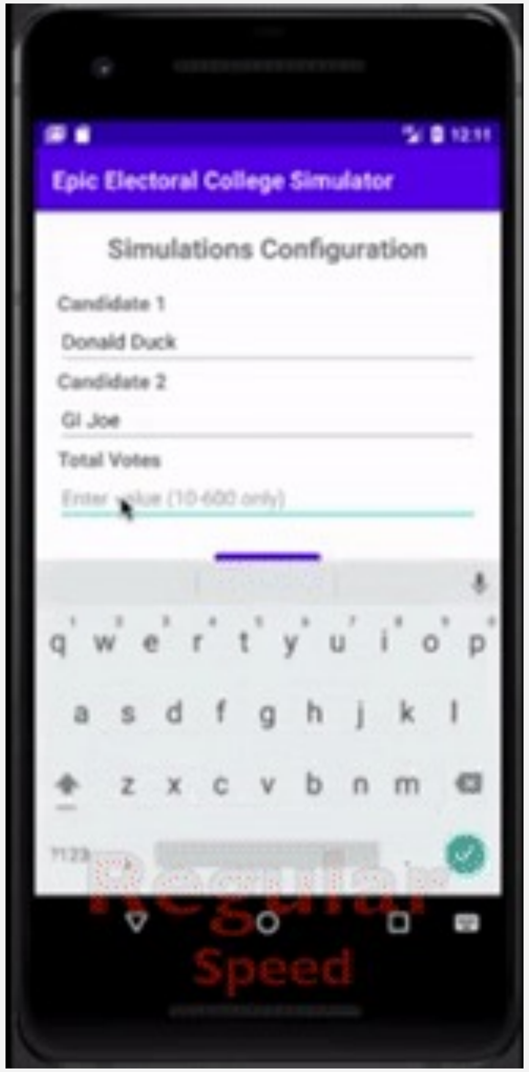
MOBILE DEVELOPMENT

Activity Lifecycle

Ever notice the **transition** between activities?

This transition includes both the stopping of one activity and the starting of another activity


Epic Electoral College Simulator

Simulations Configuration

Candidate 1

Donald Duck

Candidate 2

GI Joe

Total Votes

Enter value (10-600 only)

In this video, we'll learn more about these transitions by discussing the **Activity Lifecycle**

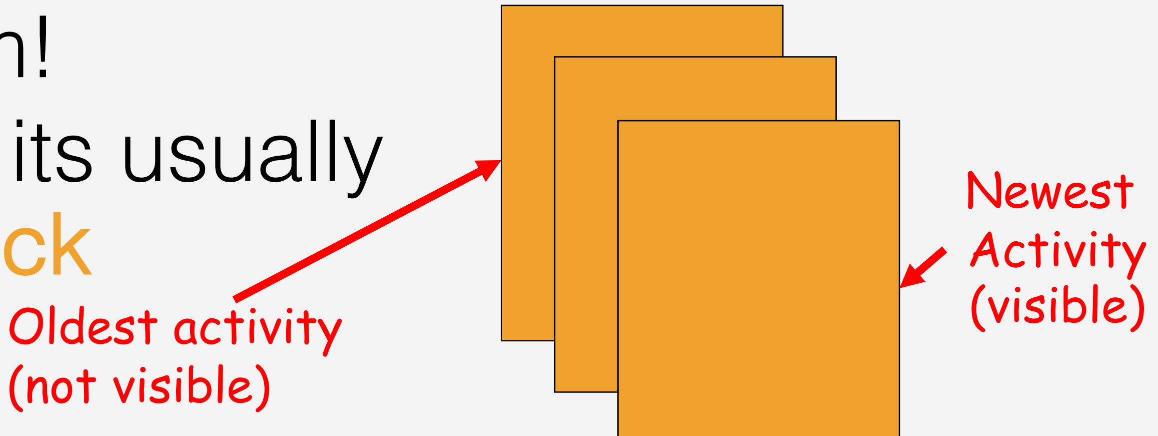
Motivation

- How does an activity start? How does it end?
- What happens to an activity when transitioning to another activity / app?

```
public class MainActivity extends AppC

    @Override
    protected void onCreate(Bundle sav
        super.onCreate(savedInstanceSt
        setContentView(R.layout.activi
    }

    @Override
    protected void onStart() {
        super.onStart();
    }

    @Override
    protected void onResume() {
        super.onResume();
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onStop() {
```

Activity Lifecycle

- **Recall**:
  - We did not instantiate an activity or specify a `main()` function.
  - We placed our code in the `onCreate()` method and started activities using the appropriate method calls.
- **Question**: So how are activities created / manages?
  - Its all with the operating system!
  - When a new activity is started, its usually placed on top of an activity stack



Activity Lifecycle

- There are a series of events that occur throughout the life of an activity
  - When the activity is created…
  - When the activity is destroyed…
  - When the activity is started…?
  - When the activity is paused…?
  - When the activity is resumed…?

What do these mean?

Activity Lifecycle

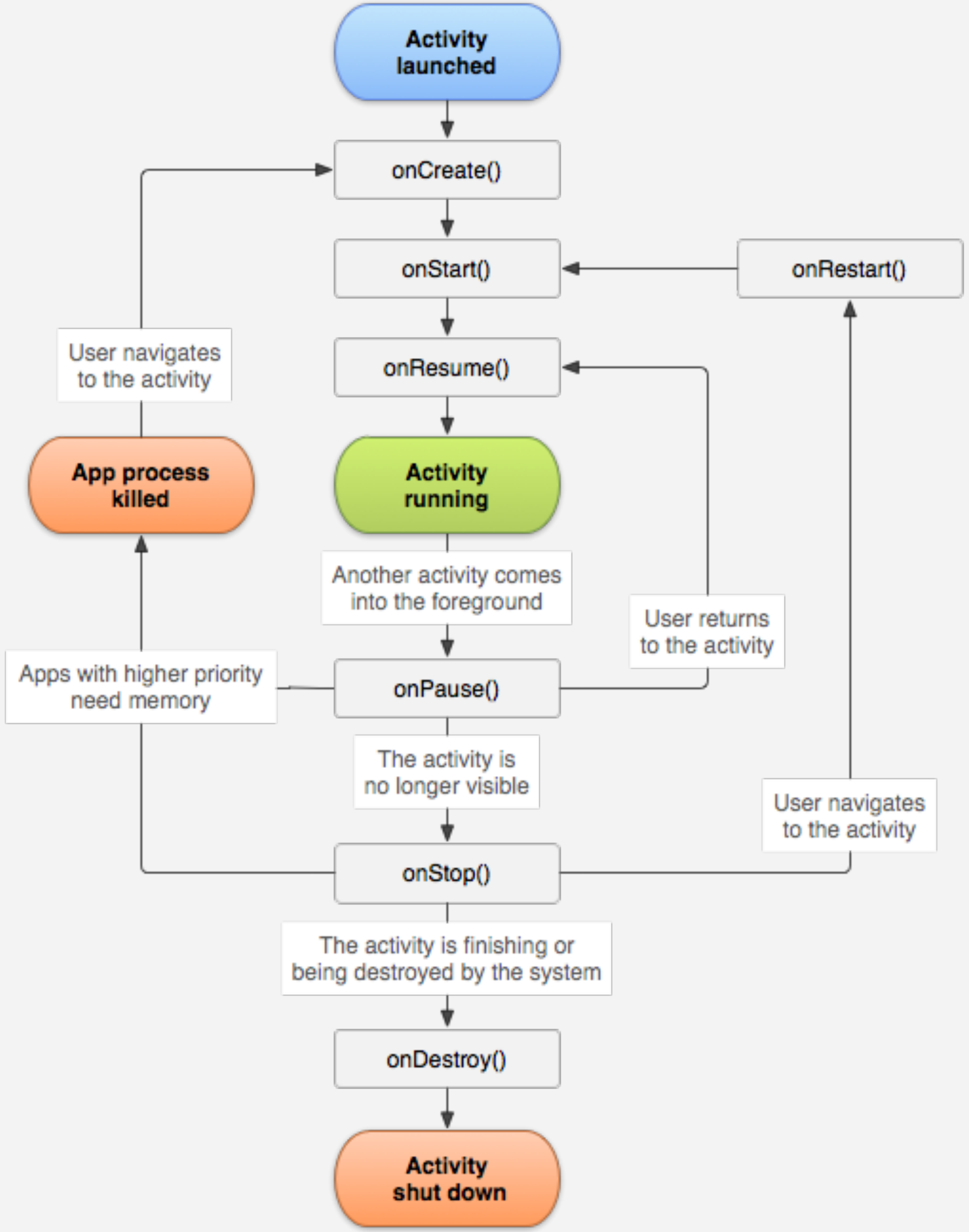
- In each event, Android executes a callback method
  - We can override these methods for them to behave the way we want them to behave
- For example, when a user exits a video streaming app, you might want to have the app pause the video and stop streaming instead of having the video continue to run in the background

