# Hiring Handbook

### Attributes of Good Hiring

* Most of your hires turn out to be great
* Hires stay with the company for a long time
* Hires are productive
* Hires are a great cultural fit

### Learning About a Candidate

* What do they love about the company they are at now?
* What’s great about the specific role there?
* What’s the most interesting project they've worked on and why it was interesting.
* What is the engineering accomplishment(s) they're most proud of
* What are the chances they’re going to be with their current employer in 1 year?
* What are they currently working on?
* What impact is it making for and within the company?
* Why did they choose the project you are currently working on?

### 

### Dimensions to Consider

* Would you have lunch with this person regularly?
* Are they fun to work with?
* Are they passionate about coding?
* Are they interested in learning?
* Are they determined and persistent?
* Are they a smart and get things done?

# 

# 

# Assess Problem Solving & Programming Abilities

## General Programming

### Go / No-Go - finish in ~10 minutes

* remove repeated chars in a string
* is str2 a substring of str1
* given an array, determine if it contains three numbers whose sum equals 0
* is palindrome (level == true, levels == false)
* fizzbuzz "Write a program that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”."

### Medium difficulty - requires some thinking

* I have an array stock\_prices\_yesterday where the indices are the time in minutes past trade opening time, which was 9:30am local time. The values are the price in dollars of Apple stock at that time. For example, the stock cost $500 at 10:30am, so stock\_prices\_yesterday[60] = 500. Write an efficient algorithm for computing the best profit I could have made from 1 purchase and 1 sale of 1 Apple stock yesterday. No "shorting"—you must buy before you sell. You may not buy and sell in the same time step (at least 1 minute must pass).
* Assume you have the facebook graph (nodes are people, edges are one way “friend” relationships). Given 2 people find the friends they have in common.
* Given an array

## Basic Computer Science Fundamentals

* Name some data structures (array, stack, queue, tree, rbtree, trie, etc)
* What is a hashtable/dictionary/associative array. When is it useful?
* Name some search algorithms, when should you use them?

## Basic Software Engineering

* How do you test? How do you know what not to test?
* Besides writing code, what do you do in your job today?
* How do you work with designers?
* How do you find and debug issues in production?

### Assess Problem Solving & Programming Abilities

## iOS Programming

* how long have you been coding for iOS or mac?
* what apps do you like the most and why?
* name the different parts of iOS you are familiar with
* write the following app
  + display an image from the internet in 100 rows of a table, center the image
  + make the images cascade in (100ms delay)
  + handle scrolling
* for different device screens, different orientations, how do you layout images of any aspect ratio?

### 

### 

### Assess Problem Solving & Programming Abilities

### Distributed Systems

* Write a function to find the max entry in an array.
* How fast is it? Can you make it faster?
* How do you do it if the array has million entries?
* How do you do it if the array has trillions of entries?

### Design / Architecture

* Assume your db only supports atomic increment / decrement and you have no other locking primitives. Your db has user accounts that get refreshed via a function called refresh. Write refresh such that a given account can not be refreshed if a refresh is already in progress for that account.
* Design priority inbox for files
* Design basic twitter (people, posts, following, streams). What components would you use (db, index, memcache, web server, etc)? Where do run it? etc...

### 

### 

### Assess Problem Solving & Programming Abilities

### Code Reading - JavaScript

**function** nameMe( obj, map ) {  
 **var** newObject = \_.pick(obj, \_.keys(map))  
 \_.each( \_.keys(newObject), **function**(key){  
 **if**( \_.has(map, key) && map[key] != key ){  
 newObject[ map[key] ] = newObject[key]  
 **delete** newObject[key]  
 }  
 });  
 **return** newObject;  
}

**Code Reading - Ruby**

**def** **self**.ineedaname1(container)  
 loop **do**  
 swapped = false  
 (container.size-1).times **do** |i|  
 **if** (container[i] <=> container[i+1]) == 1  
 container[i], container[i+1] = container[i+1], container[i]  
 swapped = true  
 **end**  
 **end**  
 **break** **unless** swapped  
 **end**  
 container  
 **end**

### 

### 

### References

* [FRC Interviewing](http://firstround.com/review/this-is-how-coursera-competes-against-google-and-facebook-for-the-best-talent/)
* [Facebook interviews](https://www.facebook.com/notes/facebook-engineering/get-that-job-at-facebook/10150964382448920)
* [Hacker Rank](https://www.hackerrank.com/)
* [Interview Cake](https://www.interviewcake.com/)
* [Basic iOS Interview Questions](https://github.com/CameronBanga/iOS-Developer-and-Designer-Interview-Questions)