1. What are the advantages of Polymorphism?

(Lợi ích của Đa hình trong Java)

- Code reuse: Bạn có thể viết code tổng quát xử lý nhiều loại đối tượng khác nhau mà không cần sửa đổi.
- Code extensibility: Khi thêm một class mới (kế thừa từ class cũ), không cần thay đổi code đang chạy.
- Interface implementation: Cho phép các đối tượng từ các class khác nhau có thể "hành xử" giống nhau qua interface.
- Reduced complexity: Code đơn giản hơn, dễ đọc và dễ bảo trì.
- **Dynamic behavior**: Tại runtime, quyết định phương thức nào được gọi dựa trên kiểu thực tế của đối tượng.
- → Ví dụ: một play() cho CD và DVD đều gọi đúng phương thức tương ứng.

2. How is Inheritance useful to achieve Polymorphism in Java? (Kế thừa hỗ trợ Đa hình như thế nào?)

- Inheritance cho phép một class con kế thừa (extends) một class cha.
- Khi bạn khai báo một biến kiểu class cha nhưng gán nó với đối tượng class con, bạn có thể dùng polymorphism để gọi đúng phương thức override.
- Bản chất là: Kế thừa tạo ra mối quan hệ is-a ("là một loại"), và nhờ đó
 Java mới cho phép Đa hình xảy ra.

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

(Phân biệt giữa Đa hình và Kế thừa)

Tiêu chí	Inheritance (Kế thừa)	Polymorphism (Đa hình)
Ý nghĩa	Một class kế thừa thuộc tính và hành vi từ một class khác.	Một đối tượng có thể có nhiều hình thức khác nhau.
Mục đích	Tái sử dụng code, xây dựng hệ thống phân cấp	Cho phép đối tượng có hành vi khác nhau khi gọi cùng 1 phương thức
Cơ chế	Dùng extends, implements	Dùng override method, interface
Khi nào?	Khi muốn reuse và mở rộng chức năng có sẵn	Khi muốn các đối tượng khác nhau cùng chia sẻ hành vi chung nhưng thực hiện theo cách riêng
Ví dụ	class DVD extends Media	media.play(); chạy theo đúng loại DVD, CD, Book