

## Answer these questions

### Question 1

Select all that are true for notification channels:

- You use notification channels to display notifications to the user in the device status bar.
- You use notification channels to group multiple notifications so that the user can control the notifications' behavior.
- Notification channels are available in older devices, those running Android 7.0 Nougat (API 24) and lower.
- Notification channels are not yet available in the Android Support Library package.

Ans :

- You use notification channels to display notifications to the user in the device status bar.
- Notification channels are not yet available in the Android Support Library package.

### Question 2

Which API do you use to show a notification in the notification drawer and in the device's status bar?

- `Notification.notify()`
- `NotificationManager.notify()`
- `NotificationCompat.notify()`
- `NotificationCompat.Builder.notify()`

Ans: `NotificationCompat.Builder.notify()`

### Question 3

Which component is *not* needed when you add a notification action?

- Icon that represents the action
- Title that describes the action
- Click listener for the action button click event.
- `PendingIntent` that's sent when the user taps the action button.

Ans: Click listener for the action button click event.

#### Question 4

Which API do you use to add an action button to a notification?

- `NotificationCompat.addActionButton()`
- `NotificationCompat.Builder.addAction()`
- `Notification.Builder.addActionButton()`
- `NotificationManager.addAction()`

Ans : `NotificationCompat.Builder.addAction()`

#### Question 5

Suppose that you create an app that downloads a work of art on the user's device every day. Once the day's image is available, the app shows a notification to the user, and the user can download the image or skip the download. What `PendingIntent` method would you use to start a service to download the image?

- `Activity.startService()`
- `PendingIntent.getBroadcast()`
- `PendingIntent.getActivity()`
- `PendingIntent.getService()`

Ans: `PendingIntent.getBroadcast()`