Additionally, think of (or literally try out and see)
how YouTube and Spotify apps differ

Activity Lifecycle

- The figure to the side is known as the Activity Lifecycle
- Apart from **`onRestart()`**, the lifecycle consists of 6 core callback methods

```
onCreate()     onPause()
onStart()      onStop()
onResume()     onDestroy()
```



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- `onCreate()`
  - Is called when the activity is first created / instantiated
    - Only called once
  - Is where UI elements and other objects that persist throughout the activity's life are initialized
  - Think of this method as things you'd do in a class' constructor

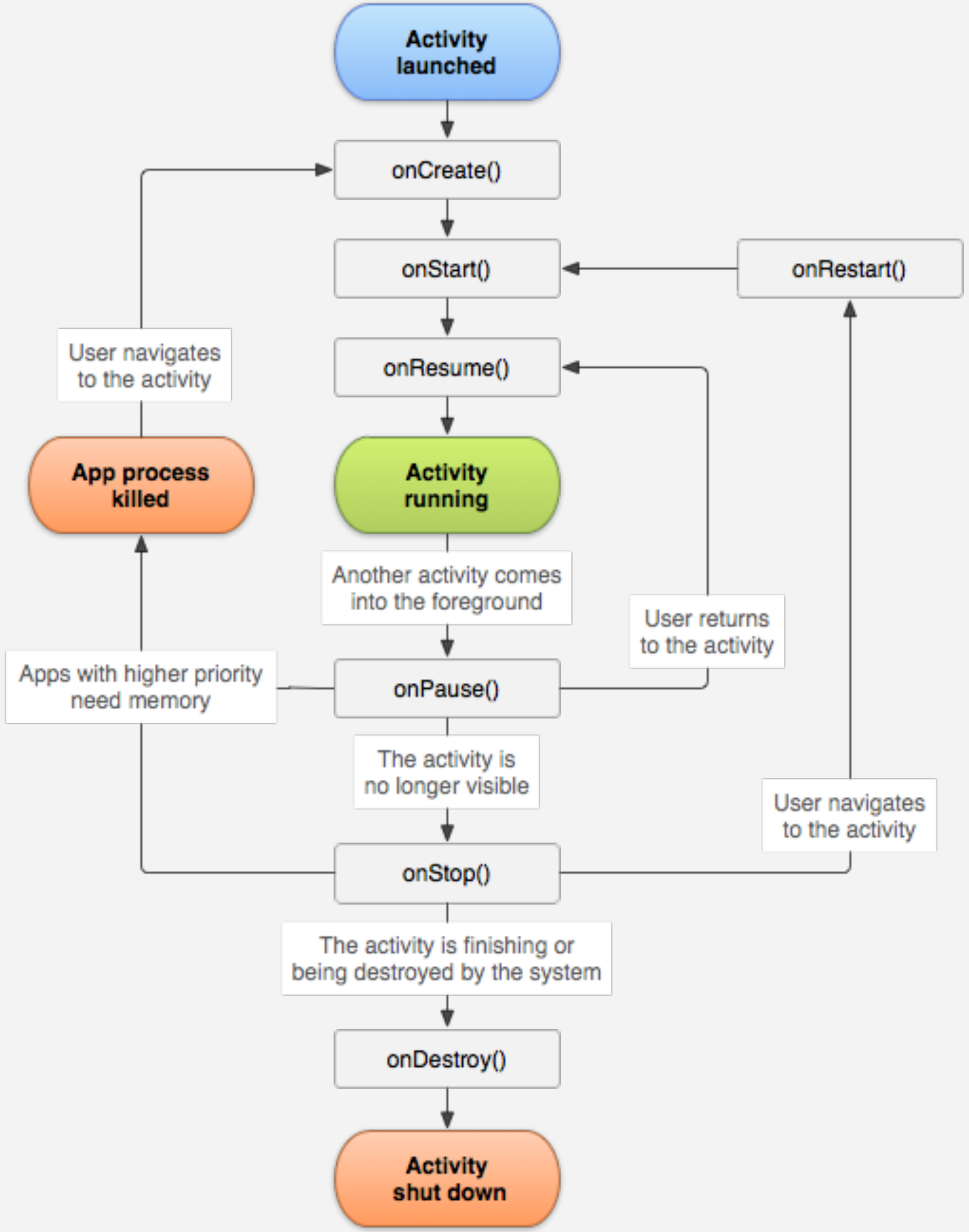

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- `onStart()`
  - Is called just before the activity becomes visible to the user
    - It makes the activity visible and prepares the activity to enter the foreground
  - Maybe called more than once [versus `onCreate()`]
    - See the flow in the diagram

