**Trần Đình Nhật Trí - 21120576**

Những vai trò của con trỏ (pointer), kế thừa (inheritance), và hàm ảo (virtual) để tạo nên tính đa hình (polymorphism):

1. Con trỏ (Pointer): Con trỏ là một biến đặc biệt trong ngôn ngữ lập trình, được sử dụng để lưu trữ địa chỉ của một biến khác trong bộ nhớ. Con trỏ cho phép chúng ta truy cập và thao tác trực tiếp với các biến mà nó trỏ tới. Trong tính đa hình, con trỏ được sử dụng để tham chiếu đến các đối tượng thuộc các lớp khác nhau những có quan hệ kế thừa.
2. Kế thừa (Inheritance): Cho phép chúng ta xây dựng một lớp con dựa trên một lớp cha. Khi một lớp con kế thừa từ một lớp cha, nó sẽ kế thừa tất cả các thuộc tính và phương thức của lớp cha đó. Kế thừa cho phép chúng ta tạo ra các lớp con có thể mở rộng hoặc sửa đổi hành vi của lớp cha. Trong ngữ cảnh tính đa hình, kế thừa cho phép chúng ta truy cập và sử dụng các phương thức và thuộc tính của lớp cha thông qua con trỏ.
3. Hàm ảo (Virtual): Hàm ảo là một phương thức được khai báo trong lớp cha với từ khóa "virtual". Khi một phương thức được khai báo là hàm ảo, các lớp con có thể ghi đè (override) lại phương thức này để cung cấp một hành vi khác nhau. Hàm ảo cho phép chúng ta gọi phương thức của đối tượng dựa trên kiểu tham chiếu của con trỏ, trong khi thực tế nó có thể đang trỏ tới một đối tượng thuộc lớp con.