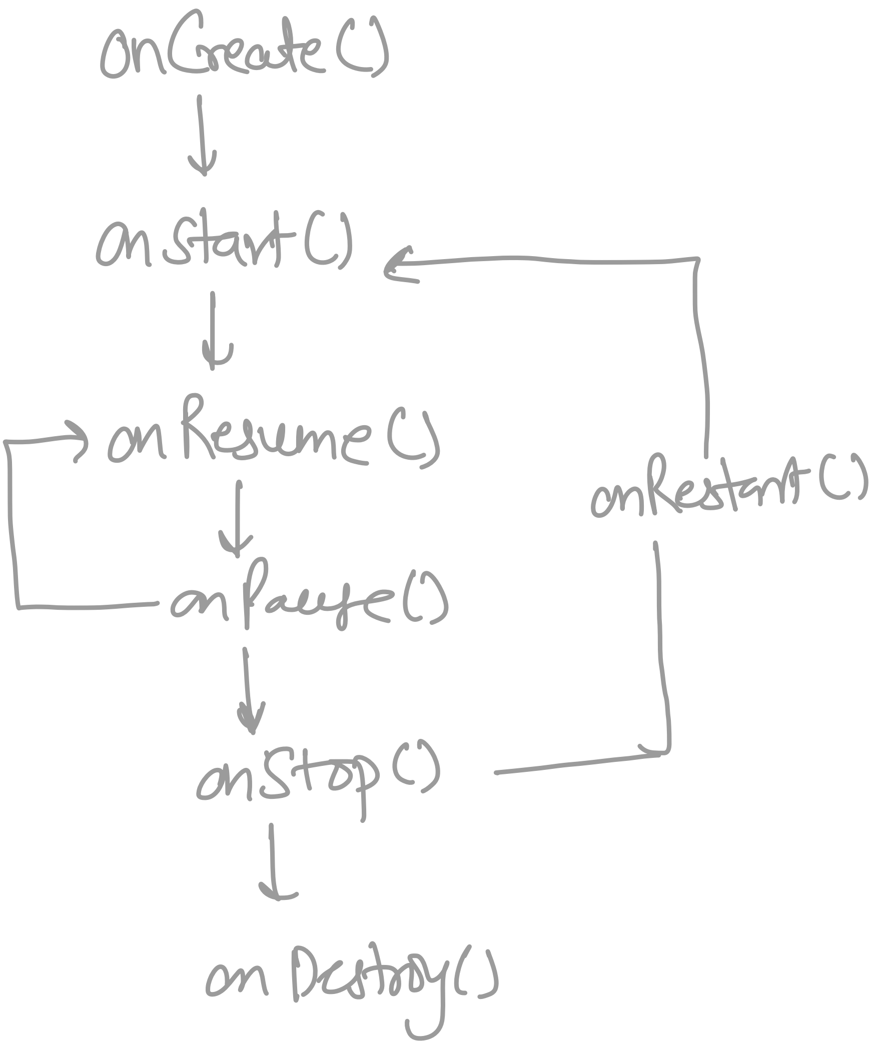
Android Interview Guide

Chapter 2
Android core concepts

notification
Garbage collector

⇒ Activity Life Cycle



onCreate() → Activity is Created

onStart() → starts when screen is visible to user

onResume() → Here User is interacting with App Screen

onPause() → Called when app is partially visible. like dialog etc

onStop() → Called when activity is not visible to user

onDestroy() →
- Called just before android shuts down
- the activity.
- It clears the activity from the memory

onRestart() → • Called when app Comes back from background state to foreground
• Also on Configuration changes

Scenario based questions [mention the Sequence of lifecycle functions in following Cases]

① When Activity is launched after Clicked

onCreate()
onStart()
onResume()

Same as this

② Now App is pushed to background

onPause()
onStop()
onSaveInstan

this background means "home" ◁ ○ ▢ button is pressed

Nothing changes on pressing

③ Running Activity → foreground to Background

onCreate()
onStart()
onResume()

when Activity opened

onPause()
onStop()
onSaveInstance()

when "Home" btn pressed → Background

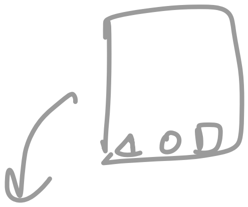
onRestart()
onStart()
onResume()

when App Comes back to foreground

④ Activity is already opened → Now Screen is rotated

onCreate()
onStart()
onResume()

