READING ASSIGNMENT

What are the advantages of Polymorphism

1. Code Reusability

Polymorphism allows methods in subclasses to override methods in their superclass, enabling code reuse and maintaining a consistent interface across related classes.

2. Flexibility and Extensibility

Polymorphism allows subclasses to provide their own implementations of methods defined in the superclass, making it easier to extend and customize behavior without modifying existing code.

3. Dynamic Method Invocation:

Polymorphism enables dynamic method invocation, where the method called is determined by the actual object type at runtime, providing flexibility in method dispatch.

4. Interface Implementation:

Interfaces in Java allow multiple classes to implement the same interface with their own implementations, facilitating polymorphic behavior and enabling objects of different classes to be treated interchangeably based on a common interface.

5. Method Overloading:

Polymorphism is also achieved through method overloading, where multiple methods with the same name but different parameter lists can be defined within a class or its subclasses, enhancing code readability and allowing flexibility in method invocation based on parameter types.

6. Reduced Code Complexity:

Polymorphism helps reduce code complexity by promoting a modular and hierarchical class structure, making it easier to understand, maintain, and extend large-scale software systems.

How is Inheritance useful to achieve Polymorphism in Java?

Inheritance enables polymorphism in Java by allowing one class to inherit the methods and structure of another. This allows objects of different subclasses to be treated as objects of a common superclass. When a method like play() is defined in the superclass and overridden in the subclasses, Java uses dynamic method dispatch to call the correct version of the method at runtime, based on the actual object type. This enables flexible and reusable code, which is a key benefit of object-oriented programming.

What are the differences between Polymorphism and Inheritance in Java?

S.NO	Inheritance	Polymorphism
1.	Inheritance is one in which a new class is created (derived class) that inherits the features from the already existing class(Base class).	Whereas polymorphism is that which can be defined in multiple forms.
2.	It is basically applied to classes.	Whereas it is basically applied to functions or methods.
3.	Inheritance supports the concept of reusability and reduces code length in object-oriented programming.	Polymorphism allows the object to decide which form of the function to implement at compile-time (overloading) as well as run-time (overriding).

S.NO	Inheritance	Polymorphism
4.	Inheritance can be single, hybrid, multiple, hierarchical and multilevel inheritance.	Whereas it can be compiled-time polymorphism (overload) as well as run-time polymorphism (overriding).
5.	It is used in pattern designing.	While it is also used in pattern designing.
6.	**Example :** The class bike can be inherit from the class of two-wheel vehicles, which is turn could be a subclass of vehicles.	**Example :** The class bike can have method name set_color(), which changes the bike's color based on the name of color you have entered.