debug key
B. The file is already aligned with zipalign
C. You can immediately install this file on a device.

Correct Answer: A, B, C
Section:
Explanation:
Reference:
https://developer.android.com/studio/run

QUESTION 24
Building your app from the command line, if you have a "demo" product flavor, then you can build the debug version with the command:

A. gradlew assembleDemoDebug
B. gradlew installDemoDebug
C. both variants are correct.

Correct Answer: C
Section:
Explanation:
Before immediately install build on a running emulator or connected device, installDemoDebug cause an APK building. Reference:
https://developer.android.com/studio/run

QUESTION 25
If no any folder like res/anim-<qualifiers>, res/drawable-<qualifiers>, res/layout-<qualifiers>, res/raw-<qualifiers>, res/xml-<qualifiers> exist in the project. Which folders are required in the project anyway? (Choose two.)

A. res/anim/
B. res/drawable/
C. res/layout/
D. res/raw/
E. res/xml/

Correct Answer: B, C
Section:
Explanation:
Reference: https://developer.android.com/guide/topics/resources/localization

QUESTION 26
Assume that an app includes a default set of graphics and two other sets of graphics, each optimized for a different device setup:
res/drawable/
Contains default graphics.
res/drawable-small-land-stylus/
Contains graphics optimized for use with a device that expects input from a stylus and has a QVGA low-density screen in landscape orientation.
res/drawable-ja/
Contains graphics optimized for use with Japanese.
What happens if the app runs on a device that is configured to use Japanese and, at the same time, the device happens to be one that expects input from a stylus and has a QVGA low-density screen in landscape orientation?

A. Android loads graphics from res/drawable/
B. Android loads graphics from res/drawable-small-land-stylus/
C. Android loads graphics from res/drawable-ja/

Correct Answer: C
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/resources/localization

QUESTION 27
Assume that you have the following situation:
The app code calls for R.string.text_a
Three relevant resource files are available:
- res/values/strings.xml, which includes text_a in the app's default language, in this case English.
- res/values-mcc404/strings.xml, which includes text_a in the app's default language, in this case English.
- res/values-hi/strings.xml, which includes text_a in Hindi.
The app is running on a device that has the following configuration:
- The SIM card is connected to a mobile network in India (MCC 404).
- The language is set to Hindi (hi).
Which is the correct statement below?

A. Android loads text_a from res/values/strings.xml (in English)
B. Android loads text_a from res/values-mcc404/strings.xml (in English)
C. Android loads text_a from res/values-hi/strings.xml (in Hindi)

Correct Answer: B
Section:
Explanation:
Android loads text_a from res/values-mcc404/strings.xml (in English), even if the device is configured for Hindi. That is because in the resource-selection process, Android prefers an MCC match over a language match (as a priority Exception).
Reference:
https://developer.android.com/guide/topics/resources/localization

QUESTION 28
What is the placeholder tag <xliff:g> used for?

A. To mark text that should not be translated.
B. To raise a translation priority to a higher level
C. To raise a quantity of translations for the string
D. To pick up and move sting translation from a different resource file

Correct Answer: A
Section:
Explanation:
Reference: https://developer.android.com/guide/topics/resources/localization

QUESTION 29
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
Section:
Explanation:
Reference: https://developer.android.com/guide/platform

QUESTION 30
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: A, C
Section:
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 â€" 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 31
Custom views and directional controller clicks. On most devices, clicking a view using a directional controller sends (to the view currently in focus) a KeyEvent with:

A. KEYCODE_DPAD_CENTER
B. KEYCODE_BUTTON_START
C. KEYCODE_CALL
D. KEYCODE_BUTTON_SELECT

Correct Answer: A
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/accessibility/custom-views

QUESTION 32
Custom views and directional controller clicks. In general, you should send an AccessibilityEvent whenever the content of your custom view changes. For example, if a text value was changed in your custom view, you should emit an event of this type:

A. TYPE_WINDOWS_CHANGED
B. TYPE_VIEW_CONTEXT_CLICKED
C. TYPE_WINDOWS_CHANGED
D. TYPE_VIEW_TEXT_CHANGED

Correct Answer: D
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/accessibility/custom-views

QUESTION 33
Select a correct statement about PagedList.

A. PagedList is content-mutable. This means that new content can be loaded into an instance of PagedList and the loaded items themselves can change once loaded.
B. PagedList is content-immutable. This means that, although new content can be loaded into an instance of PagedList, the loaded items themselves cannot change once loaded.
C. PagedList is content-accidental. This means that new content can be loaded into an instance of PagedList and the loaded items themselves can be changed to accidental values randomly.