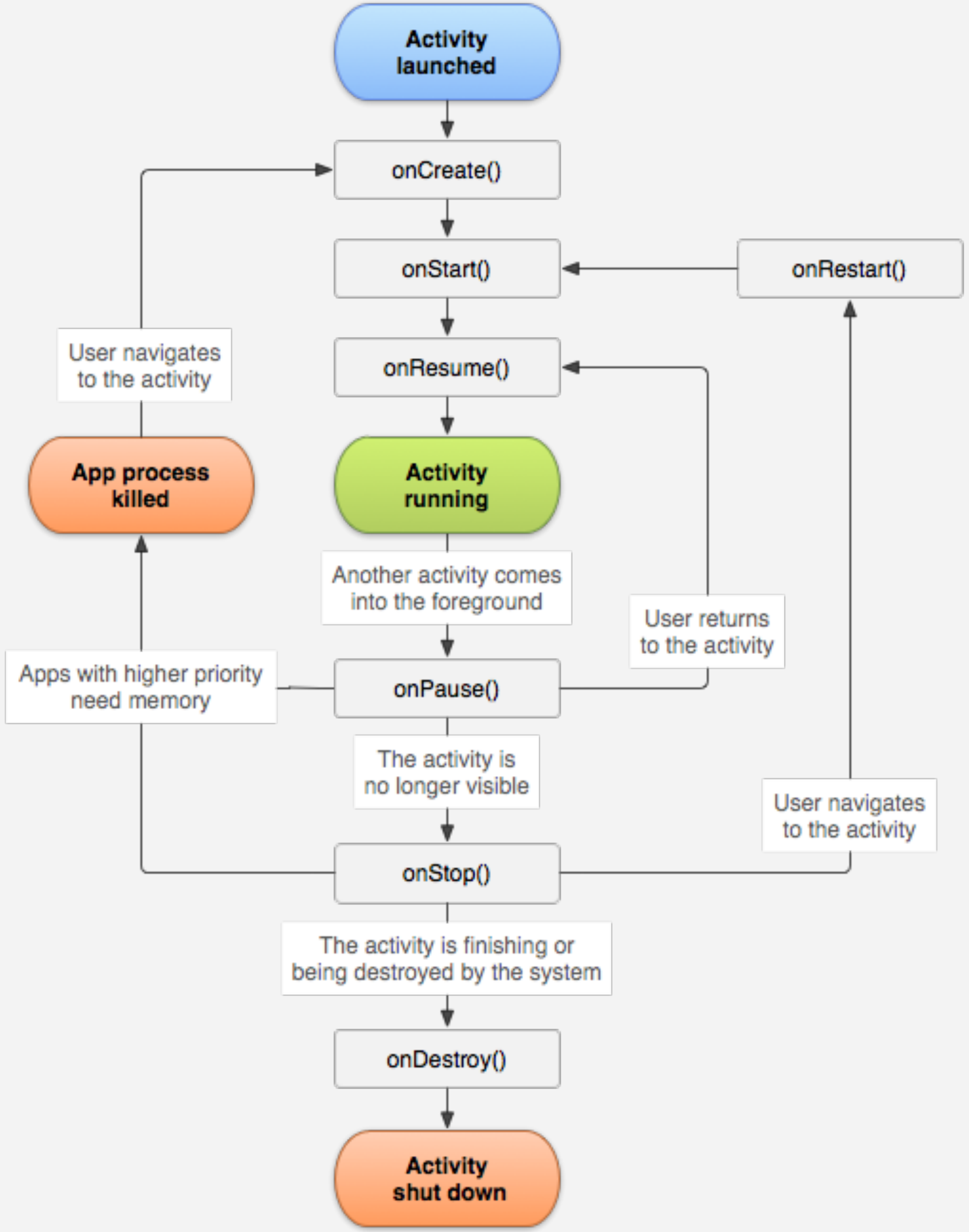
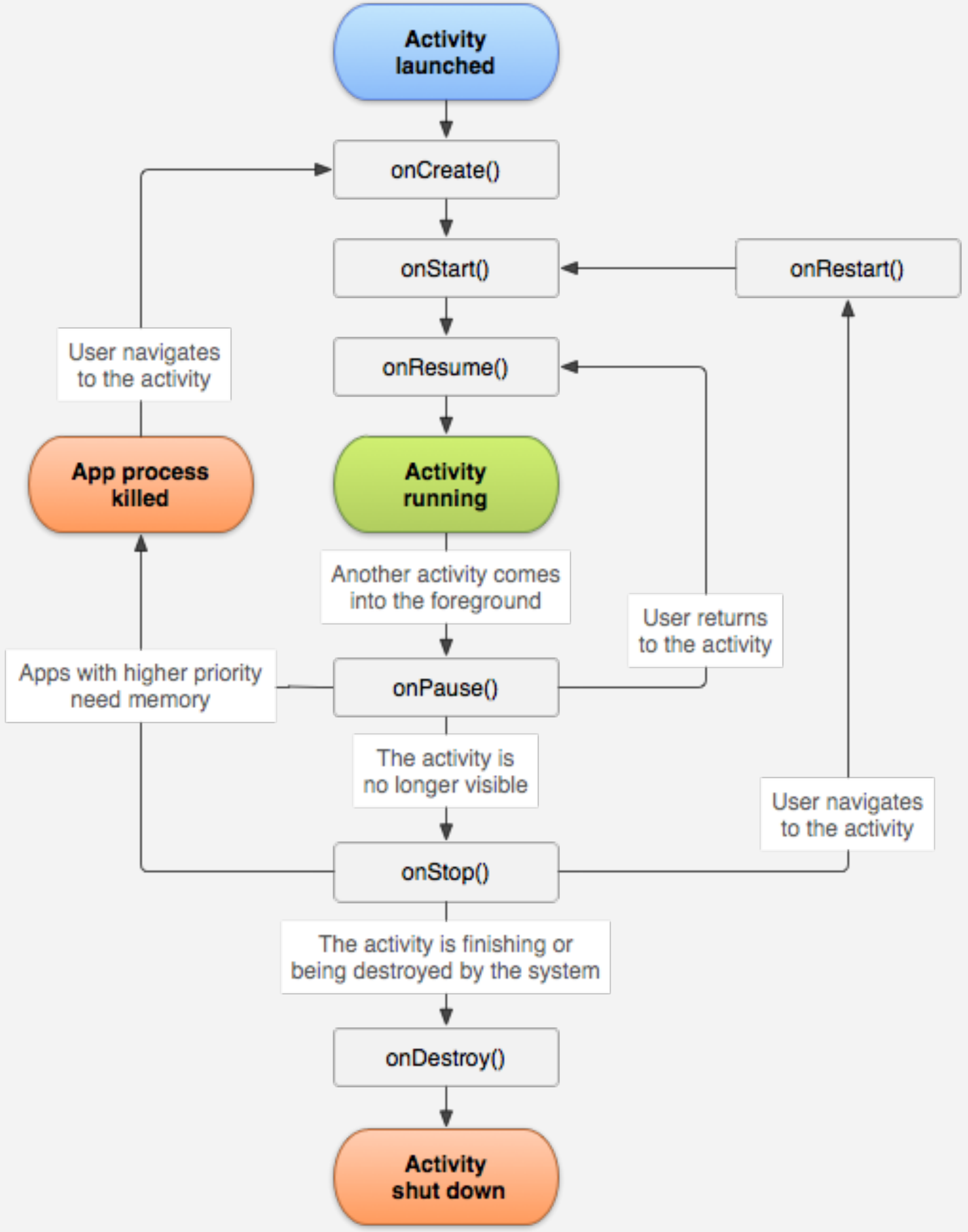