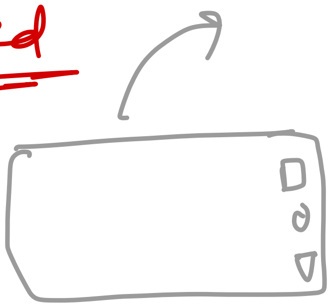
when Activity opened



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

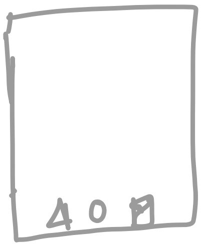
when App Screen is rotated

for Configuration changes Activity is destroyed & reCreated



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

Again this is Called.



5 Activity A → Activity B → Activity A

Activity A
- onCreate()
- onStart()
- onResume()

Activity A ≡ onPause() - - - → User click to switch to Activity B

- - - → [Activity B starts]

Activity B
- onCreate()
- onStart()
- onResume()

Activity A
- onStop()
- onSave Instance

- - - → User goes Back to Activity A

Activity B → onPause()

Activity A
- onCreate()
- onStart()
- onResume()

Again Activity A visible

Activity B
- onstop()
- onSave Instance()

In Background Activity B stops()

(6) Activity → goes to Background → User Kills the Apps

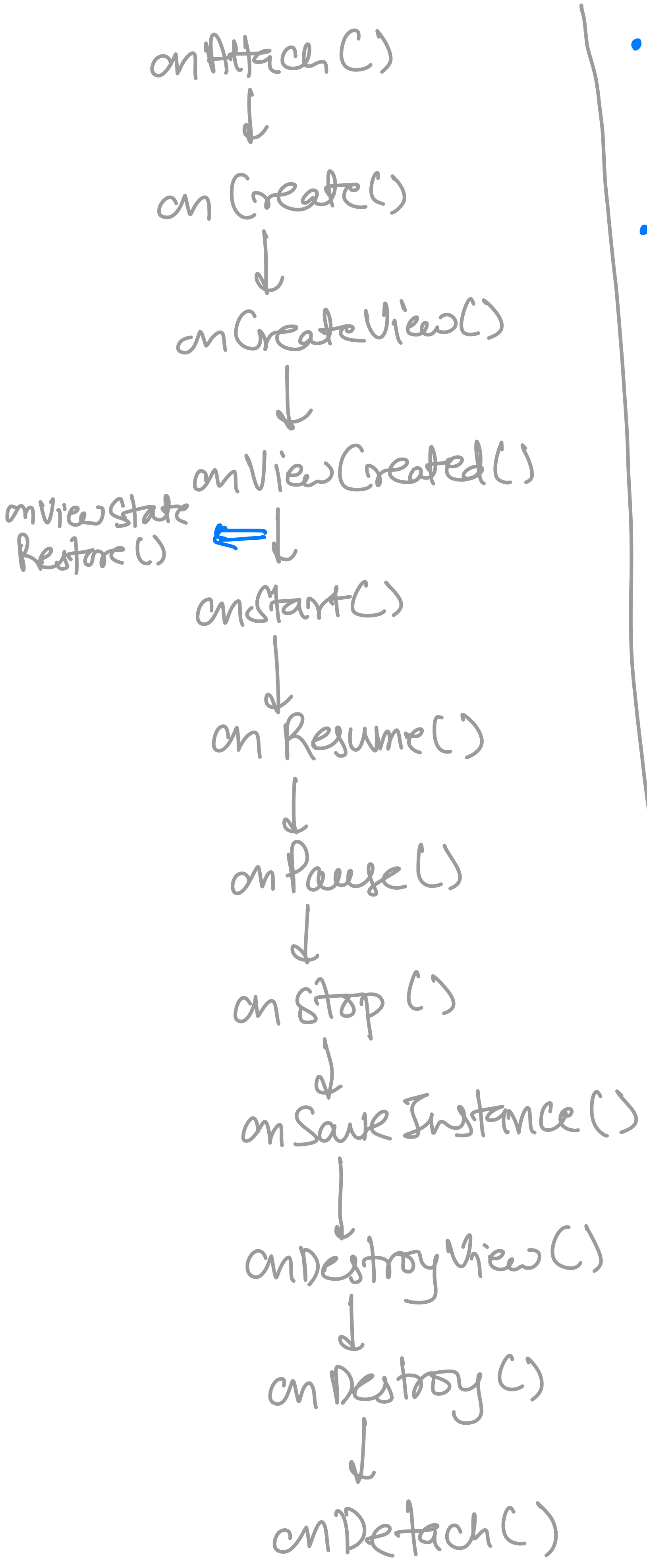
onCreate()
onStart()
onResume()

