

JAVA POLYMORPHISM

◆ What is Polymorphism?

Polymorphism means “**many forms**”. In Java, it allows **one interface to be used for a general class of actions**. It enables objects to behave differently based on their actual class, even if they're accessed through a reference of a parent class.

◆ Why Use Polymorphism?

- Code **reusability** and **flexibility**
- Enables **method overriding** and **dynamic method dispatch**
- Reduces code **complexity** by generalizing operations

◆ Types of Polymorphism in Java

Type	Explanation
Compile-time (Static)	Achieved by method overloading
Runtime (Dynamic)	Achieved by method overriding

✓ 1. Compile-time Polymorphism (Method Overloading)

Method Overloading: Multiple methods with the **same name** but **different parameters**.

```

java

class Calculator {
    int add(int a, int b) {
        return a + b;
    }

    double add(double a, double b) {
        return a + b;
    }
}

public class Main {
    public static void main(String[] args) {
        Calculator c = new Calculator();
        System.out.println(c.add(2, 3));           // Output: 5
        System.out.println(c.add(2.5, 3.5));       // Output: 6.0
    }
}

```

✓ 2. Runtime Polymorphism (Method Overriding)

Method Overriding: A **subclass** provides a specific implementation of a method that is already defined in its **superclass**.

```

class Animal {
    void sound() {
        System.out.println("Animal makes sound");
    }
}

class Dog extends Animal {
    void sound() {
        System.out.println("Dog barks");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal a = new Dog(); // Upcasting
        a.sound();             // Output: Dog barks
    }
}

```

◆ Key Points

- Method overloading = **compile-time**
- Method overriding = **runtime**
- Polymorphism enables **loose coupling** and **extensibility**

🎯 Final Tip: Think in Terms of Behavior, Not Class

Polymorphism is about **what an object can do**, not what it is. So focus on the **interface (behavior)** instead of the concrete class.