

Section 16.

Question 1: What are the advantages of Polymorphism?

- Tính đa hình (Polymorphism) là một trong những đặc điểm cơ bản của OOP, cho phép các đối tượng thuộc các lớp khác nhau có thể phản ứng khác nhau với cùng một phương thức hoặc cùng một thao tác.
- Lợi ích bao gồm:
 - o Tái sử dụng code.
 - o Đồng nhất cho tên một hành vi, một phương thức cho các lớp để thuận tiện cho việc chỉnh sửa và bảo trì code => Tăng tính linh hoạt cho code.
 - o Mở rộng dễ hơn.

Question 2: How is Inheritance useful to achieve Polymorphism in Java?

- Tính kế thừa (Inheritance) là một tính năng trong lập trình hướng đối tượng cho phép một lớp (lớp con) kế thừa các thuộc tính và phương thức từ một lớp khác (lớp cha). Kế thừa giúp tái sử dụng mã, giảm sự lặp lại, và tạo ra các mối quan hệ giữa các lớp, đồng thời cũng tạo điều kiện cho việc mở rộng chức năng của các lớp con mà không cần phải thay đổi mã nguồn của lớp cha.
- Ví dụ: Thay vì mỗi lớp Dog và Cat thêm display() thì khi kế thừa thì ta không cần làm thế mà nó đã có sẵn rồi, và giữa chúng có mối quan hệ là cùng là thuộc kiểu Animal

```
lớp Animal {  
    String name;  
    Animal(String name) {  
        this.name = name;  
    }  
    void display() {  
        System.out.println(name);  
    }  
}
```

```

lớp Dog extends Animal {
    Dog(String name) {
        super(name);
    }
}

```

```

lớp Cat extends Animal {
    Cat(String name) {
        super(name);
    }
}

```

```

public lớp Main {
    public static void main(String[] args) {
        Dog dog = new Dog("Dog");
        Cat cat = new Cat("Cat");
        dog.display(); // Giảm lặp lại mã, sử dụng phương thức từ lớp cha
        cat.display();
    }
}

```

Question 3: What are the differences between Polymorphism and Inheritance in Java?

Bảng thống kê các điểm khác nhau cơ bản

Tiêu chí	Kế thừa	Đa hình
Mục đích	Tái sử dụng mã, tạo mối quan hệ cha-con	Cho phép một phương thức có hành vi khác nhau
Tác động lên phương thức	Lớp con kế thừa hoặc ghi đè phương thức từ lớp cha	Lớp con ghi đè phương thức của lớp cha để thay đổi hành vi

Tiêu chí	Kế thừa	Đa hình
Mối quan hệ	Tạo ra mối quan hệ "is-a" (là một)	Tạo ra mối quan hệ "can do" (có thể làm)
Chế độ truy cập	Lớp con có thể truy cập các thuộc tính và phương thức của lớp cha (public, protected)	Được gọi thông qua kiểu lớp cha nhưng thực thi hành vi của lớp con
Sử dụng trong OOP	Để mở rộng và tái sử dụng mã của lớp cha	Để thay đổi hành vi của phương thức tùy theo đối tượng thực tế