Activity Visible

here → User Kills the app

onPause()
onStop()
on SaveInstance()
onDestroy()

⇒ FRAGMENT LIFECYCLE



- Each fragment has its own lifecycle.
- Each Possible "Lifecycle" state is represented with "Lifecycle.state" enum
  - INITIALIZED
  - CREATED
  - STARTED
  - RESUMED
  - DESTROYED

Ques :- onRestart() in fragment?

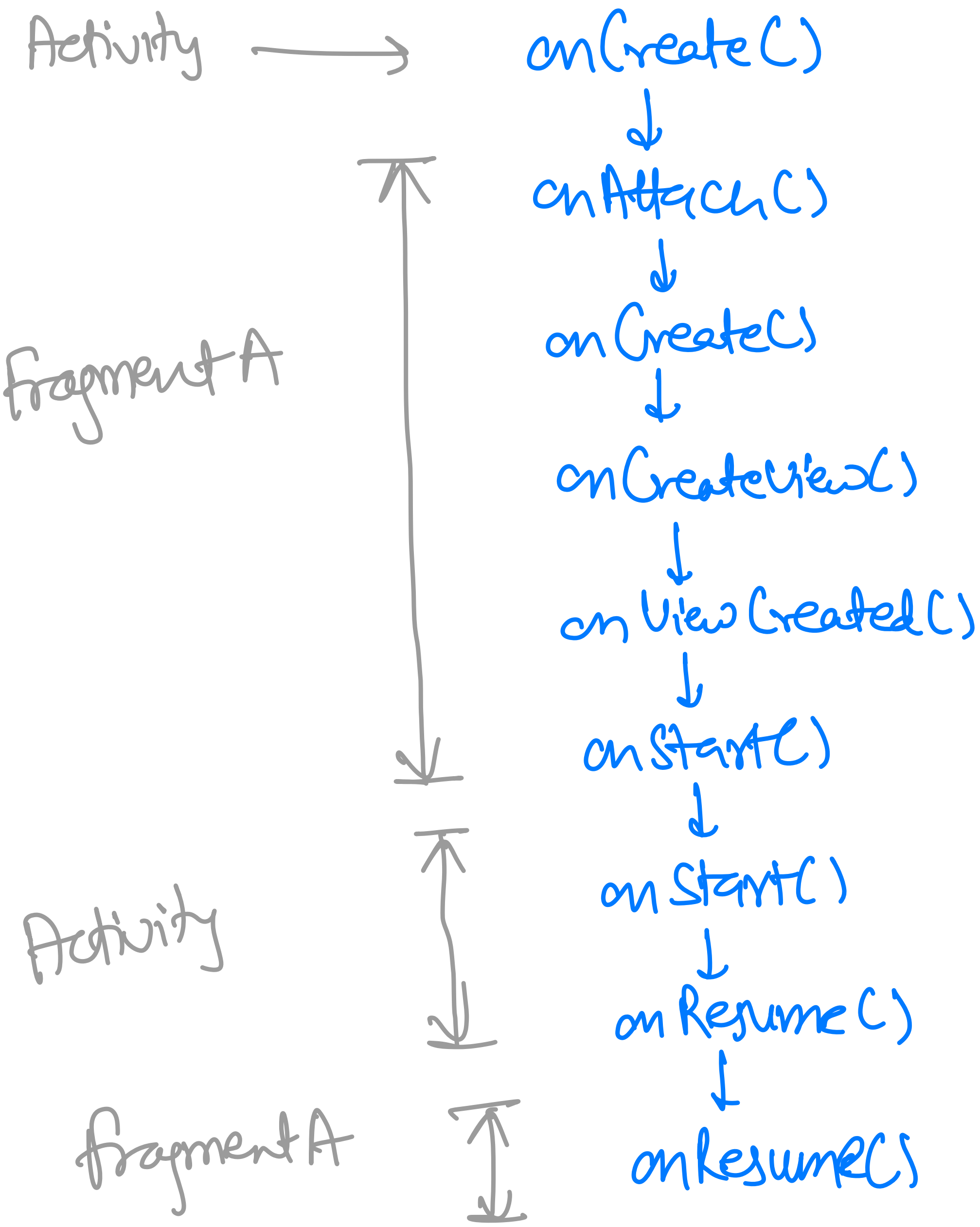
Ans :- fragment do not have onRestart() lifecycle method like Activity

