**UI Technologies:**

1. Motion UI
2. HTML5
3. CSS3

**BIG Data:**

1. Hadoop

**Web Sites:**

1. GitHub
2. StackOverFlow

**Java Script Technologies to learn:**

1. Node JS
2. Angular JS
3. React JS
4. Ember.js
5. Javascript

**Front End CSS:**

1. [Bootstrap](http://getbootstrap.com/)
2. [Foundation](http://foundation.zurb.com/)
3. [MDL](http://www.getmdl.io/)
4. [Less](http://lesscss.org/)
5. [SASS](http://sass-lang.com/)

**Classic full stack framework:**

1. [Play](https://www.playframework.com/)
2. [Spark](http://sparkjava.com/)

**NoSQL:**

1. [RethinkDB](http://rethinkdb.com/)
2. [Redis](http://redis.io/).
3. Mongo DB

**SQL:**

1. Postgres
2. MySQL.

**Tools to learn:**

1. [**Gradle**](http://www.gradle.org/)**:**Build tool. Automates the building, testing, publishing, deployment, and more of software as well as generating static websites or documentation.
2. [**Eclipse**](http://www.eclipse.org/)**:**Open-source integrated development environment (IDE). If you could have just one tool for Java development, Eclipse would be a good choice.
3. [**IntelliJ**](http://www.jetbrains.com/idea/)**:**IDE made by JetBrains, available in an Apache 2-licensed community edition and a commercial edition. IntelliJ provides similar features to Eclipse, with a smooth, developer-friendly experience.
4. [**YourKit**](http://www.yourkit.com/)**:**Java profiler. Combines powerful analysis capabilities, on-demand profiling during both development and production, free embedding into production, and seamless IDE and application server integration.
5. [**Clover**](https://www.atlassian.com/software/clover/overview)**:** Code coverage tool from Atlassian. Runs in your IDE or continuous integration system, and includes test optimization to make tests run faster and fail sooner.
6. [**Mockito:**](http://code.google.com/p/mockito/)Mock library. Open-source testing framework that enables the creation, verification, and stubbing of mocks.
7. [**Jetty**](http://www.eclipse.org/jetty/)**:** Lightweight, embeddable app server.
8. [**Hibernate:**](http://hibernate.org/) Object-relational mapper. Implements the Java persistence API.
9. [**VisualVM**](http://visualvm.java.net/)**:**JVM monitor. An all-in-one Java troubleshooting tool that comes with the JDK.
10. [**JUnit:**](http://junit.org/)Unit test framework. Core tool of test-driven development that enables repeatable, white-box testing.
11. [**Jenkins:**](http://jenkins-ci.org/) Continuous integration tool. Customizable with more than 600 plugins.
12. [**Spring Boot**](http://projects.spring.io/spring-boot/)**:** [Spring](http://spring.io/) application development system. Works in your build system. Supports Gradle and Maven.
13. [**Guice:**](https://code.google.com/p/guava-libraries/) Lightweight dependency injection/inversion of Control (IoC) framework, from Google.
14. [**Guava:**](https://code.google.com/p/guava-libraries/) Utility library. Contains core libraries that Google relies on in Java-based projects: collections, caching, primitives support, concurrency libraries, common annotations, string processing, I/O, and so forth.
15. [**FindBugs:**](http://findbugs.sourceforge.net/) Static code analyzer. Classifies potential errors in code as scariest, scary, troubling, or “of concern.” Available as a standalone GUI or as a plugin for Eclipse, NetBeans, IntelliJ, Gradle, Hudson, and Jenkins.
16. [**Jackson:**](https://github.com/FasterXML/jackson) [JSON](http://www.json.org/) parser. Aims to be fast, correct, lightweight, and ergonomic for developers.
17. [**Snappy:**](https://code.google.com/p/snappy/)Compression/decompression library from Google Code. A great resource when speed is a requirement.
18. [**JD-GUI:**](http://jd.benow.ca/) Decompiler. Standalone graphic utility that displays source codes of “.class” files. Free for non-commercial use (i.e., can’t be included or embedded in commercial products).
19. [**Plumbr**](http://www.plumbr.eu/)**–** Memory Leak Detection
20. [**JClarity**](http://www.jclarity.com/) **– Performance Monitoring**