

C Programming + Computer Science Foundations Roadmap

Introduction

This 4-week roadmap is designed for learners transitioning from Python to C. It not only teaches C syntax, but also foundational computer science concepts like memory management and data representation. By the end, you'll understand how high-level languages like Python work under the hood.

Week 1: Getting Started with C

- Installing a C compiler and writing your first C program
- Variables and data types (int, float, char)
- Input and output using printf and scanf
- if-else and switch statements
- Comparison with Python: static vs dynamic typing

Week 2: Control Flow and Functions

- for, while, and do-while loops
- Nested loops and logical operators
- Defining and calling functions in C
- Passing arguments by value
- Scope and lifetime of variables
- Python vs C: indentation vs braces, function rules

Week 3: Memory and Pointers

- Stack vs Heap memory
- Introduction to pointers and addresses
- Arrays and strings in C
- Using pointers with arrays and functions
- Dynamic memory allocation: malloc, calloc, free
- Python abstraction vs C control

C Programming + Computer Science Foundations Roadmap

Week 4: Structs, File I/O, and Foundations

- Structures (struct) and use cases
- File handling: fopen, fprintf, fscanf, fclose
- Introduction to data structures (linked lists overview)
- How computers store and manage data
- Final mini-project: student database or address book (struct + file I/O)
- Reflection: what C teaches us about how code works

Final Notes

This roadmap builds a solid base in C and computer science. Each session includes hands-on practice, code walkthroughs, and real-time debugging help. Concepts are reinforced with relatable examples and comparisons to Python.

Optional follow-ups: linked lists in detail, sorting algorithms, or transitioning into embedded systems or operating system concepts.