**21120542 – Lâm Hoàng Quốc**

**Tuần 11 – Bài 01:**

1. **Vai trò của con trỏ (pointer):**

* Con trỏ của lớp cơ sở có thể giữ địa chỉ của lớp dẫn xuất.
* Điều này tạo nên tính đa hình khi một mảng các con trỏ của lớp dẫn xuất có thể giữ địa chỉ và đặc điểm của nhiều đối tượng lớp dẫn xuất khác nhau.
* Từ đó giảm thiểu các thao tác xử lý vì ta không cần phải tạo nhiều mảng có nhiều kiểu dữ liệu là các lớp dẫn xuất khác nhau nữa mà chỉ cần một mảng sử dụng đa hình duy nhất.
* Giúp giảm chi phí bộ nhớ, thao tác và thời gian thực thi.

1. **Vai trò của kế thừa (inheritance):**

* **Tính kế thừa** cho phép xây dựng một lớp mới dựa trên các định nghĩa của lớp đã có.
* Có nghĩa là lớp cơ sở có thể chia sẻ dữ liệu và phương thức cho các lớp dẫn xuất. Các dẫn xuất con không cần phải định nghĩa lại các dữ liệu và phương thức đã được chia sẻ (kế thừa) từ lớp cơ sở, ngoài ra có thể mở rộng các thành phần kế thừa và bổ sung thêm các thành phần mới.
* Giúp tái sử dụng mã nguồn 1 cách tối ưu, tận dụng được mã nguồn.

1. **Vai trò của hàm ảo (virtual):**

* Hàm ảo (virtual function) là một hàm thành viên trong lớp cơ sở mà lớp dẫn xuất khi kế thừa cần **phải định nghĩa lại**.
* **Hàm ảo** được sử dụng trong lớp cơ sở khi cần đảm bảo hàm ảo đó sẽ được **định nghĩa lại** trong lớp dẫn xuất. Việc này rất cần thiết trong trường hợp con trỏ có kiểu là lớp cơ sở trỏ đến đối tượng của lớp dẫn xuất.
* **Hàm ảo** chỉ khác hàm thành phần thông thường khi được gọi từ **một con trỏ**. Sử dụng hàm ảo khi muốn con trỏ đang trỏ tới đối tượng của lớp nào thì hàm thành phần của lớp đó sẽ được gọi mà không xem xét đến kiểu của con trỏ.
* Trong tính đa hình, hàm ảo có vai trò quan trọng dùng để điều hướng và gọi đúng phương thức của lớp dẫn xuất cần sử dụng trong một mảng con trỏ lớp cơ sở chứa các lớp dẫn xuất khác nhau.