

---

# Table of Contents

Introduction	1.1
Let's Go!	1.2
S: Single Responsibility	1.3
O: Open/Closed	1.4
L: Liskov Substitution	1.5
I: Interface Segregation	1.6
D: Dependency Inversion	1.7

# **SOLID Design Principles In Common Lisp**

Learn how to apply SOLID design principles with Common Lisp and the powerful CLOS system.

# Let's Go!

## What is SOLID?

- Single Responsibility Principle
- Open/Closed Principle
- Liskov Substitution Principle
- Interface Segregation Principle
- Dependency Inversion Principle

## **S: Single Responsibility**

A class should have one, and only one, reason to change.

## **O: Open/Closed**

Software entities (classes, modules, functions, etc) should be open for extension, but closed for modification.

## L: Liskov Substitution

Let  $\Phi(x)$  be a property provable about objects  $x$  of type  $T$ . Then  $\Phi(y)$  should be true for objects  $y$  of type  $S$  where  $S$  is a subtype of  $T$ .

.

## **L: Interface Segregation**

Clients should not be forced to depend upon interfaces that they do not use.

## D: Dependency Inversion

- High level modules should not depend upon low level modules. Both should depend upon abstractions.
- Abstractions should not depend upon details. Details should depend upon abstractions.