**Remember the main reason people learning stuff is money. So at first learn the stuff that matters.**

**Restriction : I won’t touch anything new if it is not related to the below group. If something new comes I will put them to the bucket list.**

**I will maintain this routine or rule whatever until I complete the whole task ..**

**I won’t create any new docs for planning. I will maintain this very doc for all the planning and goals.**

**But I might create new bucket lists.**

**Target : I have to be super programmer 😎**

**What need to be good at :**

* **Multithreading (async, sync using netty, nio and blocking io)**
* **~~Docker(microservice)~~**
* **~~Stream api~~**

**~~What need to learn next:~~**

* **~~System design~~**
* **~~Akka~~**
* **~~Aop/~~**
* **~~Rx java~~**
* **~~Ci/CD~~**
* **~~Aws~~**

**—------------------------------------------------------------------------------------------------------------------------**

**Currently Learning stuff :**

* **Multithreading (async, sync using netty , nio and blocking io) → every day [main priority] [catagory 1]**
* **Solve problem from leetcode or codeforces (50 easy – 50 medium and 50 hard) :: job preparation – at least 3 days in a week [catagory -2]**
* **Block chain**

**After completing the multithreading I have to create a portfolio using (docker,java,spring,aws) and platform would be (web and mobile) I won’t use kotlin that would depend on the company.**

**—------------------------------------------------------------------------------------------------------------------------**

**Extra focus :**

* **Block chain (dapp)**

**~~Interests :~~**

* **~~SWE~~**
* **~~Block chain~~**
* **~~Ml~~**
* **~~Embedded system~~**

**My weak points that I need to rework:**

* **DS/Algo (I know them partially)**
* **Productivity**

**Plan to proceed :**

At first I will do block chain 1 month straight, then if I think there should be much to learn then I can extend 1 month more. Then I will start machine learning.

**============================>>>><<<<============================**

**Dividing skill set in 4 different part :**

* **Category 1 : The mandatory stuff for current working profession(swe or block chain or ml or embedded whatever) [high priority]**
* **Category 2 : Skill sets/knowledges that are directly connected to the current profession(e.g : react js , new languages, maths , learning about a specific technology) [second high priority]**
* **Category 3 : Learning something that is not connected with the current profession (machine learning, block chain , embedded system , art etc) [low priority]**

**What do I mean by being a super programmer ?**

The meaning of super programmer to me is, a person who knows his/her tools better. They can think very fast. The person who has done a hell lot of programming and is very proficient and willing to do any type of hard work needed to make that project successful.

So how to be a good programmer and also a swe?

First I have to make sure that I am good at the language that I am using.

Second, I have cleared all the doubts that I have related to the skill sets that I have already acquired.

I have to write some complex projects and be consistent until I complete them and try to use all the best practices that I can.

Moral of the story is I have to do a lot of programming.

Note : While learning anything my main problem is I rush too fast. I should calm myself down and try to understand the documentation properly otherwise It will waste a lot of time.