**Default interview questions**

General Questions

1. Can you explain what “Composition and inheritance” are? Which is better and why?

2. Closures/blocks… Do you love them? Or do you hate them?

3. Can you give me examples of any functional programing you have done? It can even be something very simple.

4. Do you compromise on business requirements for a story to be “DONE” to finish on time? Or do you communicate that it will be late?

5. Describe the most complicated piece of iOS software you have dealt with

6. What are some strong software opinions you have?

7. Have you done paired programming before? Explain. We do it, so would you be comfortable doing it?

8. Name some bad anti-patterns you’ve had to deal with and how you handled them.

9. Talk to me about testing…

Lone Wolf

1. How do you feel working in a 100% non-siloed environment?

2. How do you deal with conflict with other team members?

3. Have you heard the phrase “Is this the mountain you want to die on?” If so, what does it mean? And how does it apply to you?

4. Have you done paired programming?

Contractor Only

1. Are you looking for contract work? Or something full-time?

2. Can you explain maintenance you have done in your career? How does working full-time vs. contracting have on your approach to work?

3. Do you design with maintainability in mind? If so, explain.

Super Seniors/Leads/Architects

1. How much your day is decision making, and writing actual code?

2. Are you comfortable handling a core contributor role with a team as a whole instead of a traditional “tech lead” role?

3. If you haven’t done TDD, are you open to un-learning what you have learned?

4. In addition to giving orders, are you able to follow them?

5. Can you explain what over-engineering is and why it doesn’t fit with a strict TDD and agile methodology?

iOS Technical Questions

1. What should the View portion in Model-View-Controller be responsible for?

2. Can you discuss the differences between DispatchQueues and NSOperationQueues?

3. When is the last time you wrote a for-loop?

4. Explain your Core Data experience

5. Build/CI/Dependency handling experience?

6. What are child view controllers?

7. Do you like Swift or Objective-C better? Why? If you like Swift, have you heard of Type Erasure?

Questions to ask if they actually know about testing

1. Explain the difference between TDD and Unit testing.

2. Explain the difference between TDD and BDD.

3. Explain what the business gains are for Unit tests vs. Integration tests vs. UITests?

4. How would you TDD dog object?

5. What testing frameworks have you used?