Interviewing script for iOS

After interview is complete we should be able to assess candidate and set grade on every top level category of the following script.

[Timeline:](#_ie2uejx2uzf7)

[Small talk [5 mins]](#_5k67vry1s501)

[Warm up [5 mins]](#_tt22a6vhfexa)

[Architecture task [15 mins]:](#_4qotdvsl3zzm)

[Interview [50 mins]:](#_9ruhe3s03132)

[Coding assessment [30 mins]:](#_4wnb0e5s1hbk)

[Non trivial tasks [15 mins]:](#_5in7dqbn3w2p)

[Keen to study and teach [5 mins]:](#_ask7kwe2cgq)

[Feedback form:](#_12518zcv62x3)

[Questions list:](#_3lmr83reaur8)

### Timeline:

2 hours 5 mins.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | 5 | 15 | 50 | 30 | 15 | 5 |
| S.T | W.A | Arch.T. | Interview | Coding Assess. | Non T.T | K. |

### Small talk [5 mins]

### Warm up [5 mins]

Inquire about previous experience:

1. What was the most notable/significant project in the past experience?
2. What did you do on prev/current position, how team was structured, why looking for new job/opportunities
3. In the same manner ask about dev-practices, particularities on the current/prev job

### Architecture task [15 mins]:

Some task to create some app architecture:

* Messenger
* Remotely configurable app
* Application with storage and network communications

### Interview [50 mins]:

1. Computer science.
   1. Common data structures, internals, complexity
      1. Hash map
      2. Array
      3. Set
      4. Queue
      5. Stack
      6. Linked list
   2. (Heap, Stack) memory
      1. Stack overflow
      2. OS X calling convention
   3. O(n) notation
   4. Threads
      1. Mutex
      2. Semaphore
      3. Critical section
      4. Barrier
   5. Hash
   6. Common algorithms
2. OOP, Patterns.
   1. SOLID
      1. Single responsibility principle
      2. Open / closed principle
      3. Liskov substitution principle
      4. Interface segregation principle
      5. Dependency inversion principle
   2. Polymorphism encapsulation inheritance
   3. MVC
   4. GoF patterns (profs / cons)
      1. Builder
      2. Factory
      3. Singleton
      4. Composite
      5. Bridge
      6. Mediator
      7. Wrapper
      8. Chain of responsibility
      9. Delegate
      10. Decorator
      11. Observer
   5. iOS SDK Patterns (Which patterns are used in iOS SDK)
   6. MVVM
   7. VIPER
3. Objective-C
   1. Categories
      1. Inheritance and categories
   2. Memory management
      1. Blocks
      2. Weak
      3. Assign
      4. Strong
      5. Retain cycles
      6. Why should we weakify / strongify self
   3. Extensions
   4. Properties modifiers
      1. Assign
      2. Weak
      3. Strong
      4. Retain
      5. Unsafe\_unretained
   5. Blocks
   6. Runtime
      1. Method swizzling
      2. KVO
      3. Associated object
      4. Dispatching table (message sending)
      5. objc\_msgsend
      6. Selector
   7. Callbacks
   8. Protocols
   9. Compiler
4. Swift
5. Multithreading, concurrency
   1. GCD
      1. Dispatch queue
      2. Dispatch sync vs async
      3. Dead locks
   2. NSOperationQueue
      1. NSOperation
         1. Async
         2. Sync
         3. Cancel
   3. Pthread
      1. NSRunLoop in pthread
   4. NSRunLoop
   5. NSThread
6. iOS SDK
   1. UI
      1. Auto Layout
         1. NSLayoutConstraint parameters
         2. Ways to create constraint
         3. Where to add constraints (in which object)
      2. UITableView
         1. Reusable cells
      3. UIScrollView
      4. UINavigationController
      5. Bounds vs Frame
      6. UIResponder
      7. Touch events
   2. NSDictionary
   3. Mutable vs Immutable
7. Networking
   1. HTTP / HTTPS
      1. Handshake
      2. Error codes
      3. Methods
      4. Certificates
   2. NSURLSession
   3. Networking architecture
   4. Sockets
8. 3rd party libraries
   1. AFNetworking
   2. Cocoa pods
   3. Magical record
9. Other languages
   1. C++
   2. Java
   3. Python
10. Clean code, refactoring
    1. Good code vs Bad code
11. VCS
    1. GIT
       1. Resolve conflicts
       2. Rebase
       3. Merge
       4. Commit
       5. Diff
       6. Branch

### Coding assessment [30 mins]:

Examples:

|  |
| --- |
| 1. Implement a function that accepts 3 numbers as a parameters and returns SUM of squares of 2 biggest numbers. E.g. func(3, 1, 2) will return 4 + 9 = 13. |
| 1. Implement a function that returns N number in Fibonacci sequence. Fib(n + 1) = Fib(n) + Fib(n - 1). Fib(0) = 1, Fib(1) = 1 |
| 3. Give an example of code and ask candidate to make a code review |

### Non trivial tasks [15 mins]:

Examples:

|  |
| --- |
| Give a name to the following method:   * (BOOL)someMethod:(int)n {   return (n & (n -1)) != 0;  } |
| What will be printed after execution of following code:  dispatch\_queue\_t queue; // assume we have it  dispatch\_sync(queue, ^{  NSLog(@“first log");  dispatch\_sync(queue, ^{  NSLog(@“second log");  });  }); |
| What problems could cause the following code:  - (void)useALoadOfNumbers {    for (int j = 0; j < 10000; ++j) {    for (int i = 0; i < 10000; ++i) {    NSNumber \*number = [NSNumber numberWithInt:(i+j)];  NSLog(@"number = %p", number);  }    }  } |
| Is everything good with following code?  NSMutableArray \*namesArray = [@[@“ian”, @“john”, @“rob”] mutableCopy]  for(NSString\*name in namesArray) {  if ([name isEqualToString:@"john"]) {  [namesArray removeObject:name];  }  } |

### Keen to study and teach [5 mins]:

1. Ready to contribute to CoP
   1. Articles
   2. Tech talks
   3. Open source
2. What willing to do?
3. Recent changes of iOS API (WWDC, books)

### Feedback form:

### Questions list:

Examples: