



# OBJECT-ORIENTED PROGRAMMING & SOLID

Tomás González Martín  
Alberto Delgado Soler



# Index

---

1. OOP: Main concepts
2. What is OOP (Object-Oriented Programming)?
3. What is SOLID?
4. Bibliography

```
public class Vehicle {  
  
}
```

# 1. OOP: Main concepts

---

## 1.2 Object

Instance of a class.

```
Vehicle car = new Vehicle("Toyota");
```

# 1. OOP: Main concepts

---

## 1.3 Attributes

Characteristics of an object.

```
public class Vehicle {  
    private String brand_  
    private String model_;
```

# 1. OOP: Main concepts

---

## 1.4 Method

Algorithm associated with an object.

```
public double getMaximumSpeed() {  
    return maximumSpeed_;  
}
```

## 2. What is Object-Oriented Programming?

---

## 2. What is Object-Oriented Programming?

---

- Inheritance



## 2. What is Object-Oriented Programming?

---

- Inheritance
- Polymorphism

## 2. What is Object-Oriented Programming?

---

- Inheritance
- Cohesion
- Polymorphism

## 2. What is Object-Oriented Programming?

---

- Inheritance
- Cohesion
- Polymorphism
- Coupling

## 2. What is Object-Oriented Programming?

---

- Inheritance
- Cohesion
- Abstraction
- Polymorphism
- Coupling

## 2. What is Object-Oriented Programming?

---

- Inheritance
- Cohesion
- Abstraction
- Polymorphism
- Coupling
- Encapsulation

## 2. What is OOP?

---

- Inheritance
- Cohesion
- Abstraction

## Examples

- Polymorphism
- Coupling
- Encapsulation

## 2. What is OOP?

---

- Inheritance
- Cohesion
- Abstraction

## Examples

- Polymorphism
- Coupling
- Encapsulation

## 2. What is OOP?

---

- Inheritance
- Cohesion
- Abstraction

## Examples

- Polymorphism
- Coupling
- Encapsulation



## 2. What is OOP?

---

- Inheritance
- Cohesion
- Abstraction

## Examples

- Polymorphism
- Coupling
- Encapsulation

## 2. What is OOP?

---

- Inheritance
- Cohesion
- Abstraction

## Examples

- Polymorphism
- Coupling
- Encapsulation

## 2. What is OOP?

---

- Inheritance
- Cohesion
- Abstraction

## Examples

- Polymorphism
- Coupling
- Encapsulation

### 3. What is SOLID?

---

### 3. What is SOLID?

---

- Single responsibility principle

### 3. What is SOLID?

---

- Single responsibility principle
- Open-closed principle

### 3. What is SOLID?

---

- Single responsibility principle
- Open-closed principle
- Liskov substitution principle

### 3. What is SOLID?

---

- Single responsibility principle
- Open-closed principle
- Liskov substitution principle
- Interface segregation principle



### 3. What is SOLID?

---

- Single responsibility principle
- Open-closed principle
- Liskov substitution principle
- Interface segregation principle
- Dependency inversion principle

# Bibliography

---

- [https://en.wikipedia.org/wiki/Object-oriented\\_programming](https://en.wikipedia.org/wiki/Object-oriented_programming)
- <https://es.wikipedia.org/wiki/SOLID>
- [https://en.wikipedia.org/wiki/Abstraction\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Abstraction_(computer_science))
- [https://en.wikipedia.org/wiki/Inheritance\\_\(object-oriented\\_programming\)](https://en.wikipedia.org/wiki/Inheritance_(object-oriented_programming))
- [https://en.wikipedia.org/wiki/Encapsulation\\_\(computer\\_programming\)](https://en.wikipedia.org/wiki/Encapsulation_(computer_programming))
- [https://en.wikipedia.org/wiki/Cohesion\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Cohesion_(computer_science))
- [https://en.wikipedia.org/wiki/Coupling\\_\(computer\\_programming\)](https://en.wikipedia.org/wiki/Coupling_(computer_programming))
- [https://en.wikipedia.org/wiki/Encapsulation\\_\(computer\\_programming\)](https://en.wikipedia.org/wiki/Encapsulation_(computer_programming))
- <https://www.java.com/es/>