↓

Reason → fragment rely on lifecycle of Activity and Restart() is handled from there.

Comman Scenerio ≡ Activity → fragment A (default)
→ fragment B

Ques ≡ Activity is opened, fragment A is default attached.



Activity → onCreate()

↓

Fragment A

onAttach()

↓

onCreate()

↓

onCreateView()

↓

onViewCreated()

↓

onStart()

↓

Activity

onStart()

↓

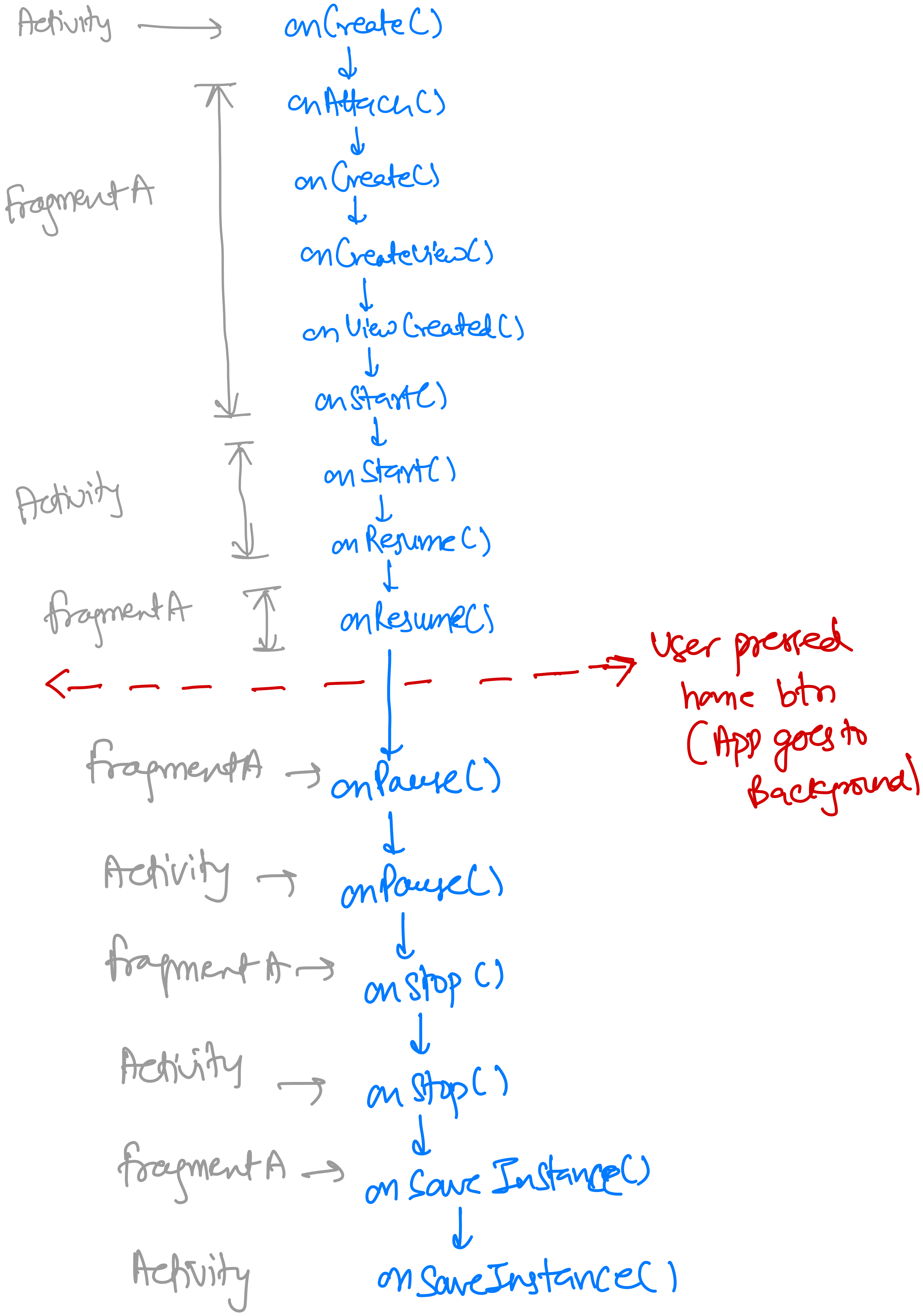
onResume()