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- **`onResume()`**
  - Is called just before the user interaction starts and signals that the UI is ready for interaction
  - Think of this as when the activity gains focus



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- **onPause()**
  - Is called when the system is just about to call another activity or when the current activity is about to lose focus

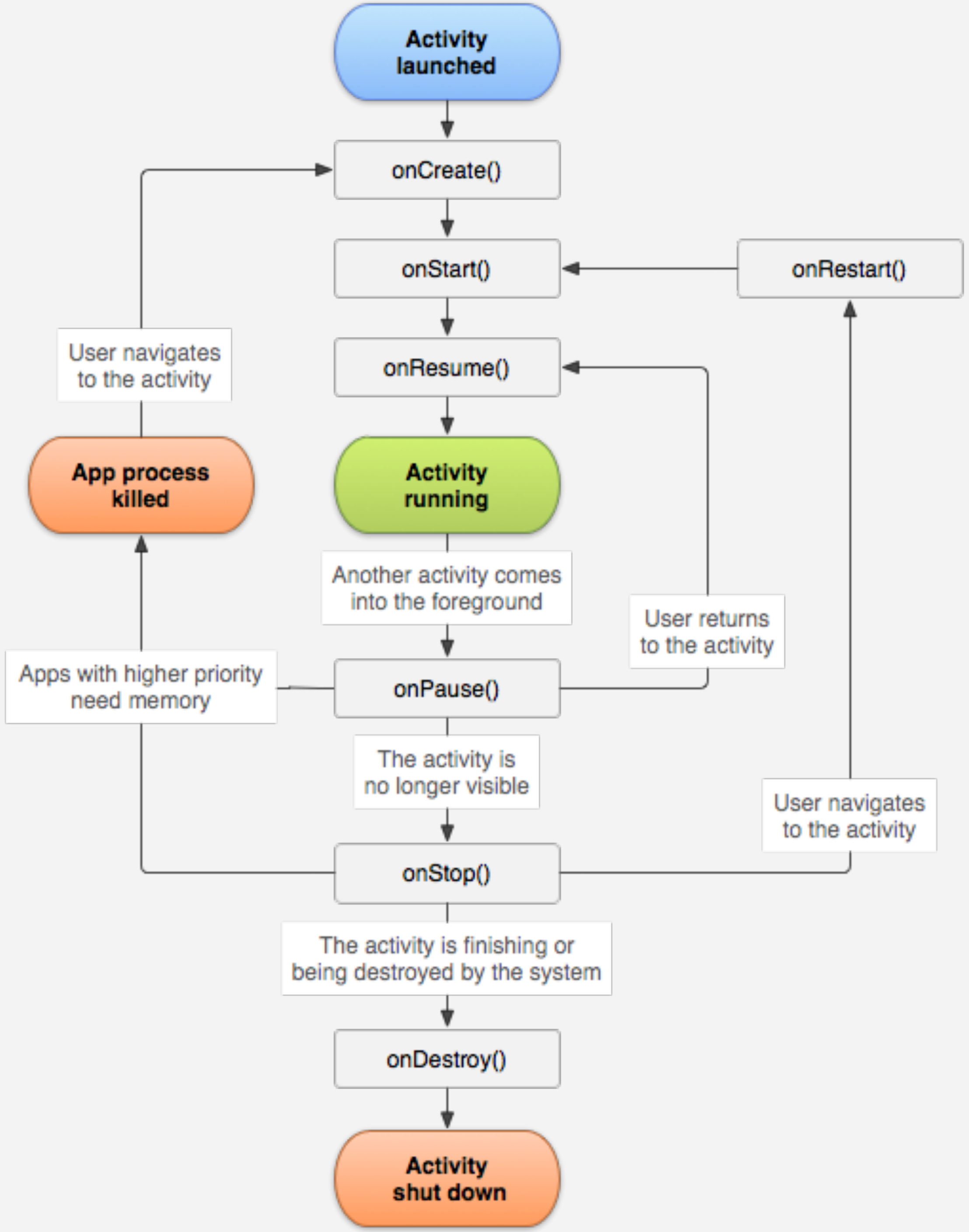
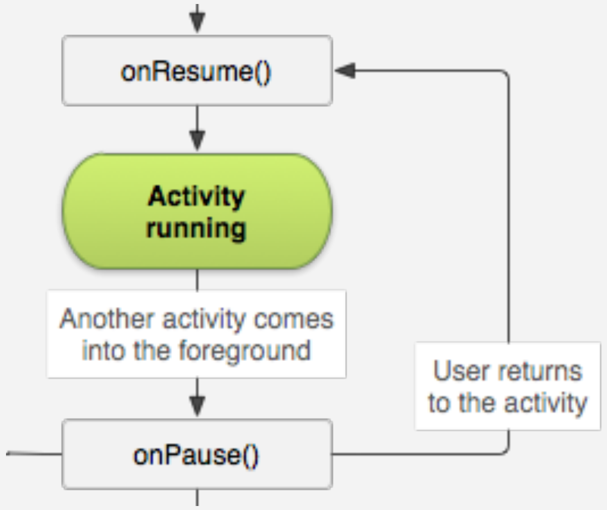
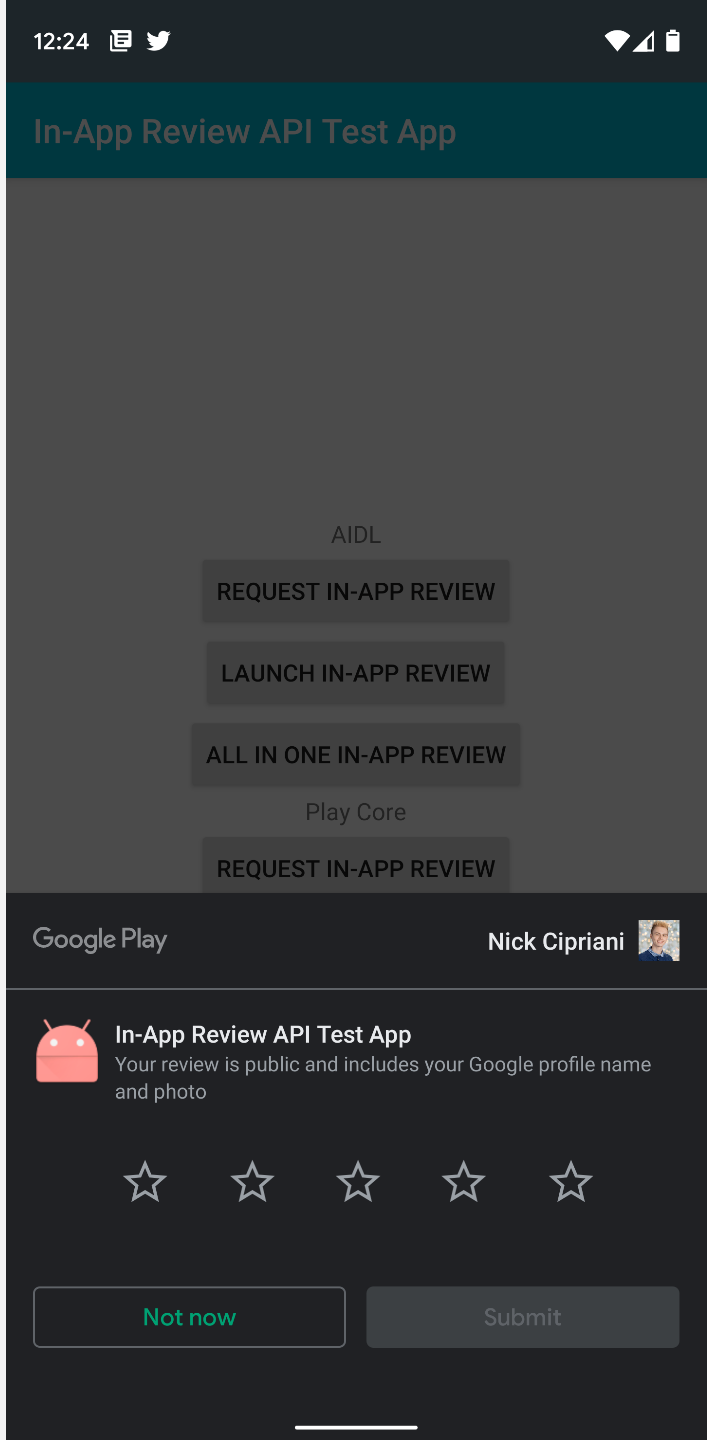

