Cả ba khái niệm con trỏ (pointer), kế thừa (inheritance) và hàm ảo (virtual) đều có vai trò quan trọng trong việc tạo nên tính đa hình trong lập trình hướng đối tượng. Dưới đây là mô tả về vai trò của từng khái niệm và cách chúng tạo nên tính đa hình:

Con trỏ (Pointer):

Con trỏ là một biến đặc biệt trong ngôn ngữ lập trình, chứa địa chỉ của một vùng nhớ. Con trỏ cho phép chúng ta truy cập và thao tác trên dữ liệu một cách trực tiếp thông qua địa chỉ của nó. Trong việc tạo nên tính đa hình, con trỏ có thể được sử dụng để tham chiếu đến các đối tượng của các lớp khác nhau trong một cấu trúc kế thừa. Bằng cách sử dụng con trỏ, chúng ta có thể gọi các phương thức và truy cập các thành viên của một đối tượng thông qua địa chỉ của nó, mà không cần biết chính xác kiểu đối tượng. Kế thừa (Inheritance):

Kế thừa là một khái niệm quan trọng trong lập trình hướng đối tượng, cho phép xây dựng một lớp mới (lớp con) dựa trên một lớp hiện có (lớp cha). Lớp con có thể kế thừa các thuộc tính và phương thức từ lớp cha mà không cần phải triển khai lại chúng. Kế thừa cho phép lớp con mở rộng và thay đổi hoặc bổ sung các thuộc tính và phương thức của lớp cha. Tính đa hình được tạo ra thông qua kế thừa bằng cách sử dụng con trỏ hoặc tham chiếu đến lớp cha, nhưng gọi các phương thức của lớp con. Điều này cho phép gọi các phương thức đúng của đối tượng dựa trên kiểu đối tượng được tham chiếu. Hàm ảo (Virtual):

Hàm ảo là một phương thức trong lớp cơ sở mà có thể được ghi đè bởi lớp con. Bằng cách khai báo một hàm trong lớp cơ sở là "hàm ảo", các lớp con có thể triển khai lại hàm đó theo cách riêng của chúng. Hàm ảo cho phép gọi các phương thức của lớp con thông qua con trỏ hoặc tham chiếu của lớp cơ sở. Tính đa hình được thể hiện khi một con trỏ hoặc tham chiếu của lớp cơ sở gọi một hàm ảo, và tùy thuộc vào kiểu đối tượng thực sự được tham chiếu, phương thức tương ứng của lớp con sẽ được gọi. Tổ hợp của con trỏ, kế thừa và hàm ảo cho phép tạo nên tính đa hình trong lập trình hướng đối tượng. Tính đa hình cho phép chúng ta gọi các phương thức của các đối tượng khác nhau thông qua cùng một giao diện chung, cung cấp tính linh hoạt và sự mở rộng trong việc xử lý các đối tượng khác nhau trong cùng một hệ thống.