↓

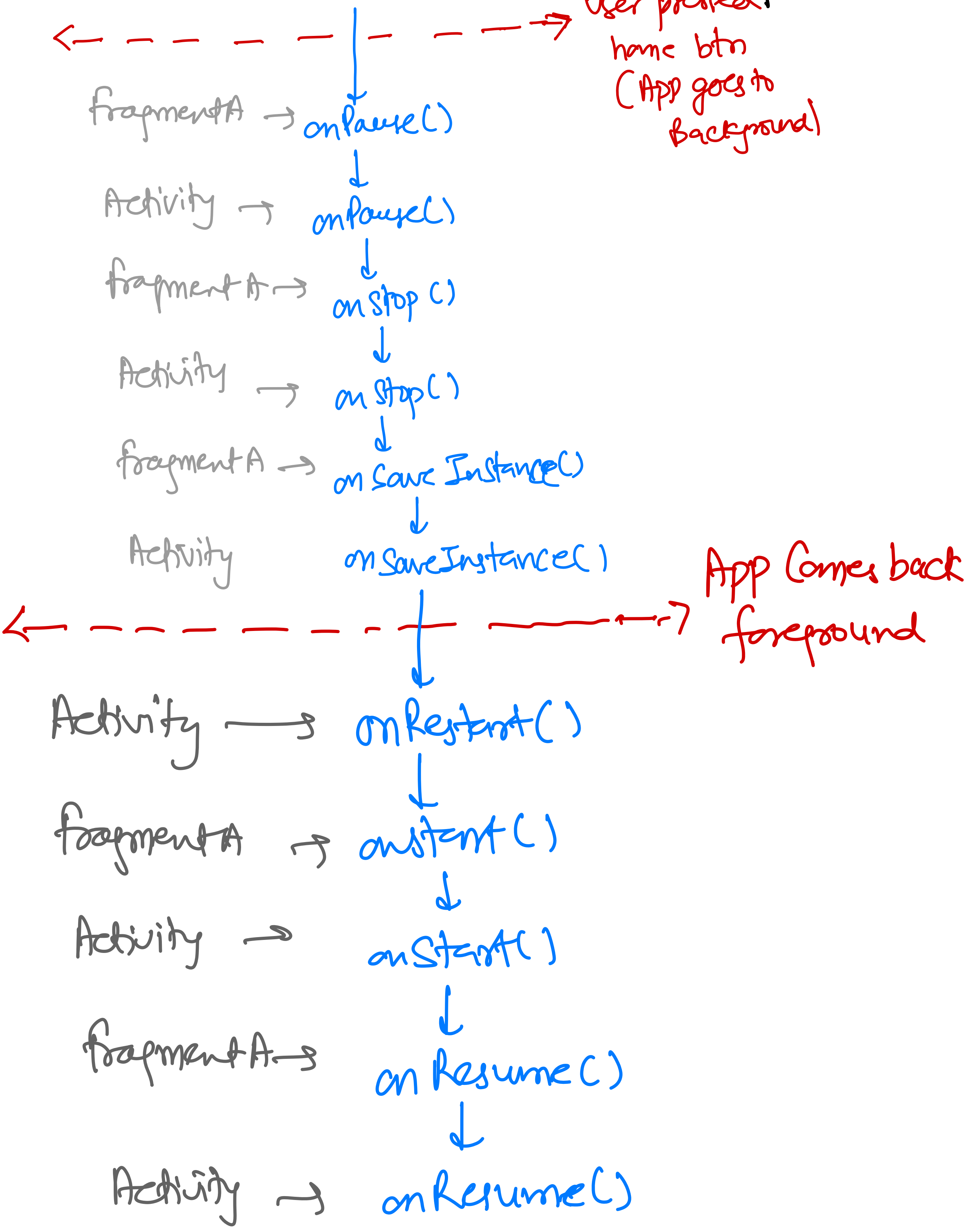
fragment A

onResume()

