

Advanced Programming

Object-Oriented Programming Introduction

Instructor: Ali Najimi

Author: Hossein Masihi

**Department of Computer Engineering
Sharif University of Technology**

Fall 2025





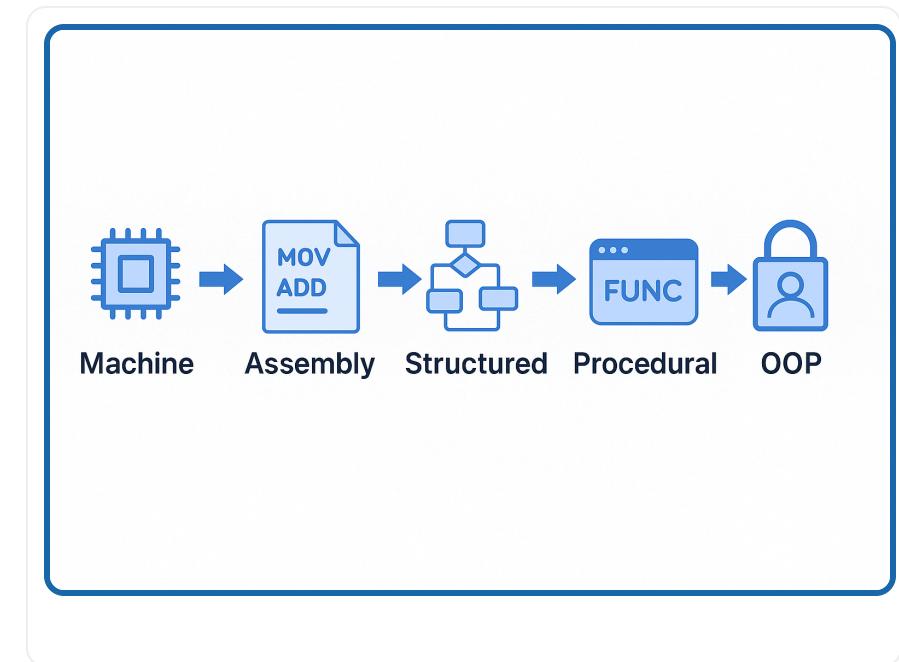
Table of Contents

1. History of Programming Paradigms
2. Structured Programming
3. Procedural Programming
4. Object-Oriented Programming (OOP)
5. Classes & Objects
6. Four Pillars of OOP
7. Advantages of OOP
8. Practice



History of Programming Paradigms

- Machine → Assembly → Structured → Procedural → OOP
- Each paradigm solved the growing complexity of software.
- OOP combines **data and behavior** into one logical unit.





Structured Programming

- Introduced **loops**, **functions**, and **modules**.
- Removed “spaghetti code”.
- Example:

```
for(int i = 0; i < 5; i++) {  
    printf("%d", i);  
}
```

