Application lifecycles

- Applications and processes may transition between several states
  - active
  - paused
  - stopped
  - inactive
  - suspended
- Callback methods allow applications to prepare for and respond to transitions
Android lifecycle methods

- Lifecycle state transitions trigger event callback methods
- Callback events differ based on type of component
- Callback methods take place in foreground and should be brief
Android Activity

Active lifetime
Visible lifetime
Full lifetime

- Activity.onCreate
- Activity.onStart
- Activity.onRestoreInstanceState
- Activity.onResume
- Activity onSaveInstanceState
- Activity.onPause
- Activity.onStop
- Activity.onDestroy
- Activity.onDestroy

Application is killable
Activity callback methods

- Implementation of lifecycle method should always first call superclass

```java
protected void onPause() {
    super.onPause();
    // ...
}
```
Activity transition

Activity A

ActivityA. onPause

Activity B

ActivityB. onCreate

ActivityB. onStart

ActivityB. onResume

ActivityA. onStop
Android Service

- Service.onCreate
- Service.onStartCommand
- Service.onDestroy
- Service.onBind
- Service.onUnbind
- Service.onRebind
Service binding methods

Activity

bindService()

ServiceConnection.
onServiceConnected

unbindService()

ServiceConnection.
onServiceDisconnected

Service

.onBind()

.onUnbind()
Android BroadcastReceiver

- Only active while handling onReceive() method
Android ContentProvider

- Only active while handling methods supporting ContentResolver
  - query()
  - insert()
  - update()
  - delete()
  - getType()
Processes and lifecycles

- Old processes removed when memory runs low
- Selection base on *importance hierarchy*
  1. Foreground process
  2. Visible process
  3. Service process
  4. Background process
  5. Empty process
iOS Application lifecycle

- UIApplication notified of state transition events
  - Handled by application delegate
    - application:didFinishLaunchingWithOptions:
    - applicationDidBecomeActive:
    - applicationWillResignActive:
    - applicationWillEnterForeground:
    - applicationWillTerminate:
Pre - iOS 4

application:didFinishLaunchingWithOptions:

applicationDidBecomeActive:

Active

applicationWillTerminate:
iOS Application launch

1. Launch the application
2. Load main nib and create application delegate
3. Initialize the application
4. Application enters the foreground

Application Delegate
- application: didFinishLaunchingWithOptions:
- applicationDidBecomeActive:
iOS Transition to background

- Application running
- User presses the Home button
- Application moves to the background

Application Delegate:
- applicationWillResignActive:
- applicationDidEnterBackground:
iOS Return to foreground
iOS Interruption handling

UIKit

Active

Phone, SMS, or Calendar notification arrives

Ignore?

Yes

No

Application moves to background

Delegate

applicationWillResignActive:

applicationDidEnterBackground:

applicationDidBecomeActive:
iOS 4 Application lifecycle

- application:didFinishLaunchingWithOptions:
- applicationDidAppear:
- applicationWillResignActive:
- applicationDidEnterBackground:
- applicationWillTerminate:
- applicationWillEnterForeground:

Ignore interrupt
Reminder

- Proposal due Wednesday