



**Free Questions for Associate-Android-Developer by dumpshq**

**Shared by Carney on 15-04-2024**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

## Question Type: MultipleChoice

---

For example, we have a BufferedReader reader, associated with the json file through

InputStreamReader. To get a file data we can do this:

### Options:

---

**A-** String line; try {

```
while ((line = reader.readLine()) != null) { builder.append(line);
```

```
}
```

```
JSONObject json = new JSONObject(builder.toString());
```

```
return json;
```

```
} catch (IOException | JSONException exception) {
```

```
exception.printStackTrace();
```

```
}
```

**B-** JSONObject line; try {

```
while ((line = reader.readJSONObject ()) != null) { builder.append(line);
```

```
}
```

```
JSONObject json = new JSONObject(builder.toString());
```

```
return json;
```

```
} catch (IOException | JSONException exception) {
```

```
exception.printStackTrace();
}
C- String line; try {
while ((line = reader.readLine()) != null) { builder.append(line);
}
JSONObject json = new JSONObject(builder.toString());
return json;
} catch (RuntimeException|ArrayIndexOutOfBoundsException exception) {
exception.printStackTrace();
}
```

**Answer:**

---

A

## Question 2

---

**Question Type:** MultipleChoice

---

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 our Context context, we can do this:

### Options:

---

- A- `InputStream input = context.openRawResource(R.raw.sample_teas);`
- B- `InputStream input = context.getRawResource(R.raw.sample_teas);`
- C- `InputStream input = context.getResources().openRawResource(R.raw.sample_teas);`

### Answer:

---

C

## Question 3

---

### Question Type: MultipleChoice

---

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

```
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:

## Options:

---

**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));

## Answer:

---

A

## Question 4

---

### Question Type: MultipleChoice

---

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

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

```
}
```

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

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

What will be a correct displayTea method definition?

### Options:

---

- A- private void displayTea()
- B- private void displayTea(Tea tea)
- C- private void displayTea(LiveData<Tea>)
- D- private void displayTea(LiveData<T>)

### Answer:

---

B

## Question 5

---

Question Type: MultipleChoice

---

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?

### Options:

---

- 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'.

### Answer:

---

B

## Question 6

---

**Question Type:** MultipleChoice

---

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?

### Options:

---

```
A- mTimerViewModel.getTimer().getValue().toString().observe(new Observer<Long>() {  
    @Override  
    public void onChanged(Long aLong) {  
        callAnyChangeUIMethodHere(aLong)  
    }  
});
```

```
B- mTimerViewModel.getTimer().observe(this, new Observer<Long>() {  
    @Override  
    public void onChanged(Long aLong) {  
        callAnyChangeUIMethodHere(aLong)  
    }  
});
```

```
C- mTimerViewModel.observe(new Observer<Long>() {  
    @Override  
    public void onChanged(Long aLong) {  
        callAnyChangeUIMethodHere(aLong)  
    }  
});
```



```
});
```

**Answer:**

---

B

## Question 7

---

**Question Type: MultipleChoice**

---

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

**Options:**

---

**A-** @Rule

```
public void 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

```
public void 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

```
public void 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()));
}
```

**Answer:**

---

B

## Question 8

---

**Question Type:** MultipleChoice

---

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?

### Options:

---

**A-** UiObject appItem = device.findObject(new UiSelector()  
.className(ListView.class)  
.instance(1)  
.childSelector(new UiSelector()  
.text('Apps')));

**B-** UiObject appItem = device.findObject(new UiSelector()  
.className('android.widget.ListView')  
.instance(0)  
.childSelector(new UiSelector()  
.text('Apps')));

**C-** UiObject appItem = device.findObject(new UiSelector()  
.className('android.widget.ListView')  
.instance(new UiSelector()  
.text('Apps')));

### Answer:

---

B

## Question 9

---

### Question Type: MultipleChoice

---

The easiest way of adding menu items (to specify the options menu for an activity) is inflating an XML file into the Menu via MenuInflater. With menu\_main.xml we can do it in this way:

#### Options:

---

**A-** @Override

```
public boolean onCreateOptionsMenu(Menu menu) { getMenuInflater().inflate(R.menu.menu_main, menu); return true; }  
}
```

**B-** @Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return super.onOptionsItemSelected(item);  
}
```

**C-** @Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.menu.menu_main);  
}
```

**Answer:**

---

A

**Explanation:**

---

<https://developer.android.com/guide/topics/ui/menus>

## Question 10

---

**Question Type:** MultipleChoice

---

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.

**Options:**

---

### 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- @Override

```
public boolean onKeyUp (int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_DPAD_LEFT) {
        currentValue--;
        sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_TEXT_CHANGED);
    }
    return true;
}
...
}
```

### C- @Override

```
public boolean onKeyUp (int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_ENTER) {
        currentValue--;
        sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_CONTEXT_CLICKED);
    }
}
```

```
return true;  
}  
...  
}
```

**Answer:**

---

B

**Explanation:**

---

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

## Question 11

---

**Question Type:** MultipleChoice

---

"workManager" is an instance of WorkManager. Select correct demonstration of WorkRequest cancellation:

**Options:**

---

**A-** `workManager.enqueue(new OneTimeWorkRequest.Builder(FooWorker.class).build());`

**B-** `WorkRequest request = new OneTimeWorkRequest.Builder(FooWorker.class).build(); workManager.enqueue(request);`  
`LiveData<WorkInfo> status = workManager.getWorkInfoByIdLiveData(request.getId ());`  
`status.observe(...);`

**C-** `WorkRequest request = new OneTimeWorkRequest.Builder(FooWorker.class).build(); workManager.enqueue(request);`  
`workManager.cancelWorkById(request.getId());`

**D-** `WorkRequest request1 = new OneTimeWorkRequest.Builder(FooWorker.class).build();`  
`WorkRequest request2 = new OneTimeWorkRequest.Builder(BarWorker.class).build`  
`();`  
`WorkRequest request3 = new OneTimeWorkRequest.Builder(BazWorker.class).build`  
`();`  
`workManager.beginWith(request1, request2).then(request3).enqueue();`

**E-** `WorkRequest request = new OneTimeWorkRequest.Builder(FooWorker.class).build(); workManager.enqueue(request);`  
`workManager.cancelWork(request);`

### **Answer:**

---

C

### **Explanation:**

---

Working with WorkManager, from the 2018 Android Dev Summit WorkManager: Beyond the basics, from the 2019 Android Dev Summit



<https://developer.android.com/reference/androidx/work/WorkManager?hl=en>

**To Get Premium Files for Associate-Android-Developer Visit**

**<https://www.p2pexams.com/products/associate-android-developer>**

**For More Free Questions Visit**

**<https://www.p2pexams.com/google/pdf/associate-android-developer>**

