**FLOWCHART TO BECOME A PROFESSIONAL SOFTWARE ENGINEER AND END LIFE**

**Different Types of Software Engineers:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Front-end Engineer | Back-end Engineer | Full-stack Engineer | DevOps Engineer | Data Engineer |
| ML Engineer | AI Engineer | Mobile Engineer | Embedded Systems Engineer | Game Engineer |
| Security Engineer | Cloud Engineer | Site Reliability Engineer | Blockchain Engineer | Test Automation Engineer |
| AR/ VR Engineer | Database Engineer | System Software Engineer | NLP Engineer | Firmware Engineer |
| API Engineer | Metaverse Engineer | Robotics Software Engineer | Ad Tech Engineer | FinTech Software Engineer |
| GIS Engineer | Automotive Software Engineer | Bioinformatics Software Engineer | Game Engine Engineer | Quantum Computing Software Engineer |
| Search Engine Engineer | Privacy Engineer | Biohacking & Neurotech Engineer | Space Software Engineer |  |

Set Yourself for Success, Be Discipline & Confident

Work Towards Multinational Companies like Google

If You Want to be a Professional Software Developer (Web Development, App Development, etc.) Learn **OOP first** because most real-world applications are built using object-oriented principles.

Learn Data Structure & Algorithm (DSA)

Learn Object Oriented Programming (OOP)

Learn Development Tools

Build Some Initial Projects and Practice Coding

Learn Programming Language as Per Choice

**Software Engineering Specialization Areas: (Common)**

* Front-end Development
* Back-end Development
* Full-stack Development
* Mobile Development
* DevOps
* Cloud Computing
* Cyber Security
* Data Science
* Machine Learning
* Artificial Intelligence
* Desktop Applications Development

Database Architecture and SQL

* DSA, System Design
* Fundamentals of CS
* Aptitude/ Behavioral Question
* Showcase Projects
* GitHub
* LinkedIn
* Communication (English)
* Teamwork
* Time Management
* Competitive Programming
* Problem Solving
* System Design
* UI/ UX Basic Principles

1. Version Control (Git/ GitHub)
2. Frameworks
3. Library

For Examples:

* Web Development: JavaScript
* App Development: Java, Swift.
* ML/ AI/ Data Science: Python

For Examples:

* Web Development
* App Development
* ML/ AI/ Data Science

Build a Strong Resume and Portfolio (Refers Your Skills)

End Life

Continuous Learning and Growth

Enhance Programming Skills

Develop Soft Skills

Prepare For Technical Interview

Apply For Jobs in Tech Companies

If You Want to be a Strong Problem Solver (Competitive Programming, AI/ML, System Design, etc.) Learn **DSA first** because it improves your logical thinking and helps in writing efficient code.

Choose a Field of Interest