Structured Programming



- functions

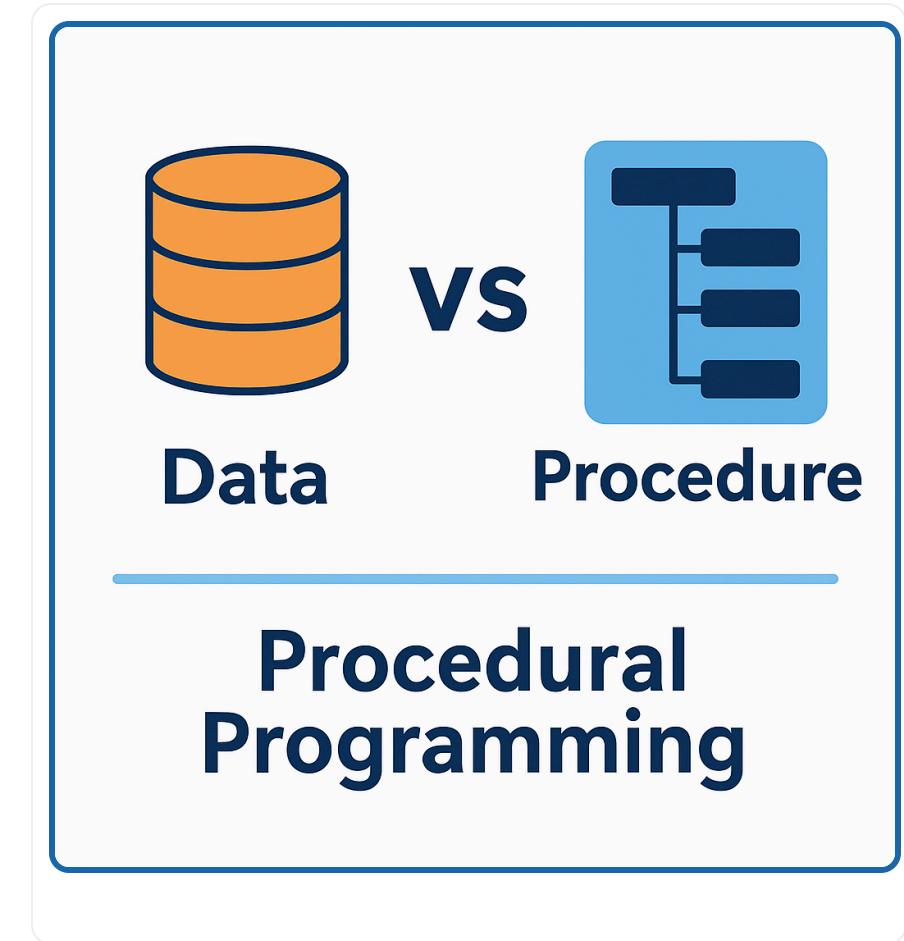
```
printf("%d", i);
```



Procedural Programming

- Uses **functions (procedures)** to organize logic.
- Data and logic are separate.
- Example:

```
void deposit(float amount) {  
    balance += amount;  
}
```

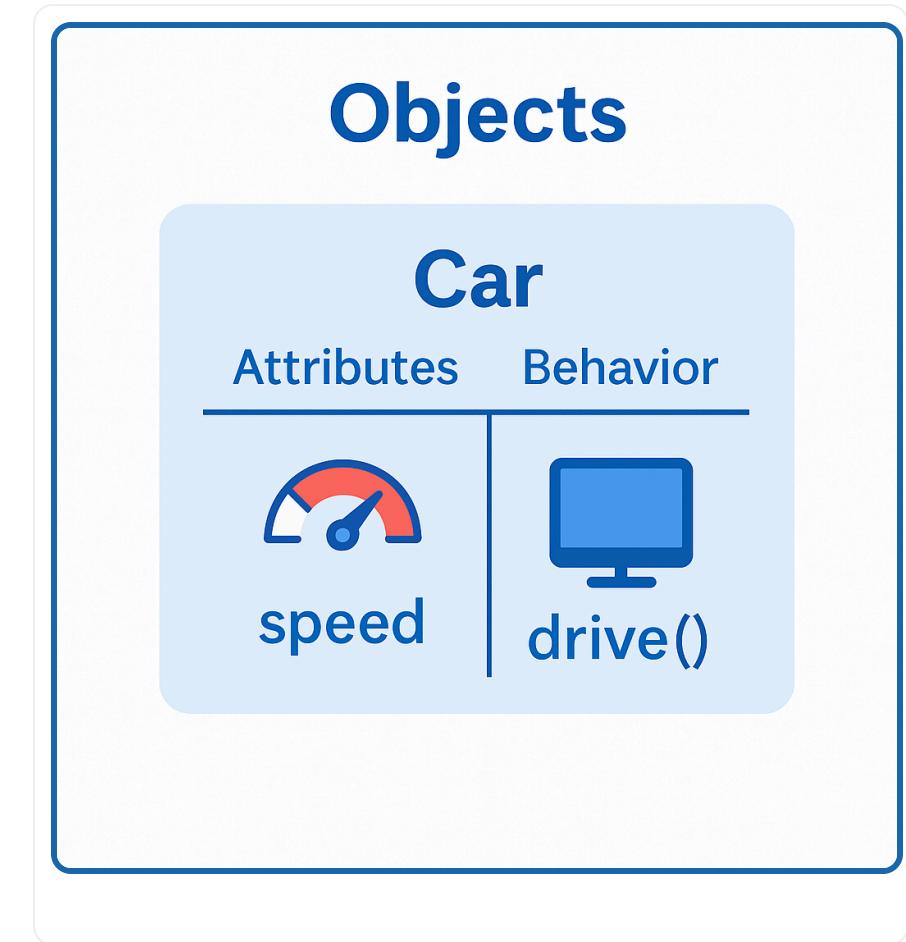




Object-Oriented Programming (OOP)

- Introduced in **Simula (1967)**, popularized by **C++** and **Java**.
- Models real-world entities as **objects**.
- Example:

```
class Car {  
    int speed;  
    void drive() {  
        System.out.println("Driving...");  
    }  
}
```