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- To understand focus, let's take the app to the side as an example
  - When a dialog box comes up, the app in the background loses focus
    - `onPause()`; App is greyed out
  - When the dialog box is removed, the app regains focus
    - `onResume()`


onResume()
Activity running
Another activity comes into the foreground
onPause()
User returns to the activity


12:24

In-App Review API Test App

AIDL

REQUEST IN-APP REVIEW

LAUNCH IN-APP REVIEW

ALL IN ONE IN-APP REVIEW

Play Core

REQUEST IN-APP REVIEW

Google Play

Nick Cipriani

In-App Review API Test App

Your review is public and includes your Google profile name and photo

☆ ☆ ☆ ☆ ☆

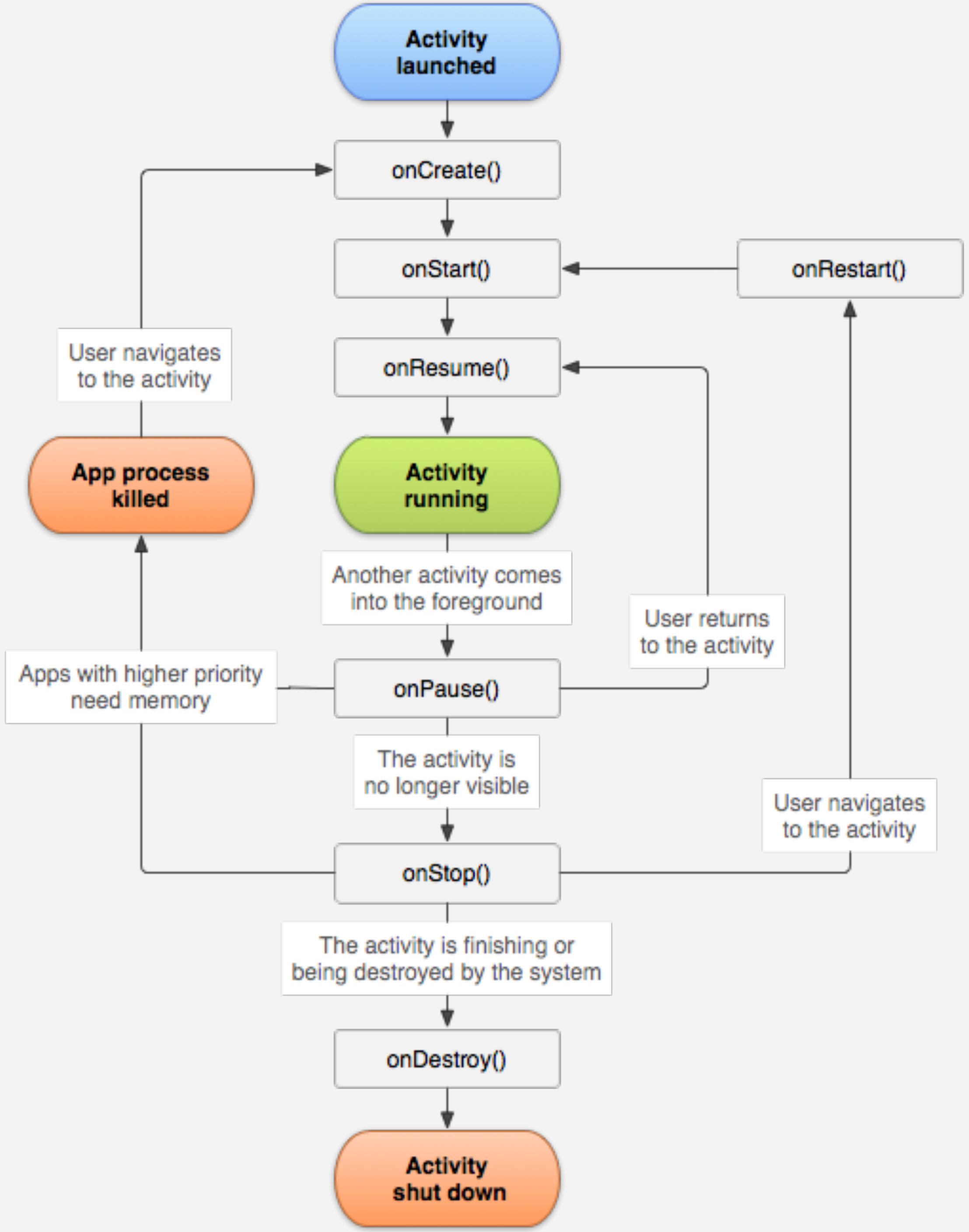
Not now

Submit

Image source: https://www.androidpolice.com/2020/01/11/play-store-tests-include-new-in-app-review-dialogs-and-a-dedicated-reviews-section/

Activity Lifecycle

- **`onPause()`**
  - Is called when the system is just about to call another activity or when the current activity is about to lose focus
    - During **`onPause()`**, the activity is still visible
  - Typically used to store minor unsaved changes or stop playbacks



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- **`onStop()`**
  - Is called directly after the activity lose focus / is no longer visible
  - Compared to **`onPause()`**, this is where more complex saving of data should occur

