Overloading and Overriding

🎯 Direct Challenges

1. Overload a method add() with two and three parameters.

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

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

public class Main {
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("Sum of 2 numbers: " + calc.add(5, 10));
        System.out.println("Sum of 3 numbers: " + calc.add(3, 7, 2));
    }
}
```

Output:

```
Sum of 2 numbers: 15
Sum of 3 numbers: 12
```

2. Override toString() method of a custom class.

```
class Product {
    String name = "Laptop";
    int price = 50000;

    @Override
    public String toString() {
        return "Product[name=" + name + ", price=" + price + "]";
    }
}

public class Main {
    public static void main(String[] args) {
        Product p = new Product();
        System.out.println(p);
    }
}
```

Output:

```
Product[name=Laptop, price=50000]
```

3. Override a display() method from base class in child class.

```
class Animal {
    void display() {
        System.out.println("This is an animal.");
    }
}

class Dog extends Animal {
    @Override
    void display() {
        System.out.println("This is a dog.");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal a = new Dog();
        a.display();
    }
}
```

Output:

```
This is a dog.
```

🧩 Scenario-Based Challenges

1. Create a Logger class with overloaded log() methods for different data types.

```
class Logger {
    void log(String msg) {
        System.out.println("Log: " + msg);
    }

    void log(int number) {
        System.out.println("Log Number: " + number);
    }

    void log(boolean status) {
        System.out.println("Log Status: " + status);
    }
}

public class Main {
    public static void main(String[] args) {
        Logger log = new Logger();
        log.log("System started");
        log.log(404);
        log.log(true);
    }
}
```

Output:

```
Log: System started
Log Number: 404
Log Status: true
```

2. Build a Vehicle class with overridden move() in Car and Bike subclasses.

```
class Vehicle {
    void move() {
        System.out.println("Vehicle is moving...");
    }
}

class Car extends Vehicle {
    @Override
    void move() {
        System.out.println("Car is driving on the road.");
    }
}


class Bike extends Vehicle {
    @Override
    void move() {
        System.out.println("Bike is zooming through traffic.");
    }
}


public class Main {
    public static void main(String[] args) {
        Vehicle v1 = new Car();
        Vehicle v2 = new Bike();


        v1.move();
        v2.move();
    }
}
```

Output:

```
Car is driving on the road.
Bike is zooming through traffic.
```