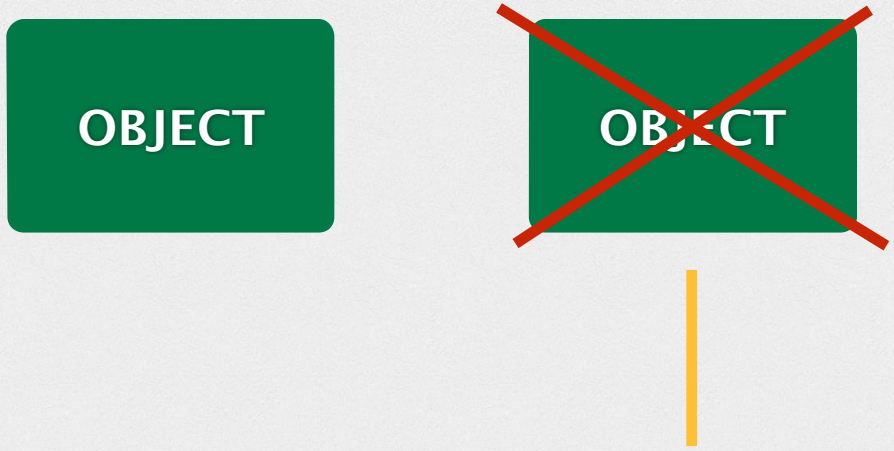
Intermediate Swift

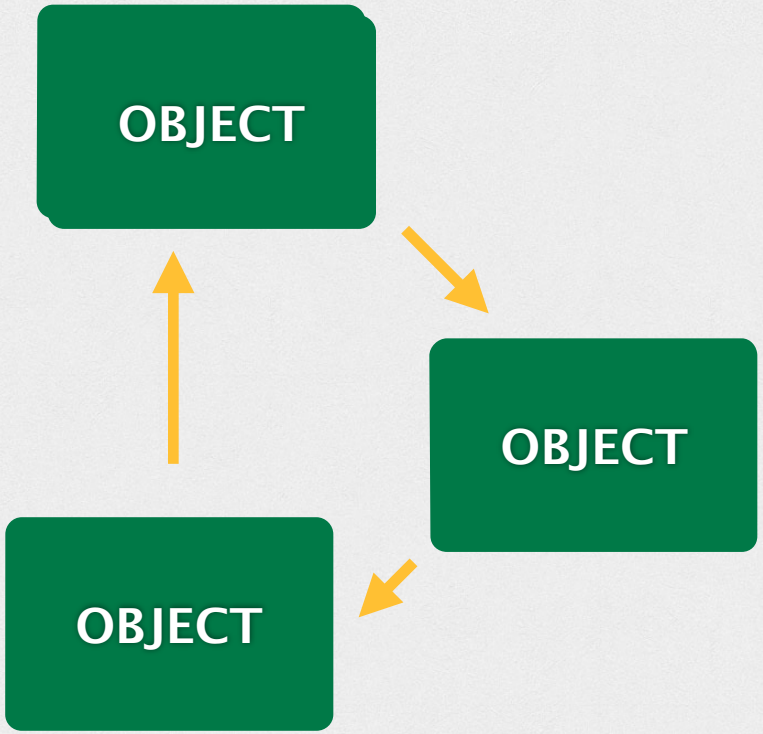
Part 9: Memory Management

raywenderlich.com

Automatic Reference Counting

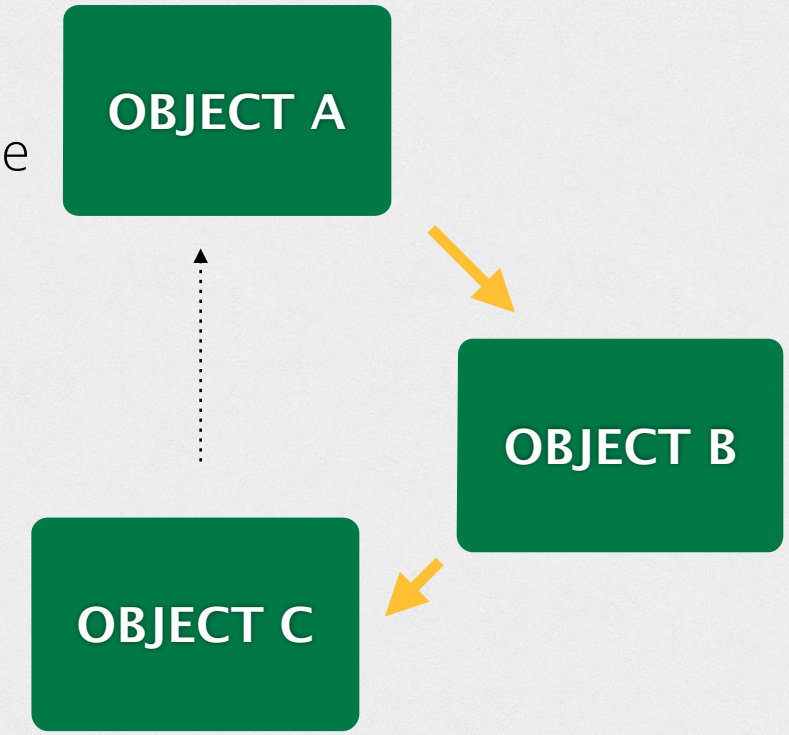


Automatic Reference Counting



Automatic Reference Counting

- Weak Reference
  - Does not stop ARC from deallocating the object.
  - Weak references must always be declared as optional types.
- Unowned Reference
  - Like weak references, but they always hold a value.
  - Accessing a deallocated unowned reference will produce a runtime error.



Memory Management in Closures

- Closures that reference **self** create a retain cycle.
- Capture lists defines the rules when capturing reference types.

```
backgroundQueue.addOperationWithBlock() { [unowned ckRecord, unowned cell] in
  let image = ckRecord.objectForKey("Photo") as CKAsset!
  ...
}
```

Demo

Challenge

```
 var ray = Instructor(name: "Ray Wenderlich")
 var greg = Instructor(name: "Greg Heo")
 var spriteKitSession = Course(name: "Intro to Sprite Kit", instructor:ray)
 var adaptiveLayoutSession = Course(name: "Into to Adaptive Layout",
instructor:greg)
 var rwDevCon = RWDevCon()
 rwDevCon.courses = [spriteKitSession, adaptiveLayoutSession]
```