Java OOP Fundamentals

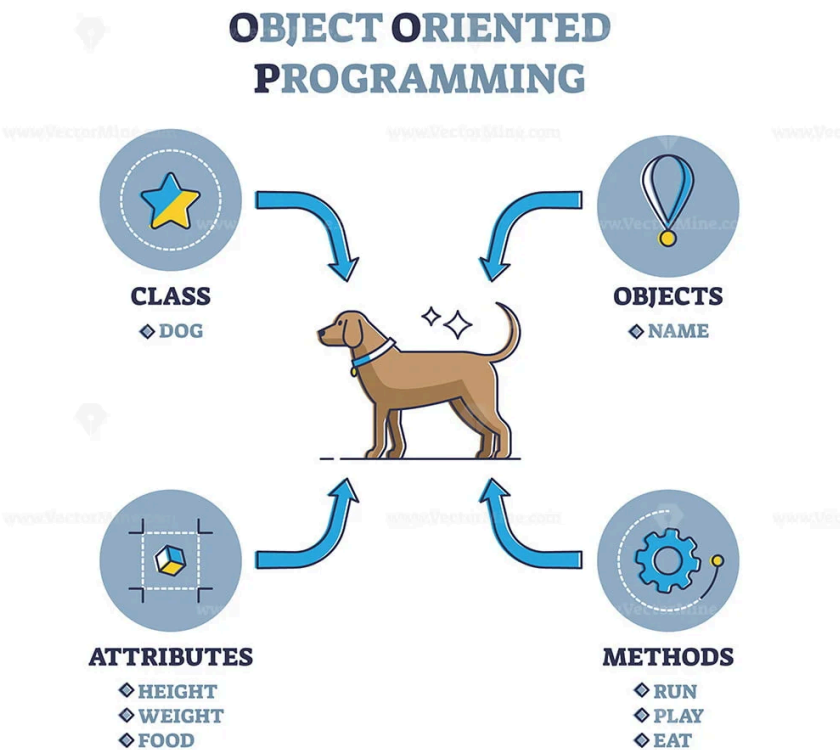
Press Space for next page →

Agenda

1. Java Object-Oriented Programming
2. Principles/Pillars of OOP
3. Class
4. Object
5. Constructor
6. Types of Constructors
7. Code Example of Default Constructor
8. Code Example of Parameterized Constructor

Java Object-Oriented Programming

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects".



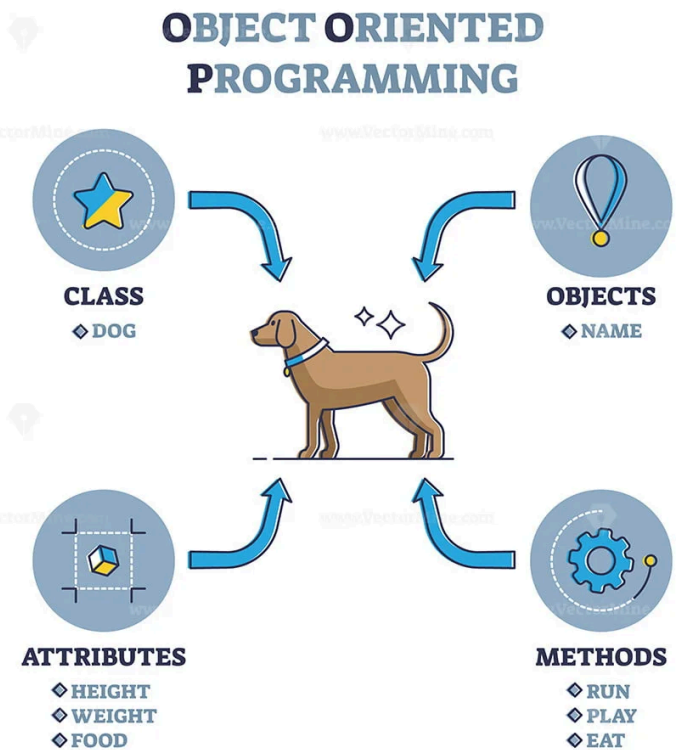
Learn more about Java OOP

Java Object-Oriented Programming

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects".

