That's a great realization and a very common situation — you're **not alone** in feeling this way. Many engineering students (even in computer-related fields) don’t start with any programming knowledge. The fact that you're asking for help and want to improve is a **strong first step**.

Let’s break it down into simple, structured steps so you can start your journey to becoming a **software engineer**, even if you’re starting from zero.

**🚀 Step-by-Step Roadmap to Become a Software Engineer**

**🔹 1. Understand What a Software Engineer Does**

A software engineer:

* Writes code to build applications, websites, systems, etc.
* Understands how to design software that works well and can scale.
* Works on teams with other developers, testers, and designers.
* Solves real-world problems using technology.

💡 **Designations in software engineering** (you'll understand these better as you progress):

* **Frontend Developer** (focuses on UI/design – what users see)
* **Backend Developer** (focuses on logic, databases, server-side)
* **Full Stack Developer** (both front and back end)
* **Mobile Developer** (Android/iOS apps)
* **DevOps Engineer** (automation, deployment)
* **Data Engineer/Scientist** (data handling, machine learning)
* **QA Engineer** (testing and quality assurance)
* **Security Engineer** (focuses on cybersecurity)

**🔹 2. Start with Programming Basics (Coding)**

Start with an **easy and useful** language: **Python**.

**Resources:**

* ✅ [w3schools Python Tutorial](https://www.w3schools.com/python/)
* ✅ [Python for Beginners - freeCodeCamp YouTube](https://www.youtube.com/watch?v=rfscVS0vtbw)

**Concepts to Learn:**

* Variables, Data Types
* If-else conditions
* Loops (for, while)
* Functions
* Lists, Dictionaries
* Basic problem-solving (practice on [HackerRank](https://www.hackerrank.com/skills-directory/python_basic) or [LeetCode](https://leetcode.com/))

📌 **Practice 30 minutes to 1 hour daily.** Make it a habit.

**🔹 3. Learn How Software Is Built (Big Picture)**

Understand:

* What is **frontend** and **backend**
* How websites/apps are made
* How developers **collaborate using Git & GitHub**

📘 Learn Git:  
[Git & GitHub Crash Course](https://www.youtube.com/watch?v=RGOj5yH7evk)

📘 Learn Web Basics (HTML, CSS, JS):  
[freeCodeCamp Responsive Web Design Course](https://www.freecodecamp.org/learn/)

**🔹 4. Choose a Path You Like (Specialization)**

You don't have to pick now — explore first. Here's what each path might involve:

| **Path** | **What You Build** | **Key Skills** |
| --- | --- | --- |
| Frontend Dev | Websites, UI | HTML, CSS, JavaScript, React |
| Backend Dev | Server-side logic | Python, Node.js, Databases |
| Full Stack | Both front and back | Combination of above |
| Mobile Dev | Android/iOS apps | Kotlin, Swift, Flutter |
| Data Science | Data analysis, ML | Python, Pandas, ML basics |
| DevOps | Automation, Cloud, CI/CD | Linux, Docker, AWS, Bash |

Try a few mini projects from different areas to see what you like!

**🔹 5. Work on Real Projects**

Projects show you can build real things — employers love that.

💡 Example Starter Projects:

* To-do app (frontend)
* Calculator or Quiz (Python)
* Weather app using API (backend)
* Personal portfolio website
* Simple chatbot

**🔹 6. Build Your Resume and GitHub Profile**

✅ Keep your code and projects on GitHub  
✅ Start writing a basic resume  
✅ Add a LinkedIn profile (important later)

**🔹 7. Internships and Industry Exposure**

In your 3rd year, start applying for:

* Internships (remote or local)
* Open-source projects
* College-level tech fests, hackathons

Use platforms like:

* Internshala
* LinkedIn
* HackerRank
* AngelList
* GitHub (for open-source)

**🔹 8. What to Do RIGHT NOW**

Here’s your **Week 1 Plan**:

| **Day** | **Task** |
| --- | --- |
| Mon | Install Python, write your first program |
| Tue | Learn variables and data types |
| Wed | Learn if-else, write simple conditions |
| Thu | Practice loops |
| Fri | Functions and small programs |
| Sat | Lists and dictionaries |
| Sun | Build your first mini project (e.g. calculator, number guessing game) |

**🎯 Final Words of Motivation**

* Start **small but consistent** – it’s a marathon, not a sprint.
* No one becomes a software engineer overnight.
* Don’t be afraid to Google everything — even pros do!
* You don’t need to know everything, just keep learning step by step.

If you want, I can:  
✅ Build a custom weekly learning plan  
✅ Suggest YouTube/video courses  
✅ Help you choose a specialization  
✅ Help with projects, GitHub, resume

Let me know what you'd like to do next.