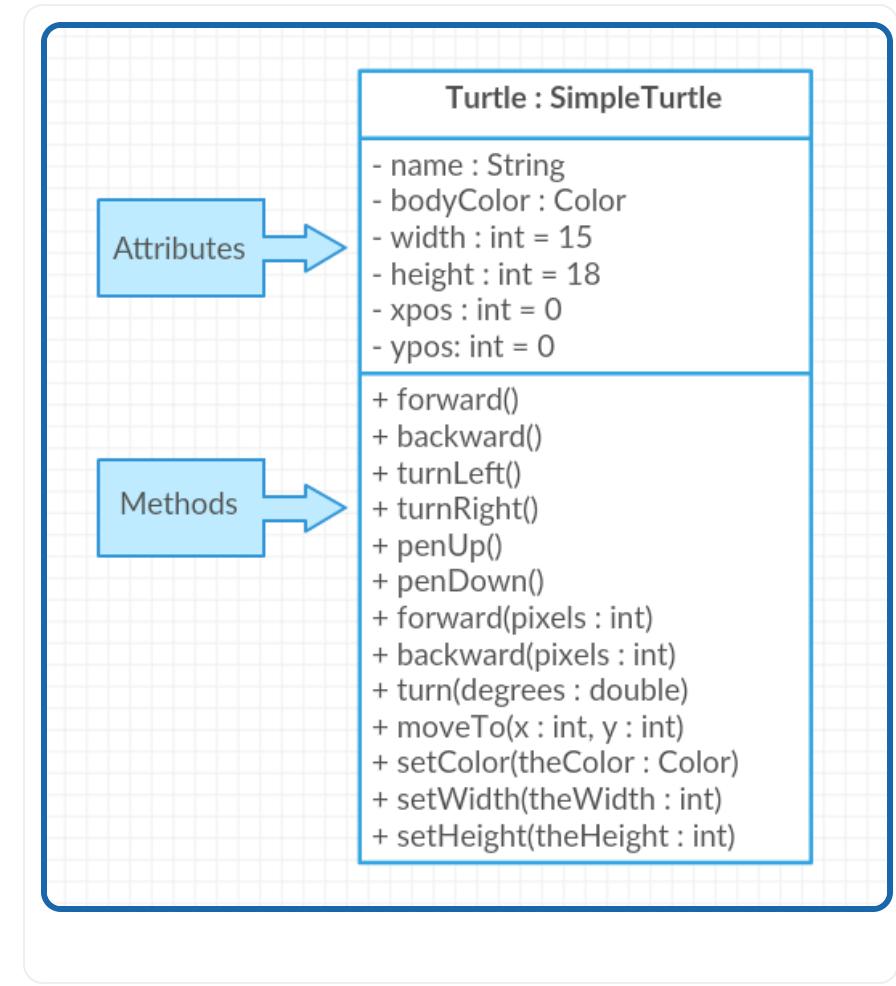




Classes and Objects

- **Class:** blueprint (e.g., Car, Student).
- **Object:** instance of a class.
- **Properties:** Attributes
- **Procedures:** Methods

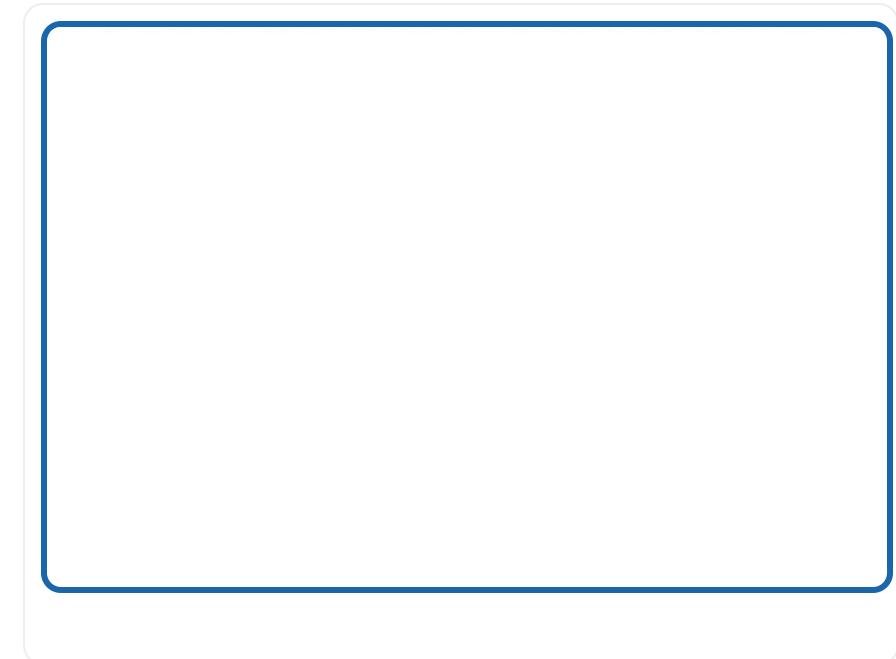
```
class Circle {  
    double radius;  
    double area() {  
        return Math.PI * radius * radius;  
    }  
}  
Circle c = new Circle();  
c.radius = 5;  
System.out.println(c.area());
```





