Encapsulation and **Polymorphism** as two important ideas in Java OOP.

1. Encapsulation — "Protect the data"

Simple meaning:

Encapsulation means **keeping data private and accessing it through methods**.

Think of it like an **ATM**: you cannot directly access the bank's internal data. You use buttons/options to interact with it.

Example

```
class Student {

    private int age;

    public void setAge(int age) {
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

public class Main {
    public static void main(String[] args) {

        Student s = new Student();

        s.setAge(20);

        System.out.println(s.getAge());
    }
}
```

Output

```
20
```

Here:

- `private int age` → data is protected
- `setAge()` → used to set the value
- `getAge()` → used to get the value

Remember

Encapsulation = Data hiding + controlled access

---

2. Polymorphism — “One name, many forms”

Simple meaning:

Polymorphism means **the same method name can behave differently depending on the situation/object**.

There are two main types.

Method Overloading

Same method name, different parameters.

```
class Calculator {

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

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

Now:

```
Calculator c = new Calculator();

System.out.println(c.add(10, 20));
System.out.println(c.add(10, 20, 30));
```

Output:

```
30
60
```

Same method name `add()`, but different behavior based on the arguments.

Method Overriding

A child class changes the behavior of a method inherited from the parent.

```
class Animal {

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

class Dog extends Animal {

    @Override
    void sound() {
        System.out.println("Dog barks");
    }
}
public class Main {
    public static void main(String[] args) {

        Animal a = new Dog();

        a.sound();
    }
}
```

Output:
Dog barks
The reference is Animal, but the actual object is Dog, so the Dog version of sound() runs.

Easy way to remember

Concept	Simple meaning	Example
Encapsulation	Protect/hide data	private + getter/setter
Polymorphism	One name, different behavior	Method overloading/overriding

One-line memory trick

Encapsulation → "Hide the data."
Polymorphism → "Same method, different behavior."