Acing the Software Engineering Interview

For a while now I am having second thoughts on pursuing a career in the software industry. I did not roll into it naturally with a degree in CS. I initially pursued a degree in Business before I made the switch to Business & Information Technology. I discovered the wonderful world of Machine Learning along the way and slowly drifted towards becoming a Data Scientist / Software Engineer. Yet, since leaving college I have done little with either. I have a high paying job at a defense contractor, but due to the nature of working on an established application my abilities are hardly utilized. I am a software developer not an engineer - it’s time to do something about.

For some time, I have been interested on becoming a fully developed Data Scientist and Software Engineer. Recently I came across John Washam’s blog and GitHub page on how to ace the interview at a big company and decided I want do the same. However, John readily admits he made a few mistakes that he would undo if he could. I intend to learn from them, and I hope you will too.

This list of items will prepare you for becoming a Software Engineer and a Data Scientist at any software company. And just like John Washam’s aim high attitude, that includes Google, Apple, Amazon and Microsoft.

So essentially this list is structured into 2 knowledge parts: must know knowledge for being a Software Engineer and an introduction to Machine Learning.

The list includes references to texts, web pages, YouTube videos, MOOCs and Books. I intend to go easy on reading books and just give you the essential list. Reading takes up a lot of time and I fear knowledge is not retained. The scope of this list is also paired down as much as possible.

# Short description of Goals

## Thing I want to achieve

* **Go through all the relevant subject material - even stuff I already know**. By proceeding through all the material over a period of several months may help me understand the big picture of *Computer Science*.
* **Code a lot.** I want to implement all the relevant examples of algorithms, data structures and applications to gain practical experience.
* **Test a lot.** Code testing is laborious, boring but absolutely necessary. As this guide primarily deals with Java I will write test code using Junit and Mockito.
* **Retain as much knowledge as possible.** Try and find a way to retain the knowledge.
* **Learn practical tips to becoming a Software Engineer**. This means practicing for the interview, deals with issues such as fear and how to work in an Agile / SCRUM way.

## Things I don’t want to do

* **Spend all of my free time on becoming a Software Engineer**. John Washam admits he spent far too much time on this project. Currently I have a job and I don’t need a burnout.
* **Lose overview.** There is such a thing as doing too much. Knowing when to stop is important and I will put that to the test.
* **Not to be afraid to re-invent the wheel**. The goal is not to do something new, the goal is to *learn* to do something new.

# Interview Process & General Interview Prep

Here are some quick tutorials and tips to prepare for the interview process. I think it will give you a quick impression of what you don’t know!

* **ABC - Always Be Coding:** <https://medium.com/always-be-coding/abc-always-be-coding-d5f8051afce2>
* **Whiteboarding:** <https://writing.pupius.co.uk/whiteboarding-4df873dbba2e>

# Pick a language

For the interview you will need to select one programming language. I have made the choice for you as I use Java. However, you will need to know more. You will also need to know a scripting language. The obvious choice is Python, which is also popular with Machine Learning.

# Book List

## Java

* Algorithms (Sedgewick and Wayne)

## Python

* An introduction to Programming in Python (Sedgewick, Wayne and Dondero)

# Software Engineering

## Basics of Java

## Algorithmic complexity / Big-O / Asymptotic analysis

* **Cheat sheet:** <http://bigocheatsheet.com/> also exists as a PDF in this repository.

## Data Structures

For the study of data structures and algorithms a three-pronged approach is used. First, a short descriptive video is linked (usually YouTube). Second, a simple implementation is referenced which the reader can examine. Third and finally a short descriptive text is linked in which the topic is discussed.

* Arrays
* Linked Lists
* Stack
* Queue
* Hash table

## Algorithms

### Sorting algorithms

* Selection sort
  + Video: <https://www.youtube.com/watch?v=cJF9vjz89vU>
* Insertion sort
  + Video: <https://www.youtube.com/watch?v=alh3Jme9LZs>
* Heapsort
* Quicksort
* Merge sort

### Searching algorithms

## Advanced Java

* Recursion
* Object-Oriented Programming
* Design Patterns
* Multi-Threading
* String searching
* Tries
* Unicode
* Endianness
* Network coding

# Machine Learning

# Other topics

## More Books

## More Programming languages

The coding interviews taking by large companies usually concentrate on just one language. The interviewee can choose form a small: usually C, Java or Python. However, there is a much larger world involving databases and web programming. Knowing other languages becomes vital – just not for the interview. Try and choose from the list below.

Go

R

Swift

Ruby

Julia

JavaScript

## Web Programming

## Learning Linux