Correct Answer: B
Section:
Explanation:
Reference:
https://developer.android.com/topic/libraries/architecture/paging/ui

QUESTION 34
If content in a PagedList updates, the PagedListAdapter object receives:

A. only one item from PagedList that contains the updated information.
B. one or more items from PagedList that contains the updated information.
C. a completely new PagedList that contains the updated information.

Correct Answer: C
Section:
Explanation:
Reference:
https://developer.android.com/topic/libraries/architecture/paging/ui

QUESTION 35
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

Correct Answer: C
Section:
Explanation:
Reference:
https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout

QUESTION 36
Which statement is most true about layout_constraintLeft_toRightOf and layout_constraintStart_toEndOf constraints ?

A. layout_constraintLeft_toRightOf is equal to layout_constraintStart_toEndOf in any case
B. layout_constraintLeft_toRightOf is equal to layout_constraintStart_toEndOf in case if user choose a language that uses right-to-left (RTL) scripts, such as Arabic or Hebrew, for their UI locale
C. layout_constraintLeft_toRightOf is equal to layout_constraintStart_toEndOf in case if user choose a language that uses left-to-right (LTR) scripts, such as English or French, for their UI locale
D. layout_constraintLeft_toRightOf works with horizontal axes and layout_constraintStart_toEndOf works with vertical axes

Correct Answer: C
Section:
Explanation:
Reference:
https://developer.android.com/training/basics/supporting-devices/languages

QUESTION 37
In application theme style, flag windowNoTitle (<item name="windowNoTitle">) indicates:

A. whether this window should have an Action Bar in place of the usual title bar.
B. whether there should be no title on this window.
C. that this window should not be displayed at all.
D. whether this is a floating window.
E. whether this Window is responsible for drawing the background for the system bars.

Correct Answer: B
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/look-and-feel/themes https://developer.android.com/reference/android/R.styleable.html

QUESTION 38
"Set the activity content to an explicit view. This view is placed directly into the activity's view hierarchy. It can itself be a complex view hierarchy." This can be done by calling method:

A. findViewById
B. setContentView
C. setActionBar
D. setContentTransitionManager
E. setTheme

Correct Answer: B
Section:
Explanation:
Reference:
https://developer.android.com/training/basics/firstapp/building-ui https://developer.android.com/reference/android/app/Activity

QUESTION 39
A content label sometimes depends on information only available at runtime, or the meaning of a View might change over time. For example, a Play button might change to a Pause button during music playback. In these cases, to update the content label at the appropriate time, we can use:

A. View#setContentDescription(int contentDescriptionResId)
B. View#setContentLabel(int contentDescriptionResId)
C. View#setContentDescription(CharSequence contentDescription)
D. View#setContentLabel(CharSequence contentDescription)

Correct Answer: C
Section:
Explanation:
Reference:
https://support.google.com/accessibility/android/answer/7158690?hl=en

QUESTION 40
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
Section:
Explanation:
Reference:

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

QUESTION 41
When using an EditTexts or editable TextViews, or other editable View. What attribute to use to provide a content label for that View?

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

Correct Answer: B
Section:
Explanation:
Reference: https://support.google.com/accessibility/android/answer/7158690?hl=en

QUESTION 42
Content labels. What attribute to use to indicate that a View should act as a content label for another View?

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

Correct Answer: C
Section:
Explanation:
Reference: https://support.google.com/accessibility/android/answer/7158690?hl=en

QUESTION 43
For example, suppose that in a XML file (res/menu/menu_main.xml as an example), where menu items are described, we have such item:

A. .
<item
android:id="@+id/action_settings" android:orderInCategory="100"
android:title="@string/menu_action_settings" app:showAsAction="never" />
B. .
Attribute "app:showAsAction" shows when and how this item should appear as an action item in the app bar. What value "never" in this attribute means?
C. Only place this item in the app bar if there is room for it. If there is not room for all the items marked by this value, the items with the lowest orderInCategory values are displayed as actions, and the remaining items are displayed in theoverflow menu.
D. Also include the title text (defined by android:title) with the action item. You can include this value along with one of the others as a flag set, by separating them with a pipe.
E. Never place this item in the app bar. Instead, list the item in the app bar's overflow menu.
F. Always place this item in the app bar. Avoid using this unless it's critical that the item always appear in the action bar. Setting multiple items to always appear as action items can result in them overlapping with other UI in the app bar.
G. The action view associated with this action item (as declared by android:actionLayout or android:actionViewClass) is collapsible.

Correct Answer: C
Section:
Explanation:
Reference:
https://developer.android.com/guide/topics/ui/menus