- Objects are instances of classes

Learn more about Java OOP

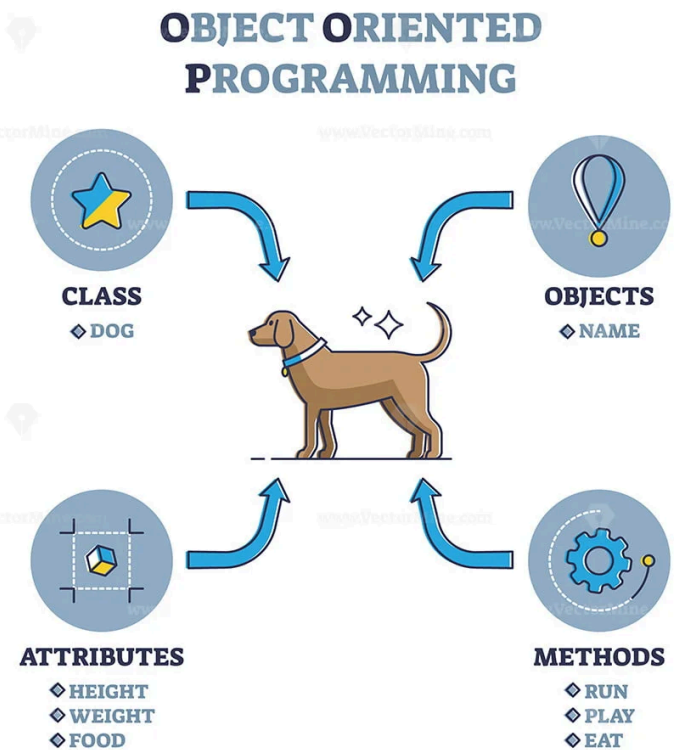


Java Object-Oriented Programming

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects".

- Objects are instances of classes
- Classes define the properties and behaviors of objects

Learn more about Java OOP

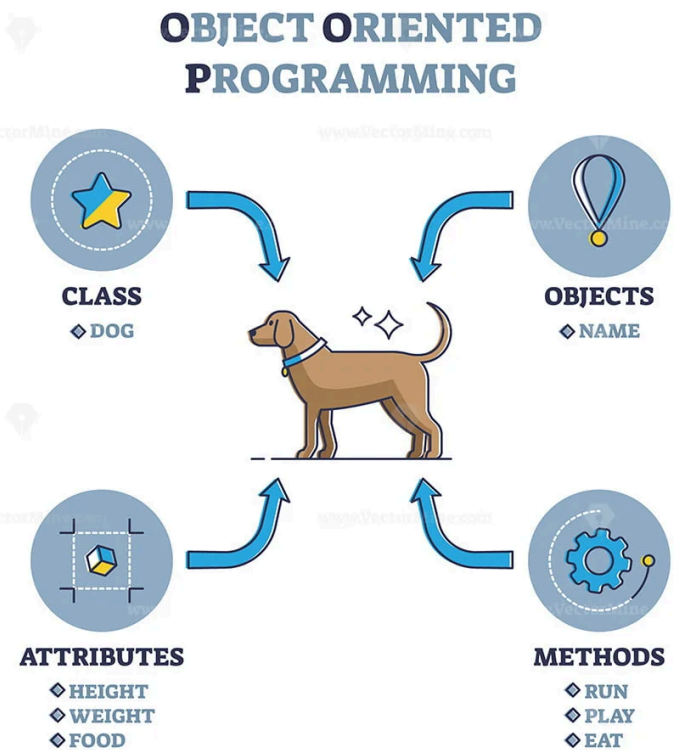


Java Object-Oriented Programming

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects".

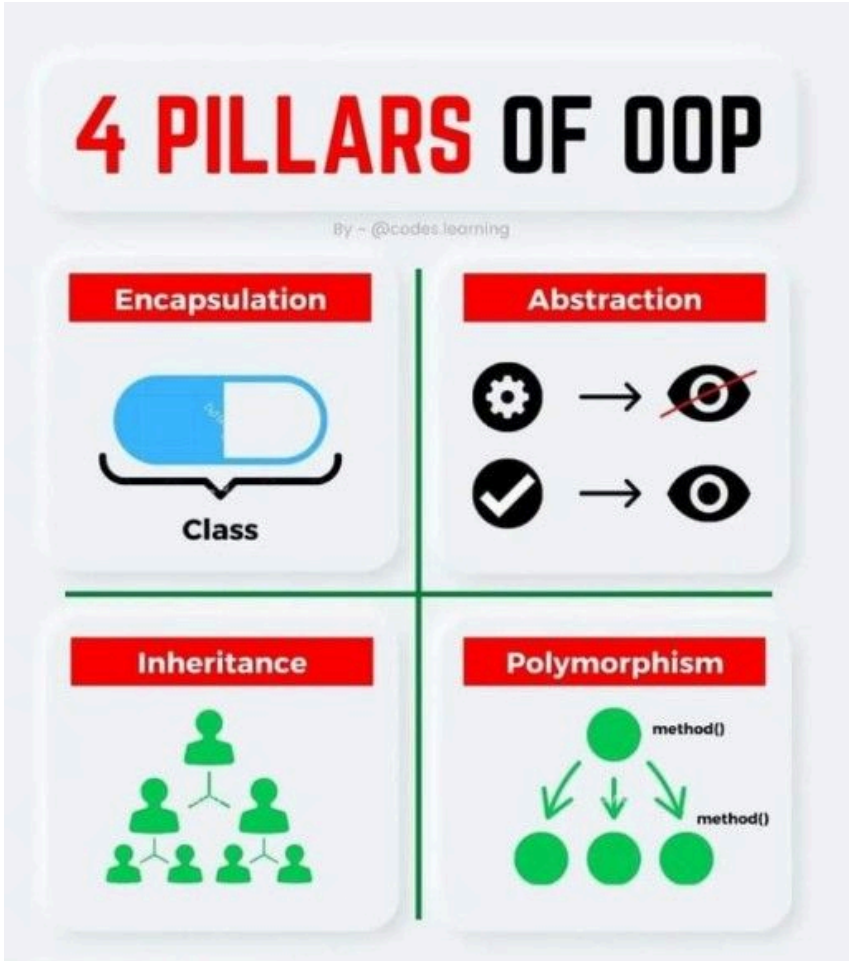
- Objects are instances of classes
- Classes define the properties and behaviors of objects
- Inheritance, Encapsulation, Abstraction , and polymorphism are the main concepts of OOP