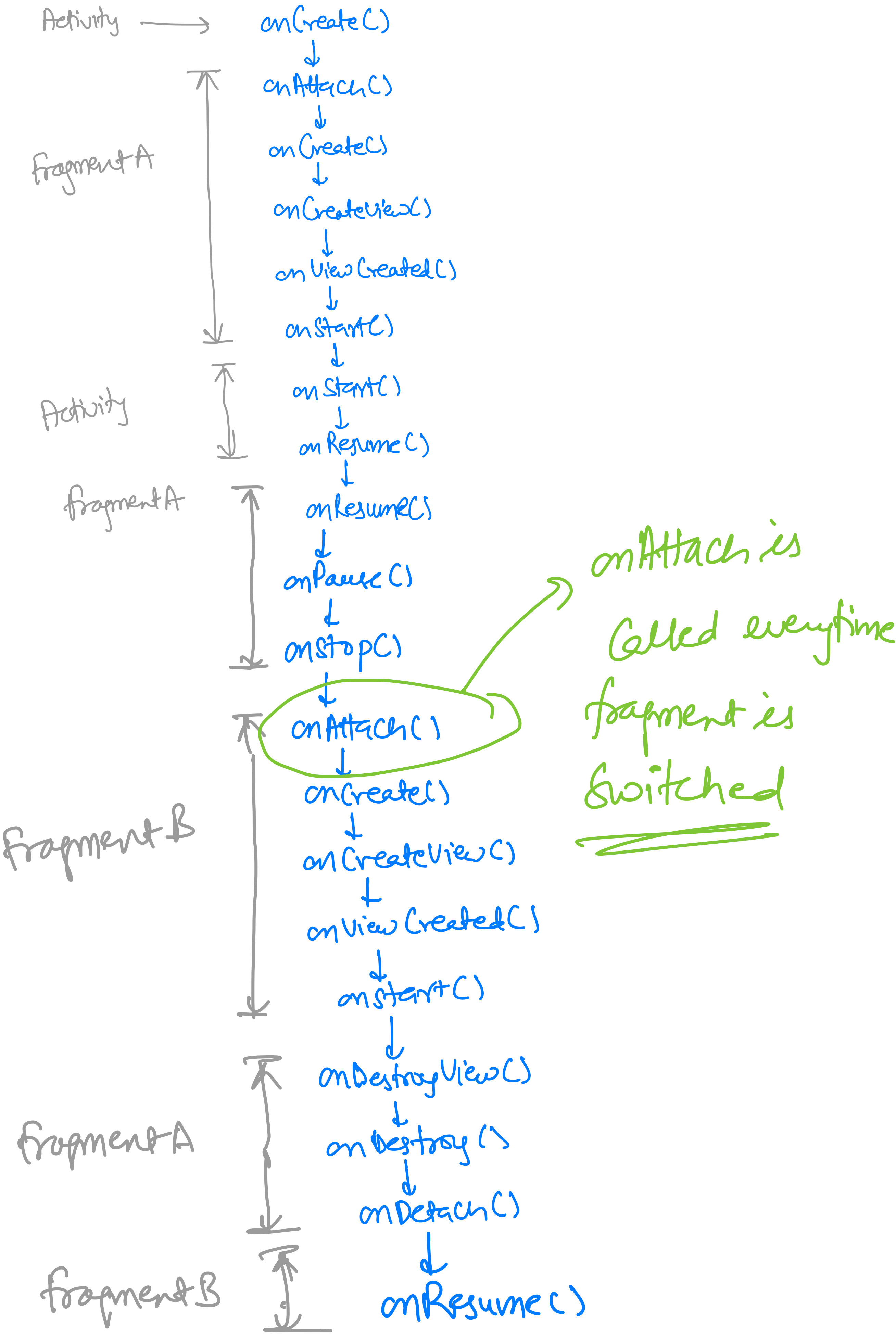
Ques: Now Activity with FragmentA → Background



Ques = App Comes back foreground [Activity + fragment A]



