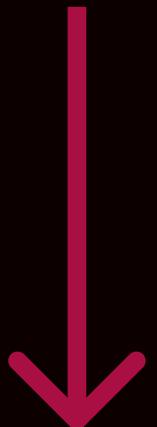
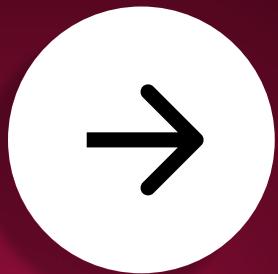


ERROR 307!



CSE Student Induction Program

A roadmap for first-year students to grow in tech.



Welcome to Computer Science

1. What is CSE?

Tech isn't just coding – it's solving real-world problems through apps, AI, cybersecurity, and more. As beginners, what usually people lack is the language of the problem space. This language is not python, or Java, or even C. Its core principles of computer science.

Its understanding how a computer works and the data structures and algorithms that are endemic to converting that which is in the problem space to the solution space. Regardless of programming language or operating system or hardware. Study from first principles and the programming language will come naturally based on the best fit for your problem.

Whether that's embedded, cryptography, kernel, ML, DS, AI, Web, etc.. learning the language is the least of your challenges. Check out these resources. This is hard stuff. Its supposed to be. Otherwise anybody can do it by simply learning the syntax of an "easy" language.

1. [Code: The Hidden Language of Computer Hardware and Software](#)
2. [Exploring How Computers Work](#)
3. [Watch all 41 videos of A Crash Course in Computer Science](#)
4. [Take the CS50: Introduction to Computer Science course.](#)
5. [Take the Build a Modern Computer from First Principles: From Nand to Tetris \(ProjectCentered Course\)](#)
6. [Ben Eater's Build an 8-bit computer from scratch](#)
7. [Here is a decent list of 8 Books on Algorithms and Data Structures For All Levels](#)

You can also check out [Teach Yourself Computer Science](#) And finally, [play the long game when learning to code](#). Reading the first mentioned book is mandatory, I'd say read it before learning any programming language, it's worth the time.

Next is the CS50 course, you probably know about it, take it if you haven't already.

🔗 Explore paths: [MIT Open Course Ware](#)

2. Passion > Skill

Start with "Why?" – build projects you care about (like a meme generator), not just textbook examples.

🔗 Inspiration: [CS50's First Week](#)

3. Build Systems

Code 30 mins daily >> 5-hour weekend marathons. Track streaks with [GitHub Contributions](#).

Growth Mindset

4. Confidence ≠ Ego

Admit "I don't know" – here's how to ask:

- Bad: "Can someone help?" Good: "
- Stuck on Python loop syntax – tried X and Y. Screenshot attached."
(dontasktoask.com)

5. Communication 101

Practice elevator pitches:

"I built a chatbot using Python's NLTK library to handle customer queries."

- Improve: Toastmasters International
- How to ask quality questions: <https://youtu.be/tYnGiWlwcj4?si=73QFk74-gxZi2vqi>

Tech Toolkit

6. First Programming Language

Python: Automate Boring Stuff

JavaScript: [FreeCodeCamp](#)

7. Git/GitHub Basics

git clone [URL] → Download projects

git commit -m "Added login feature" → Save checkpoints

🔗 Tutorial: [GitHub Hello World](#)

8. Solve Real Problems

Practice on:

- [LeetCode Easy](#)
- [Advent of Code](#)

Career Launchpad

9. Internship Hunt

Where to look:

LinkedIn → Follow #internships

AngelList → Startup roles

10. Resume That Stands Out

BAD: "Knows Python"

GOOD: "Built attendance tracker using Python (reduced manual work by 30%)"

🔗 Template: Harvard Resume Guide

Community & Safe

11. Join Communities

Dev.to → Blogs

Stack Overflow → Q&A

12. Avoid Scams

Red flags:

"Pay ₹5000 for guaranteed internship"

"Download this .exe for interview prep"

Failure → Growth

13. Handle Rejection

Post-interview ritual:

8. Write feedback
9. Fix 1 weakness (e.g., array problems)
10. Apply again in 2 weeks

🔗 Read: [Google's Failure Method](#)

Build Your Brand

14. LinkedIn Profile

Headline: "CSE Student | Python Enthusiast" > "Looking for job"

Post project demos with #buildinpublic

15. Portfolio Website

Free hosts:

[GitHub Pages](#)

[Netlify](#)

Start Today

16. 30-Day Challenge

Days 1-5: Python basics on [Codecademy](#)

Days 6-15: Build a calculator

Days 16-30: Add features (history, themes)

17. Stay Updated

Podcasts:

[Nikhil Kamath](#)

[Lex Fridman](#)

Curated AI Prompts

Prompt: "Provide me a detailed roadmap for [Skill Name] covering [X] months, including free resources, websites, and YouTube channels."

Example: "Provide me a detailed roadmap for Java covering 6 months, including free resources, websites, and YouTube channels"

Prompt: "Suggest some unique and advanced project ideas for [Skill Name] that are not basic ones like a calculator or to-do list."

Example: "Suggest some unique and advanced project ideas for Web Development that are not basic ones like a calculator or to-do list."

Prompt: "I am facing an issue in my [Programming Language/Technology] code. Here's my problem: [Describe your issue]. Can you help debug it?"

Example: "I am facing an issue in my Python Django backend. Here's my problem: The API is not returning the expected JSON response. Can you help debug it?"

Prompt: "Which is better for the future: [Skill A] or [Skill B]? What are the key differences and job opportunities?"

Example: "Which is better for the future: Data Science or Cybersecurity? What are the key differences and job opportunities?"

<https://www.promptingguide.ai>

Bonus: Cheat Sheet

Topic	Best Resources
Web Development	MDN Web Docs
Open Source	First Contributions
Full-Stack Development	The Odin Project
Machine Learning	Coursera - Andrew Ng's ML Course
Cloud Computing	AWS Training
Python Programming	Real Python
Angular	Angular Official Docs
Database Design	Database Design Book by Michael J. Hernandez
Microservices	Microservices.io
Machine Learning APIs	Hugging Face
DevOps	Learn DevOps - The Complete Kubernetes Course
Java Programming	Java Programming by GeeksforGeeks
JavaScript	JavaScript.info
Node.js	Node.js Official Docs
React	React Official Docs
Java Web Development	Spring Framework
Python Web Development	Flask Documentation
Java OOP	Object-Oriented Programming in Java - Udemy
JS Frameworks	Vue.js Official Docs

Bonus: Cheat Sheet

Next Steps:

11. Bookmark 3 resources from this guide

12. Join 1 coding community this week

13. Build a "Hello World" program in any language

Remember: Every expert was once a beginner who kept showing up.