Step 1

* Python intro course CS50 on python
* Prompt Engineering – Coursera}

Step 2

* Finish Q Learning
* Data structures and algorithms:
  + CS50x 2024
  + **C++ Programming Essentials Professional Certificate:** <https://www.edx.org/certificates/professional-certificate/ibm-c-programming-essentials>
    - Three courses: C++, OOP, and **Data Structures and Algorithms**
    - $357

Step 3

* Software Design Principles Python

Step 4

* Software Design Principles C++

1. **Data Structures and Algorithms**:
   * **Focus**: Understanding various data structures (like lists, stacks, queues, trees, graphs) and algorithms (sorting, searching, dynamic programming).
   * **Why**: These are fundamental to solving complex problems efficiently and are a core part of any computer science curriculum.
   * **Resources**: Books like "Introduction to Algorithms" by Cormen et al., online courses from platforms like Coursera or edX.
2. **Advanced Programming**:
   * **Focus**: Mastering a programming language (beyond Python, consider learning C/C++ or Java), software design principles, and debugging techniques.
   * **Why**: Building software is more than just writing code; it's about writing good, maintainable, and efficient code.
   * **Resources**: Books like "Effective Java" by Joshua Bloch (for Java), or online tutorials for advanced language-specific topics.
3. **Operating Systems**:
   * **Focus**: Learning about processes, threads, memory management, file systems.
   * **Why**: Understanding how operating systems work is crucial for efficient programming and resource management.
   * **Resources**: "Operating System Concepts" by Silberschatz, Galvin, and Gagne.
4. **Computer Networks**:
   * **Focus**: Basics of networking, OSI and TCP/IP models, routing, switching, network security.
   * **Why**: Knowledge of networks is essential as most applications today are networked.
   * **Resources**: "Computer Networking: A Top-Down Approach" by Kurose and Ross.
5. **Databases**:
   * **Focus**: Database design, SQL, NoSQL, transactions, concurrency control, database optimization.
   * **Why**: Data is at the heart of many applications, and knowing how to store, retrieve, and manipulate data is key.
   * **Resources**: "Database System Concepts" by Silberschatz, Korth, and Sudarshan.
6. **Software Engineering**:
   * **Focus**: Software development lifecycle, methodologies (like Agile, Scrum), testing, version control, CI/CD.
   * **Why**: Building software in a team setting requires understanding these principles.
   * **Resources**: Online courses on software engineering practices, collaboration tools like Git.
7. **Electives/Specializations**:
   * **Options**: Web development, mobile app development, machine learning, security, etc.
   * **Why**: Diversifying or specializing based on interests or career goals.
   * **Resources**: Specific courses or projects in areas of interest.
8. **Capstone Project**:
   * **Goal**: Implement a substantial project that incorporates the learned skills.
   * **Why**: Demonstrates ability to apply knowledge in a practical, cohesive manner.
   * **Approach**: Could be a software application, a contribution to open-source, or a research project.
9. **Continuous Learning**:
   * **Focus**: Keeping up-to-date with the latest technologies and methodologies.
   * **Why**: The field of computer science is always evolving.
   * **Resources**: Online tutorials, workshops, conferences, and reading research papers.