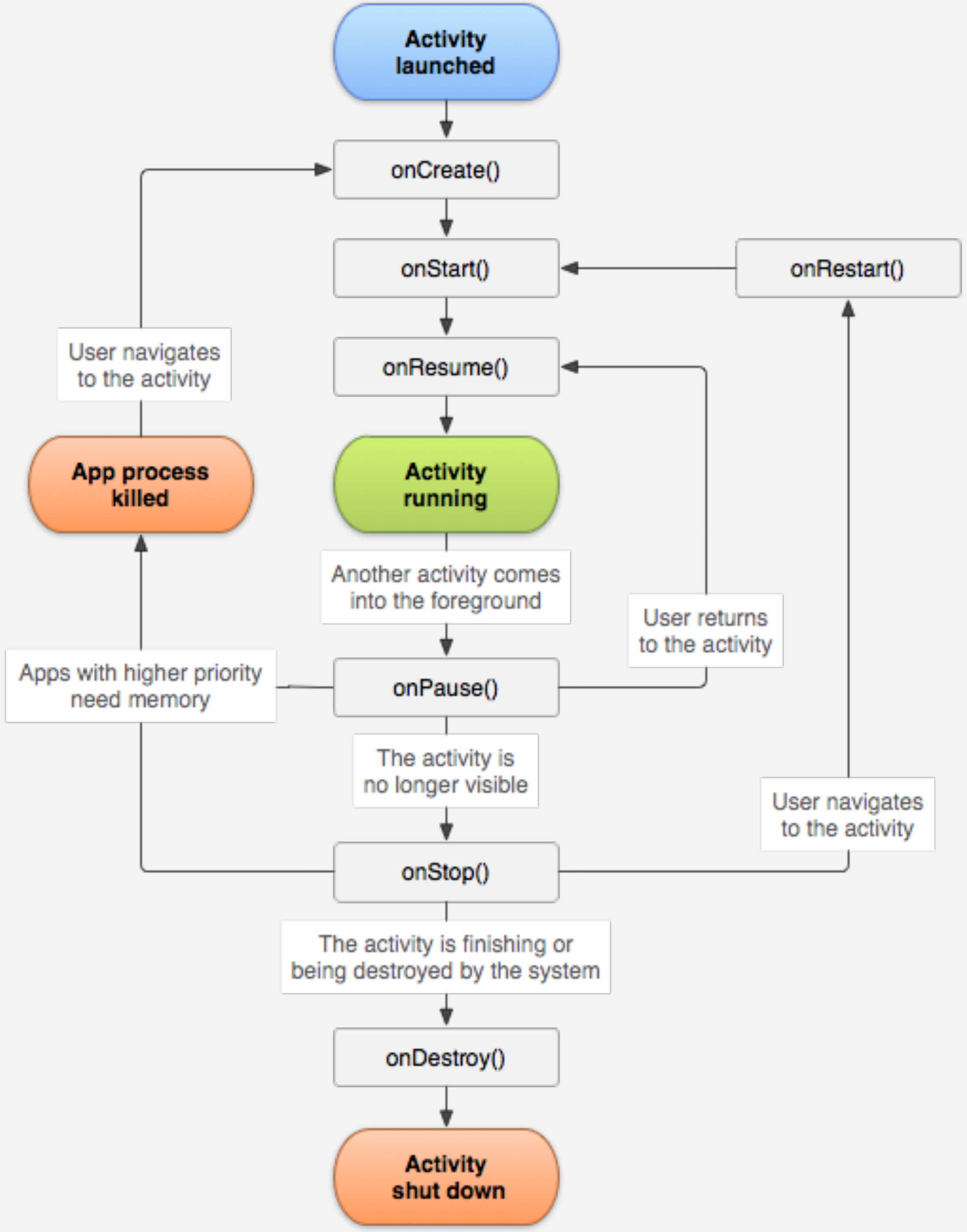
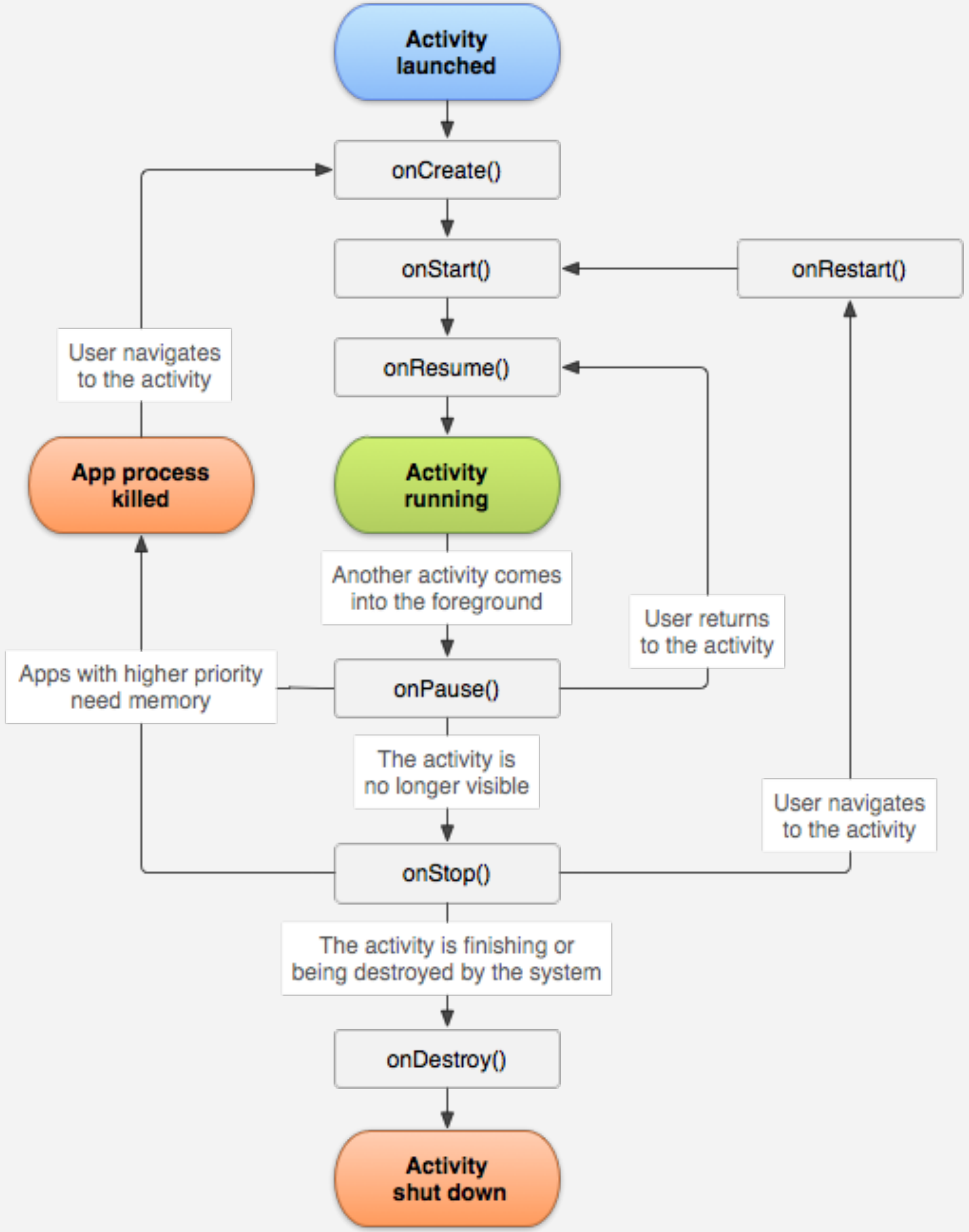

