- What are the advantages of Polymorphism?

+ Polymorphism allows you to write code that works on the superclass but can be applied to all of its subclasses. This reduces the need to duplicate similar logic for different types
+ With polymorphism, you can easily introduce new types (such as new subclasses) without modifying existing code. The existing code will still function correctly with the new types.

+ With polymorphism, changes made in a subclass's implementation (e.g., modifying how toString() works) do not affect the general program structure, thus making the code easier to maintain.

- How is Inheritance useful to achieve Polymorphism in Java?

+ Because it allows one class to inherit properties and methods from another. When a subclass inherits from a superclass, it can override methods in the superclass.

+ Inheritance establishes a "parent-child" relationship between classes, and polymorphism allows objects of child classes to be treated as objects of the parent class, ensuring that the appropriate method from the child class is executed when the parent class's method is invoked.

- What are the differences between Polymorphism and Inheritance in Java?

Inheritance:
- A mechanism where one class (child class) ***inherits*** properties and methods from another class (parent class).

- Enables hierarchical relationships between classes and allows child classes to inherit functionality from the parent class.

Polymorphism:

- Allows objects of different subclasses to be treated as objects of a common superclass.

- Provides flexibility and extensibility by allowing a method or operation to behave differently based on the object type that is calling it.

➔ **Inheritance** is the act of "child" class reuses code from the "parent" class, while **polymorphism** is about dynamically deciding which method to invoke at runtime based on the object's actual class, not the type of reference.