Four Pillars of OOP

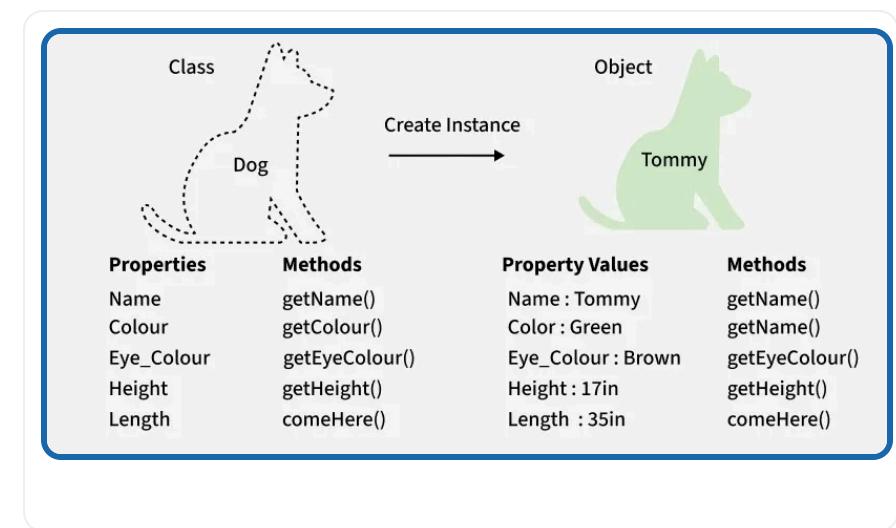
- 1. Encapsulation** – Hide internal data
- 2. Inheritance** – Reuse behavior
- 3. Polymorphism** – Many forms of one interface
- 4. Abstraction** – Show essentials, hide details





Classes and Objects: Example

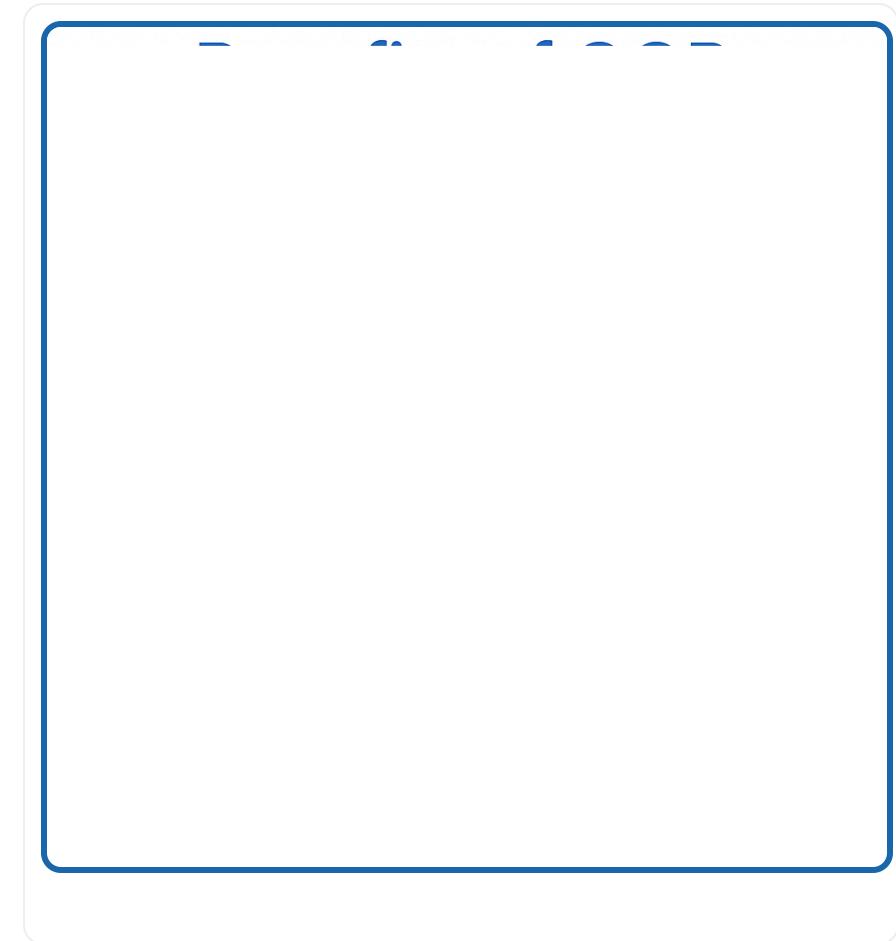
- Define Animal
- Define Dog
- Define Cat
- Define Table
- Define Chair





Advantages of OOP

- Improves reusability
- Supports modular design
- Easier debugging
- Enables teamwork
- Better mapping to real-world problems





Thank You

AP – OOP Introduction



I choose a lazy person to do a hard job. Because a lazy person will find an easy way to do it.

— Bill Gates —

AZ QUOTES