****AI-Integrated 6-Month Coding & Problem-Solving Plan****

Note: For optimal progress, I recommend the student have at least 3 one-hour sessions/week recommended for best results. This regular practice is essential for consistent improvement and helps the student stay engaged with the material. Please consider implementing this suggestion for the best outcomes.

AI Tools : GitHub Copilot, Cursor, ChatGPT — prompt design, code review, tests, docs

---

Month 1 – Programming Foundations (With & Without AI)

1. Introduction to programming:

- What is a programming language?

- Understanding syntax in programming (using examples from simple English sentences)

- Difference between high-level and low-level languages.

- What is a compiler/interpreter?

2. Memory Fundamentals:

- What is memory?

- How data is stored in memory using cell notation, with an emphasis on how each piece of data has a specific address.

3. Introduction to variables:

- What is a variable? (using names to illustrate) and how variables map to memory.

- Explanation of how variables can store values for future use and can be updated as needed.

4. Functions:

- Introducing the concept of functions (using examples like dialing a WhatsApp call to different numbers or an additional function).

5. Input and output:

- Understanding input and output with examples (e.g., sending a message).

- Introduction to Python's `input` and `print` functions.

- Mini-project: Simple calculator.

6. Writing the first program with AI Integration:

- Intro to **GitHub Copilot** for syntax hints.

- Demonstration of AI as an *assistant*, not an *answer machine*.

- Creating a simple Python program that adds two numbers stored in different variables and displays the result.

- Covering beginner-level functionalities in Python, including if-else statements, functions, classes, and objects.

Month 2 - Python Data Structures

1. Deep dive into Python data structures:

- Lists, Tuples, Dictionaries, and Sets — creation, modification, iteration.

- Begin problem-solving exercises with these data structures.

- Introduction to data structures and their applications.

2. AI Integration:

- Using **Cursor AI** to generate test cases and example datasets..

- Comparing AI-generated solutions with manual solutions.

- **Critical Skill:** Avoid AI for first attempt → then compare & optimize.

Month 3 - Linear Data Structures

- Arrays, Stacks, Queues, Linked Lists.

- Real-world applications.

- AI integration: ChatGPT for pseudocode generation, AI-assisted debugging walkthrough.

- Practice problem-solving questions focused on linear data

Month 4 - Non-Linear Data Structures

- Trees, Graphs (BFS, DFS basics)
- Hands-on coding challenges.
- AI integration: Using AI to visualize graphs, AI as a partner for edge case discovery.
- Problem-solving exercises with non-linear data structures.

Month 5 – Algorithm Foundations

- Asymptotic Notation, Time & Space Complexity.
- Covering foundational topics like Asymptotic Notation, Master Theorem, and Divide and Conquer principles.
- Studying various sorting algorithms like Bubble, Merge, Quick, etc.
- AI integration: Copilot for boilerplate, manual optimization by student.

Month 6 - Advanced Algorithms

- Continuation with advanced algorithm types:

- Greedy algorithms and Dynamic Programming techniques.

- Further development of algorithmic problem-solving skills.

- AI integration: Using AI for brainstorming alternative approaches, AI-generated solution review — identifying mistakes.

This structured plan ensures the student builds a solid foundation in programming fluently in Python., data structures, and algorithms, gradually advancing their skills over six months. Also make use of AI productively while maintaining independent problem-solving skills. Eventually, the student will be ready for advanced computer science studies or real-world projects.

—---