1. What are advantages of Polymorphism

- **Code Reusability:** Polymorphism provides the reuse of code, as methods with the same name can be used in different classes.
- **Flexibility**: Polymorphism allows for more flexible and dynamic code, where the behaviour of an object can change at runtime depending on the context in which it is being used.
- **Code Simplicity:** It can reduce the complexity of code by allowing the use of the same method name for related functionality, making the code easier to read and maintain.
- **Better Organization:** Polymorphism allows for better organization of code by grouping related functionality in one class.
- **Extensibility:** Polymorphism enables code extensibility, as new subclasses can be created to extend the functionality of the superclass without modifying the existing code.
- **Increased Efficiency:** Compile-time polymorphism can lead to more efficient coding. The compiler can choose the appropriate method to call at compile time, based on the number, types, and order of arguments passed to it.

2. How is Inheritance useful to achieve Polymorphism in Java?

- inheritance in Java enables polymorphism by facilitating method overriding, allowing base class references to be used to refer to subclass objects, supporting interface inheritance, and promoting code reusability through inheritance and polymorphic behavior.

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

Inheritance	**Polymorphism**
Derived class inherits features from base class	Same base class but defined in multiple forms
Apply to class	Apply to methods
Support code reusability	Allows the obj to decide which form of the function to implement at compile-time and run-time