1. Advantages of Polymorphism

Polymorphism is a fundamental concept in object-oriented programming (OOP) that allows objects of different classes to be treated as objects of a common superclass or interface. Its key advantages include:

- **Code Reusability & Flexibility**: Polymorphism enables a single method or interface to work with multiple object types.
- **Simplified Code Maintenance**: By reducing the need for lengthy if-else or switch-case statements to handle different object types, polymorphism makes code cleaner and easier to modify.
- **Extensibility**: New subclasses can be introduced without altering existing polymorphic code. For instance, adding a new Triangle class doesn't require changes to a method that accepts Shape objects.
- **Runtime Dynamic Binding**: Java resolves method calls at runtime (dynamic polymorphism), allowing the correct overridden method to execute based on the actual object type, not just the reference type.
- **Interface-Based Design**: Polymorphism promotes coding to interfaces (e.g., List can be ArrayList or LinkedList), making systems more modular and adaptable to change.

---

2. How Inheritance Enables Polymorphism in Java

Inheritance establishes an "is-a" relationship between classes, forming the foundation for polymorphism:

- **Method Overriding**: A subclass can override a superclass method. When a superclass reference, calls the method, the JVM invokes the subclass version at runtime.
- **Upcasting**: Assigning a subclass object to a superclass reference allows polymorphic usage. The superclass reference can interact with any subclass object uniformly.
- **Abstract Classes/Interfaces**: These define contracts that multiple classes implement differently.
- **Dynamic Method Dispatch**: Java's JVM decides which overridden method to execute at runtime, enabling flexible behavior without modifying existing code.

---

3. Polymorphism vs. Inheritance

Inheritance focuses on **code reuse** through hierarchical relationships. It’s static (compile-time) and defines "what a class is."

Polymorphism focuses on **behavior flexibility** by leveraging inheritance. It’s dynamic (runtime) and defines "how an object behaves" through:

- **Method Overriding** (runtime polymorphism).
- **Method Overloading** (compile-time polymorphism).

Key Difference: Inheritance establishes relationships, while polymorphism exploits them for adaptable behavior.