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- **`onRestart()`**
  - Is called when the activity is coming back from being previously stopped
  - This event allows you to run coded when the app is being restarted rather than being created for the first time



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- **onDestory()**
  - Is called when the activity is about to close
  - This can happen when…
    - An activity is "finished"
    - The application was manually "ended" by the user
    - The OS "kills" the activity in a need to free up memory

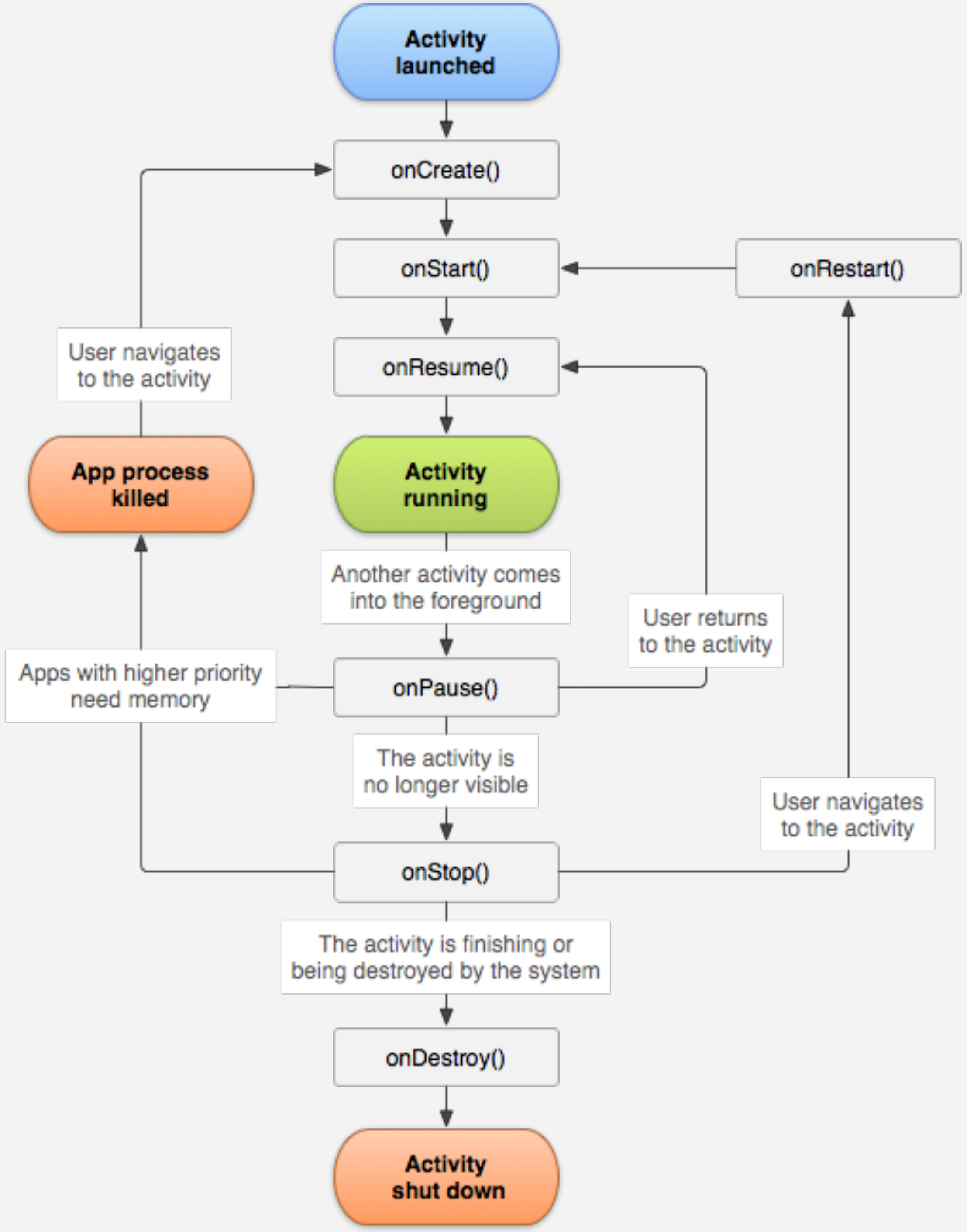
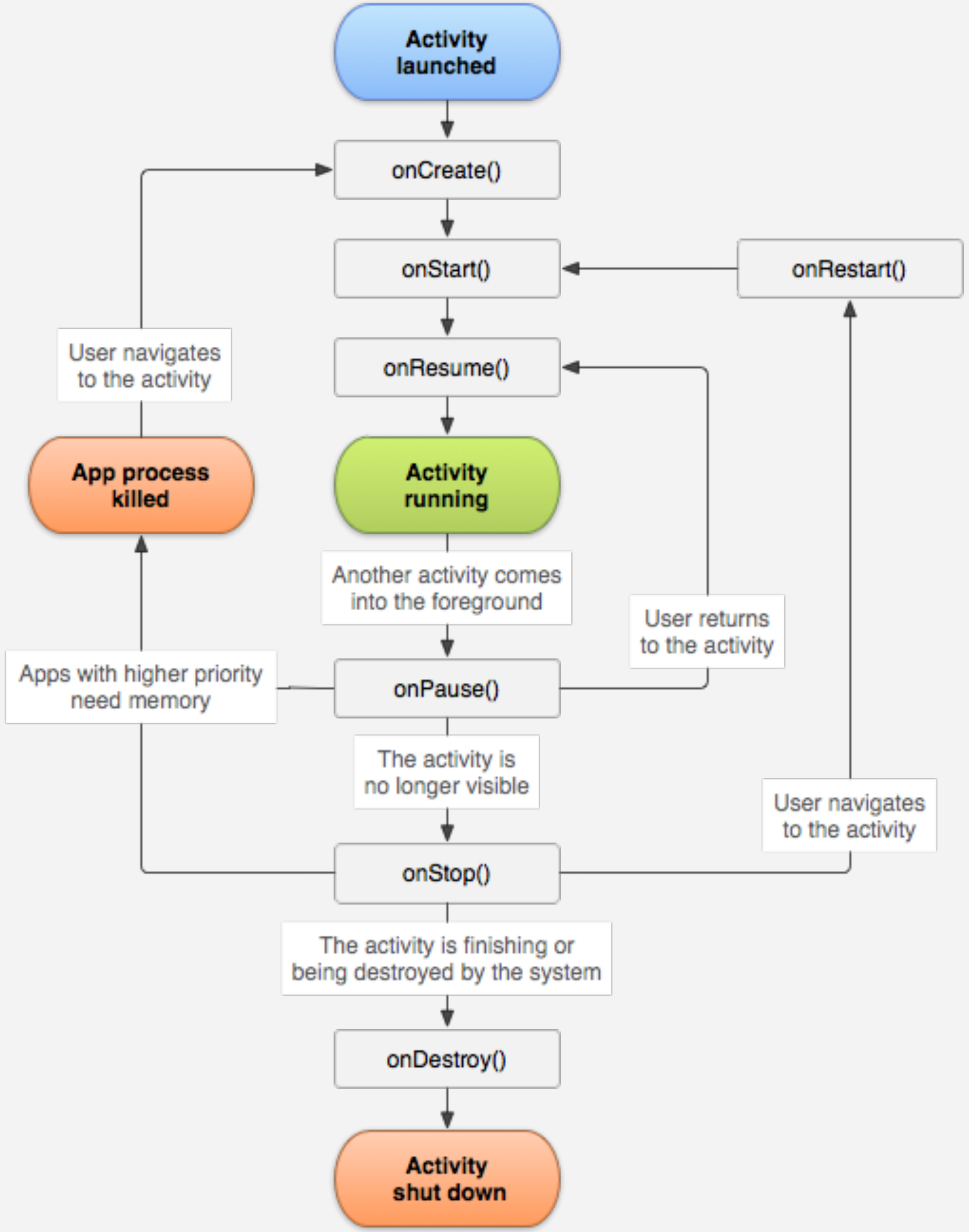

