1. Các tính chất quan trọng của hướng đối tượng (OOP)

Encapsulation (Đóng gói):

Đóng gói dữ liệu (thuộc tính) và hành vi (phương thức) vào bên trong class.

Hạn chế truy cập trực tiếp từ bên ngoài và chỉ cho phép truy cập thông qua getter/setter.

Inheritance (Kế thừa):

Cho phép một class con kế thừa lại thuộc tính và phương thức của class cha.

Tái sử dụng mã nguồn, giảm thiểu lặp code.

Polymorphism (Đa hình):

Một phương thức có thể hoạt động khác nhau tùy theo đối tượng gọi nó.

Có hai loại đa hình:

Overloading (Nạp chồng): Cùng tên phương thức nhưng khác tham số.

Overriding (Ghi đè): Phương thức của class con ghi đè phương thức của class cha.

Abstraction (Trừu tượng hóa):

Chỉ thể hiện những gì cần thiết và che giấu chi tiết phức tạp.

Sử dụng abstract class và interface để tạo ra các khung chương trình.

2. Access Modifier trong Java

Public: cho phép truy cập bất cứ đâu trong chương trình

Protected: truy cập tuè các subclass của class đó và trong package

Default: truy cập từ cùng pakage

Private: cho phép truy cập bên trong class

3. class và instance

Class là một lớp để tạo ra các đối tượng

Instance là đối tượng tọa ra từ class

4. Phân biệt Abstract và Interface

Abstract có thể chứa cả phương thức có và không triển khai

Cho phép khai bòa biến instance và phương thức không trừu tượng

Interface chỉ chứa các phương thức có thân

Không thể chứa biến instance(chỉ chứa final static)

Nếu hai phương thức trùng tên và khác kiểu trả về → Lỗi biên dịch (compile-time error).

Nếu hai phương thức trùng tên và cùng kiểu trả về:

Class phải override lại phương thức đó để tránh mơ hồ.

5. Phân biệt Overriding và Overloading

Overriding (Ghi đè):

Lớp con ghi đè lại phương thức của lớp cha để thay đổi hành vi.

Phải có cùng tên, kiểu trả về và tham số.

Overloading (Nạp chồng):

Một lớp có nhiều phương thức cùng tên nhưng khác nhau về số lượng hoặc kiểu tham số.

6. Private và Static Method có Overriding được không?

Private method: Không thể overriding vì nó chỉ thuộc phạm vi trong class đó.

Static method: Không thể overriding, nhưng có thể hides (ẩn phương thức) nếu khai báo lại cùng tên ở lớp con.

7. Phương thức final có thể kế thừa được không?

Final method: Không thể override trong lớp con.

Nếu một phương thức được khai báo là final, lớp con sẽ không được phép ghi đè lại.

8. Phân biệt this và super

this:

Dùng để tham chiếu đến đối tượng hiện tại.

Gọi phương thức hoặc biến của class hiện tại.

Ví dụ: this.name = "Honda";

super:

Dùng để tham chiếu đến lớp cha của đối tượng.

Gọi phương thức hoặc constructor của lớp cha.

Ví dụ: super.sound();