Self Study for Computer Science and Engineering

Self Study Plan

This document organizes resources for self-learning computer science, engineering, electronics, and mathematics, including free and paid platforms.

---

1. Math (Foundation for Engineering & CS)

- OneMathematicalCat.org - Algebra, Precalc, Calculus
- Greenemath.com - Algebra & Precalc practice
- CalcWorkshop.com – Calculus
- Geometry
- https://www.allmath.com/
- desmos.com
- AllTheMath.org / Discrete Math - Logic, sets, graphs, combinatorics
- MathIsPower4U.com - Videos for all math topics
- Runestone Academy - CS-focused math exercises
- ProPrep.com - Optional paid practice
- Engineer4Free.com - Free math & basic engineering resources

---

2. Programming & Computer Science Fundamentals

- **OSS University Path to a free self-taught education in Computer Science!**
- NeetCode.io - Algorithms, DSA practice (lifetime membership)
- Introduction to parallel programming (in c++)
- Software contruction elements
- software contruction abstraction

- Java Programming
- CS61A - Intro to CS, Python, Data Structures
- TeachYourselfCS.com - Core CS concepts
- LearnCPP.com - C++ basics & intermediate
- Technology Library
- HackingCPP.com - C++ exercises
- StudyPlan.dev - Structured C++ curriculum
- BigMachine.io - Self-taught programming exercises
- Sp21.datastructur.es - Data structures & algorithms course
- C++ multithreading Modern C++ Features & Concurrency

---

3. Systems & Low-Level Programming & intoducton to Engineering

toptechboy.com

- Nand2Tetris.org - Computer architecture & logic design
- turning complete game
- NandGame.com - Hands-on digital logic
- Advance C++ course
- Introduction to Electricity and Magnetism Specialization
-
- wokwi.com
- Linear Circuits 1: DC Analysis
- Linear Circuits 2: AC Analysis
-
-
- Electric Power Systems

Digital Signal Processing Specialization

-
- Computer Architecture
- Electrical engineering 101
- Teach yourself Electrcity and electronics
- Assembly Language PDFs:
- Learning the art of electronics
- Introduction to 64-bit Intel Assembly Language Programming for Linux
- Assembly Language Step-by-Step Programming with Linux 3rd Edition
- Introduction Operating system
- CS423 Operating System Design - OS concepts
- LabEx.io - Linux virtual lab environment
- FastbitLab.com - Embedded programming
- QT course udemy for c++ beginer course link

Introduction to Engineering Book

Design of Digital Circuits with VHDL Programming

Crash Course Electronics and PCB Design

Mastering Microcontroller: STM32-LTDC, LCD-TFT, LVGL (MCU3)

Mastering Microcontroller and Embedded Driver Development

4. Networking & Telecommunications

- Teracom Training - Online telecom courses
- TCP/IP Guide - Networking theory
- Computer Networking: A Top-Down Approach PDF
- Su25 CS168.io - Networking exercises

---

5. Electronics & AI & Robotics

- DCACLab.com - Online circuits
- Power systems
- CircuitLab.com - Circuit design
- AllAboutCircuits.com - Electronics tutorials
- Ohmify.com - Electronics fundamentals
- Kevin Wood Robotics - Robotics projects
- Udacity Robotics Software Engineer - Robotics programming
- Data camp

Quirk: Quantum Circuit Simulator

---

6. Physics

- FlippingPhysics YouTube - Physics tutorials
- PhysicsLab.app - Interactive simulations
- Quntum Computing
- MyPhysicsLab.com - Physics simulations
- PhysicsLab.org - Interactive physics exercises

---

7. Supplemental / Special Topics

- DevOps: DevOps Bootcamp
- Software Engineering: Coursera, Udemy
- GPU Programming: Coursera GPU specialization
- Parallel / Concurrent Programming: Udemy C++ concurrency courses
- Web Development / Android / Kotlin: Udacity
- AI / Machine Learning: Coursera AI Intro

---

8. Study & Practice Tools

- Visualgo.net - Visualize algorithms & data structures
- CSVIsTool.com - Interactive algorithm visualization
- OverTheWire.org / Bandit - Cybersecurity practice
- Excalidraw.com - Diagramming & flowcharts
- Timer

- Roadmap.sh - Career learning paths & skill maps
- brain.fm – focus music and relax music

Usage Tips

1. Start with Math, then move to Programming Fundamentals.
2. Parallel-track Discrete Math for logic skills.
3. Continue to Systems & Low-Level Programming, then Networking.
4. Follow with Physics, Electronics & Robotics.
5. Use study/practice tools to reinforce learning and visualize concepts.

Note: Clickable hyperlinks are included for each online resource.

Self Study Plan for Computer Science and Engineering

Additional Information

About Me:
https://antp1997.github.io/about.html

Computer Science Information (OSSU):
https://ossu.firebaseapp.com/

Course Pace:
- Finish in approximately 5 months
- Take only 3 to 4 courses at a time