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- **onDestory()**
  - The method handles freeing up resources that the app is responsible for
    - Processes (e.g. threads, services)
    - Connections
  - If there is no need explicitly free up resources, there is no need overwrite the method



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: https://developer.android.com/reference/android/app/Activity

Activity Lifecycle

- If the OS decides your app is no longer necessary, it will kill your application
- This destruction is somewhat unpredictable and is dependent on resource allocation
  - Hence, implement your activities as if they could be destroyed at any moment

More on this in the asynch+optional material for memory management ☺

Activity Lifecycle

- `onCreate()` → initialization
- `onStart()` → right before app is visible
- `onResume()` → app is visible / has focus
- `onPause()` → app is still visible / lost focus
- `onStop()` → app is no longer visible
- `onRestart()` → when app is restarting
- `onDestroy()` → when app is about to end

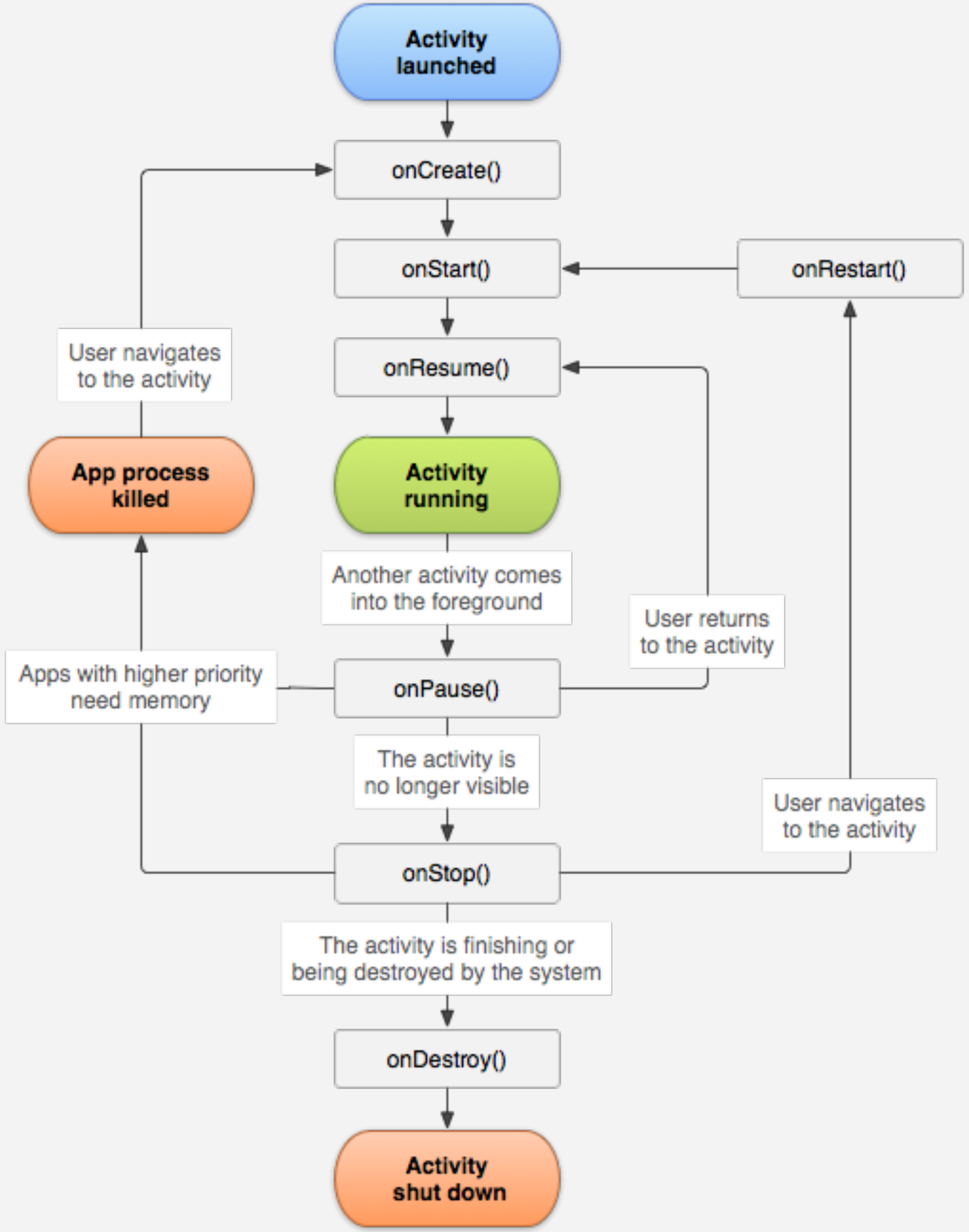

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Image sauce: *https://developer.android.com/reference/android/app/Activity*

Activity Lifecycle

- Not all callback methods need to be implemented
  - Recall that for our first few lessons we’ve only been dealing with `onCreate()` – for initialization
- These callback methods help in modularization and can reduce redundancy in your code

Activity Lifecycle – Summary

- An Activity moves through different phases depending on the current situation
  - When it is created, about to be shown, about to be hidden, or to be destroyed
- We can add behavior / code to these phases by overriding the appropriate methods
  - `onCreate(), onStart(), onResume(), onPause(), onStop(), onDestroy()`

If there are any questions, check out the discussion page / question board in the same module you found this video ☺

Questions?

Experiment: Let’s create an app that…

- Contains 2 activities
  - One main activity that…
    - Logs every time a lifecycle callback method is called
    - Has a button that starts another activity (the 2nd activity)
  - Second activity that…
    - Logs every time a lifecycle callback method is called
- Then, let’s experiment with different scenarios that trigger the different callback methods

Let’s pull up Android Studio!

Before we end…

- Please note that there is also an Android project in the module for Activity Lifecycle
- It is similar to the app we built in today's session, but it also counts the logs and displays the counts on screen
- Please consider checking it out in your free time

Thanks everyone!

Next session, we'll talk about the Activity State and SharedPreferences

Image sauce: https://www.dejaysblog.com/2020/05/our-memes-of-week-18.html