A list of curated helpful materials for CS path

#Quotes and Ideas

- “Indecision creates information debt. Action repays it.”
- "work expands so as to fill the time available for its completion" | Parkinson's law
- "Let us weigh the gain and the loss in wagering that God is. Let us estimate these two chances. If you gain, you gain all; if you lose, you lose nothing" | Pascal's wager
- Roadmap for career: “Achievements (CV) > Skill > Networking / Business literacy / how to sell yourself (your skill) > Leadership / Teamwork”
- “The Pareto Principle, also known as the 80/20 Rule, is the observation that for many outcomes, roughly 80% of consequences come from 20% of causes.”
- “Whatever you choose to do, look for ways to monetize it. Learn two skills: One that the market needs, and another one that aligns with your hobby. Earn using the skills first and then invest in your hobbies.” - MK Shishir | “Tumi pochondo koro sheita diye always taka ashbe emon na .... Taka jekhan theke ashe sheitake pochondo koro ....”
- “All these coding tools are great, but none of them will teach you what not to do. That part only comes from doing the same things over and over, making mistakes, and figuring it out yourself. Eventually, the steps just stick.” - Talha Chowdhury
- Idea: Watch videos of newly learned topics. (ex. Saw a Facebook post on vite+npm and watched videos on YouTube)
- “Visualization is key when it comes to programming. I am a believer of visualization more than theory.” - Anik Sir, UITS
-

- Explore
    - Math Olympiad
    - https://github.com/trending
    - Good Guides
        - Talha Chowdhury https://talhachowdhury.com/
        -
    - https://roadmap.sh
        - Guides - roadmap.sh

        - [Best Practices]()
        - [Illustrated Videos - roadmap.sh]()
    - https://80000hours.org/ | You have about 80,000 working hours in your career: We aim to help you work out how you can best use your 80,000 hours to help others, and to take action on that basis

- Research
    - https://github.com/Tuccinator/hn-moocs
    - https://www.coursera.org/learn/research-methods/home/welcome

- CV / Resume
    - Jake's Resume https://www.overleaf.com/latex/templates/jakes-resume/syzfjbzwjncs
    - Google X-Y-Z Resume Formula https://majestic-epoch-fb4.notion.site/Google-X-Y-Z-Resume-Formula-16ed553489b24c078c5c1d2bfd60ca7a
    - Awesome CV - Overleaf, Online LaTeX Editor https://www.overleaf.com/latex/templates/awesome-cv/dfnvtnhzhhbm
    - r/EngineeringResumes Wiki: Resume Templates https://www.reddit.com/r/EngineeringResumes/wiki/resumetemplates/
    -
- Readings
    - Manga list for life lesson: "Berserk > Vagabond > Monster > Vinland"
    - James Scholz's Top 3:
        - Deep Work Book by Cal Newport
        - Atomic Habits Book by James Clear
        - Can't Hurt Me Book by David Goggins

- YouTube Playlists
    - 3b1b video https://youtube.com/@3blue1brown
    - TED Talks
    - Essence of calculus - YouTube https://www.youtube.com/playlist?list=PLZHQObOWTQDMsr9K-rj53DwVRMYO3t5Yr

I believe 3Blue1Brown's Playlist that discusses Calculus in detail is a great way to learn the actual theory behind calculus and every student should go through it instead of just memorizing formulas and proof methods.

  - https://www.youtube.com/@profjeffreykaplan/playlists
    - How to Do Well in College - YouTube

- Study Better
  - Marty Lobdell - Study Less Study Smart
  - I think that every student should have a good grasp of the topics taught in the **Discrete Math** course. As many CS concepts are built on top of the basic concepts of Discrete math, it should act as a solid introduction to the CS career much like CS50 https://pll.harvard.edu/course/cs50-introduction-computer-science
  -

- How to?
  - How To Study Programming The Lazy Way
  -

- Project Farming
  - Just cramming theories and a measly amount of practical learning is not beneficial to anyone. I believe that doing projects and learning along the way is the best approach for any study topic. And for CS students, having experience in projects should be mandatory for both their academic and career after graduation. These are 2 GitHub repositories that list some of the projects:
    - https://github.com/practical-tutorials/project-based-learning
    - https://github.com/codecrafters-io/build-your-own-x

    I am also interested in OSSU. They have a CS curriculum filled with open-source projects, some with reputed certifications. This is their curriculum:
    - https://cs.ossu.dev/#curriculum

    Find other projects: https://trendshift.io/

- Interview
  - https://www.techinterviewhandbook.org/ | Tech Interview Handbook goes straight to the point and tells you the minimum you need to know to excel in your technical interviews. Having personally gone through the interviewing process, it was frustrating to have to find resources from everywhere to prepare for my technical interviews.
  - https://www.geeksforgeeks.org/blogs/interview-preparation/ | The preparation for acing a tech interview starts with a complete and worthwhile roadmap or preparation plan. It is quite obvious that until and unless you won't know what to prepare, where to prepare, what subjects hold more weightage, etc.

- Reddit Guides
  - https://www.reddit.com/r/learnprogramming/wiki/faq/
  - https://www.reddit.com/r/cscareerquestions/wiki/index/

- Reliable/Reputed/Certification Courses (Source)
  - https://grow.google/certificates/
  - https://www.freecodecamp.org/
  - https://www.netacad.com/catalogs/learn
  - https://www.mooc.org/#course-categories
  - https://ocw.mit.edu/search/ | https://ocw.mit.edu/courses/8-01sc-classical-mechanics-fall-2016/
  - https://www.edx.org/ | https://www.edx.org/cs50
  - https://www.coursera.org/
  - https://www.udemy.com/
  - https://pll.harvard.edu/course/cs50-introduction-computer-science
  - https://education.oracle.com/ | https://education.oracle.com/database/oracle-database/pFamily_32

- Other Sources | **https://www.classcentral.com/** Class Central aggregates courses from many providers to help you find the best courses on almost any subject, wherever they exist.
  - https://backbencher.club/
  - https://teachyourselfcs.com/

    - https://labex.io/linuxjourney
    - https://www.w3schools.com/
    - https://www.w3resource.com/index.php
    - https://www.geeksforgeeks.org/
    - https://cses.fi/
        - https://cses.fi/book.pdf | https://cses.fi/book/index.php | https://codeforces.com/blog/entry/50728 | https://www.geeksforgeeks.org/competitive-programming/competitive-programming-cp-handbook-with-complete-roadmap/
    -

- Visualizer
    - https://csvistool.com/
    - https://visualgo.net/en
    - https://www.cs.usfca.edu/~galles/visualization/Algorithms.html

- Practice
    - Emotional Intelligence??
    - Cyber Security → https://ctf.hacker101.com/ | https://tryhackme.com/
    - Programming
        - Problemsets → https://neetcode.io/ | https://youkn0wwho.academy/topic-list
        - Online Judge → https://www.hackerrank.com/ | https://codeforces.com/ | https://leetcode.com/
        - Tutorial sources → Bro Code | Farhan Hossan
    - Database →
    -
    - Programming Language
        - C++ → https://cplusplus.com/ | https://www.learncpp.com/ | https://www.learn-cpp.org/ | https://devdocs.io/cpp/ | https://cppreference.com/
        - Python → https://docs.python.org/3/ | https://www.learnpython.org/ |
    - Human Language
        - Arabic → Arabic 101 - YouTube |

        - English
-