Ques= fragment A → fragment B



Activity → onCreate()
↓
Fragment A: onAttach()
↓
onCreate()
↓
onCreateView()
↓
onViewCreated()
↓
onStart()
↓
Activity: onStart()
↓
onResume()
↓
fragment A: onResume()
↓
onPause()
↓
onStop()
↓
fragment B: onAttach()
↓
onCreate()
↓
onCreateView()
↓
onViewCreated()
↓
onStart()
↓
fragment A: onDestroyView()
↓
onDestroy()
↓
onDetach()
↓
fragment B: onResume()

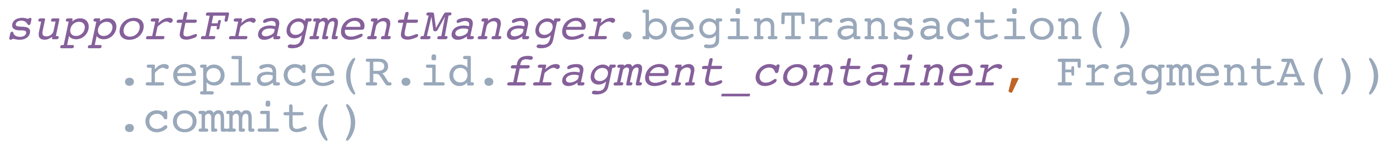
onAttach is called everytime fragment is switched

Methods of the Android Fragment

Methods	Description
onAttach()	The very first method to be called when the fragment has been associated with the activity. This method executes only once during the lifetime of a fragment. When we attach fragment(child) to Main(parent) activity then it call first and then not call this method any time(like you run an app and close and reopen) simple means that this method call only one time.
onCreate()	This method initializes the fragment by adding all the required attributes and components.
onCreateView()	System calls this method to create the user interface of the fragment. The root of the fragment’s layout is returned as the View component by this method to draw the UI. You should inflate your layout in onCreateView but shouldn’t initialize other views using findViewById in onCreateView.
onViewCreated()	It indicates that the activity has been created in which the fragment exists. View hierarchy of the fragment also instantiated before this function call.
onStart()	The system invokes this method to make the fragment visible on the user’s device.
onResume()	This method is called to make the visible fragment interactive.
onPause()	It indicates that the user is leaving the fragment. System call this method to commit the changes made to the fragment.
onStop()	Method to terminate the functioning and visibility of fragment from the user’s screen.
onDestroyView()	System calls this method to clean up all kinds of resources as well as view hierarchy associated with the fragment. It will call when you can attach new fragment and destroy existing fragment Resoruce
onDestroy()	It is called to perform the final clean up of fragment’s state and its lifecycle.
onDetach()	The system executes this method to disassociate the fragment from its host activity. It will call when your fragment Destroy(app crash or attach new fragment with existing fragment)

Ques= How to integrate fragment into Activity?

Ans=



```
supportFragmentManager.beginTransaction()
    .replace(R.id.fragment_container, FragmentA())
    .commit()
```

fragment_Container → FrameLayout in xml.

Ques= SupportFragmentManager. ??

Ans= SupportFragment Manager

- is part of Android X fragment Library.
- is used to manage framents within the activity
- It provides interfaces to perform operation in fragment Called "Transactions"
  - replace
  - add
  - remove

  See above Exaple
- Fragments are Components that do not exist independtly, they are hosted by an activity

  So "supportfragmentManger" act as Controller which handles followilg tasks
  - ① lifecycle
  - ② Transaction
    - Replace
    - add
    - remove
  - ③ Backstack Management

Imp

Ques ≡ fragment Transaction ??

Ans ≡ • represents a sequence of operations that Can be performed on fragments.

• Transaction → add()
replace()
remove()

Ques ≡ fragment Back-stack ?

↳ It is a mechanism to maintain History of Transactions that users Can navigate back through fragments using the back button.

↳ Default Behavior ≡ • Transactions are not automatically added to the back stack.

• If replaced → Can't be returned back to prev. fragment.

for replace Case

↳ manually have to add fragment.

If you replace `FragmentA` with `FragmentB`, what lifecycle methods are triggered in both fragments?

Answer:

FragmentA:
`onPause()`
`onStop()`
`onDestroyView()`
FragmentB:
`onAttach()`
`onCreate()`
`onCreateView()`
`onStart()`
`onResume()`