**Bài 2 :**

**5 nguyên tắc cá nhân khi sử dụng AI trong học tập lập trình**:

* **Đặt câu hỏi rõ ràng và cụ thể :** Giúp AI hiểu đúng vấn đề để trả lời chính xác.

**VD:** Thay vì hỏi “Giải thích hàm này”, tôi hỏi “Giải thích hàm map() trong JavaScript hoạt động như thế nào với mảng?”

* **Không phụ thuộc hoàn toàn vào AI :** Giữ thói quen tự tìm hiểu và tự sửa lỗi.

**VD:** Khi code bị lỗi, tôi tự debug trước rồi mới hỏi AI để hiểu nguyên nhân sâu hơn

* **Viết lại code theo cách hiểu của bản thân :** Thay vì chỉ sao chép đoạn code mà AI tạo ra, hãy thử viết lại bằng ngôn ngữ hoặc cấu trúc khác. Cách này giúp bạn ghi nhớ lâu hơn và phát triển kỹ năng tư duy lập trình.

**VD :** Sau khi AI cho ví dụ về vòng lặp for, tôi tự viết lại bằng while để rèn luyện khả năng chuyển đổi cấu trúc.

* **Hỏi AI để hiểu sâu từng phần trong code:** Khi gặp đoạn code chưa hiểu, đừng chỉ hỏi “Code này làm gì?”, mà hãy yêu cầu AI giải thích từng dòng, từng biến, hoặc cách hoạt động của một hàm cụ thể.

**VD:** Tôi hỏi “Tại sao cần dùng & trong scanf("%d", &n);?” và AI giúp tôi hiểu rằng scanf cần địa chỉ ô nhớ để lưu giá trị.

* **Dùng AI để tối ưu và phát hiện lỗi logic :** AI có thể giúp phân tích hiệu suất, chỉ ra lỗi logic khó thấy, hoặc gợi ý cách cải thiện thuật toán. Tuy nhiên, cần kiểm tra kỹ lại trước khi áp dụng.

**VD:** Tôi nhờ AI kiểm tra chương trình sắp xếp mảng, AI gợi ý dùng qsort() để thay cho thuật toán bubble sort chậm hơn.