Polymorphism

Monday, September 8, 2025 3:21 PM

◇ Definition of Polymorphism in Java

The word **Polymorphism** means *"many forms"*.
In Java, **Polymorphism** is the ability of an object to take on many forms.
It allows one **interface** to be used for a general class of actions, while the specific action is determined by the actual object at runtime.

👉 In simple words: **The same method or object can behave differently depending on the context.**

◇ Types of Polymorphism in Java

1. Compile-time Polymorphism (Static Polymorphism)

- Achieved using **Method Overloading**.
- The method to be executed is decided **at compile-time**.
- Example: same method name but different parameter lists.
-

```
// methods
public void favTask(){  2 usages  new *
    System.out.println(name + " favourite task is football") ;
}

// - overloading
public void favTask(String time){  1 usage  new *
    System.out.println(name + " favourite task is newspaper at " + time );
}
```

2. Runtime Polymorphism (Dynamic Polymorphism)

- Achieved using **Method Overriding**.
- The method call is resolved **at runtime** (dynamic binding).
- Example: A subclass provides a specific implementation of a method already defined in its superclass.
-

```
// - overriding
public void favTask(){  2 usages  new *
    System.out.println(name + " favourite task is Online gaming");
}
```

Note :

1. Dynamic object dispatch
    a. Means we can store a sub class object into a ref of a super class
    b. This is also a runtime polymorphism