Learn more about Java OOP



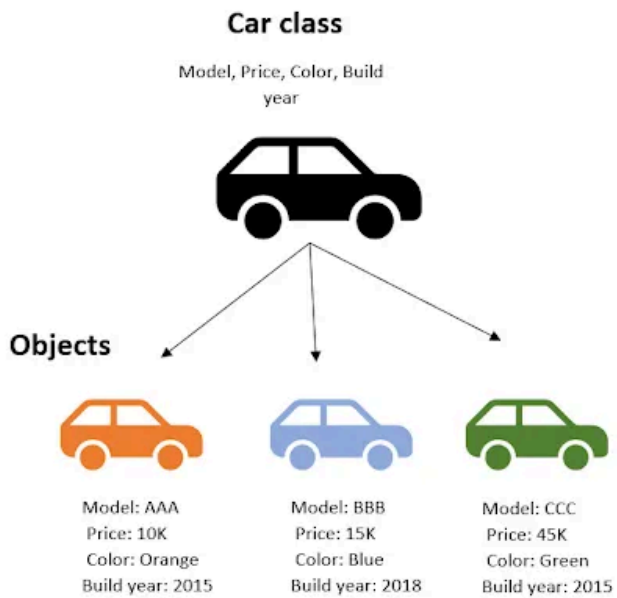
Principles/Pillars of OOP

- **Inheritance**: Ability to create a new class from an existing class.
- **Abstraction**: Hiding the implementation details and showing only the functionality.
- **Encapsulation**: Wrapping up of data and methods into a single unit and keeping some data hidden to protect the data.
- **Polymorphism**: Ability of an object to take many forms.



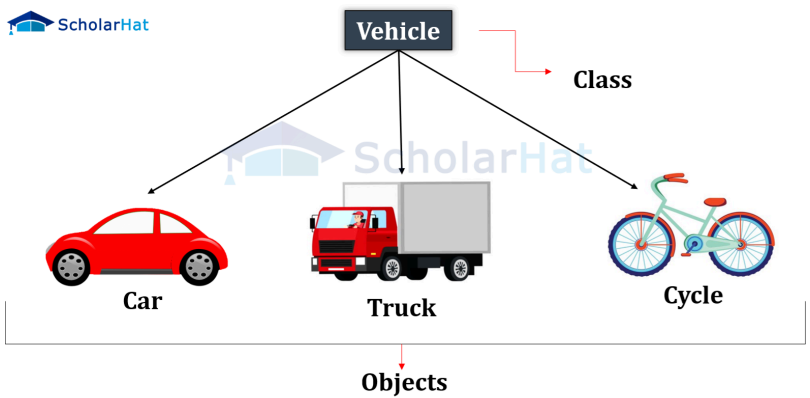
Class

- A class is a blueprint for objects that defines the properties and behaviors of objects.



Object

- Object can be defined as an instance of a class.
- An object has two characteristics:
  - State: represents the data (value) of an object.
  - Behavior: represents the behavior (functionality) of an object.
- For example, a chair, pen, table, keyboard, bike, etc.



Constructor

- In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created.
- Name of the constructor is the same as the class name.
- It is a special type of method which is used to initialize the object.
- Every time an object is created using the ***new*** keyword, at least one constructor is called.
- It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.

Types of Constructors

> There are two types of constructors in Java:

- **Default Constructor**: A constructor that has no parameter is known as a default constructor.
- **Parameterized Constructor**: A constructor that has parameters is known as a parameterized constructor.

Code Example of Default Constructor

```
class Student {
    int id;
    String name;

    //creating a default constructor
    Student() {
        id = 101;
        name = "John";
    }

    void display() {
        System.out.println(id + " " + name);
    }

    public static void main(String args[]) {
        Student s1 = new Student();
        s1.display();
    }
}
```

Code Example of Parameterized Constructor

```
class Student {
    int id;
    String name;

    //creating a parameterized constructor
    Student(int id, String name) {
        this.id = id;
        this.name = name;
    }

    void display() {
        System.out.println(id + " " + name);
    }

    public static void main(String args[]) {
        Student s1 = new Student(101, "John");
        Student s2 = new Student(102, "Smith");
        s1.display();
        s2.display();
    }
}
```

Thank you ❤️

qa-june-2024-automation-with-java-slides