

# SOLID PRINCIPLES IN C++

Explore C++ Design Goals, Common Smells, Process and the SOLID Principles and Best Practices for Modern Application Design.

## Target Audience:

C++ Developers

## Duration:

2 days

## Prerequisites:

This is an intermediate-level C++ development course geared for experienced C++ developers.

## Outcome:

Using C++ to demonstrate with extensive hands-on demos, this course teaches:

- Design Goals - What are we trying to accomplish?
- Design Smells - How to identify and objectively articulate bad design choices.
- Test Driven Development and minimalistic design and implementation - Avoid overdesign and facilitate design changes.
- Process - What are the best ways to proceed with software development and measure progress.
- Single Responsibility Principle - Only one reason to change
- Open/Close Principle - Open for business, closed for modification
- Liskov's Substitution Principle - a subtype can do more and require less
- Interface Segregation Principle - interfaces are for clients
- Dependency Inversion Principle - don't call us, we'll call you!

## Course Content

### Introduction

- Design
- Design Goals
- Process - Good, Bad, Ugly
- Design Smells
- Minimalistic Design and Test-Driven Development
- The Company Class
- The Employee Class

### Let's Get S.O.L.I.D.

- Single Responsibility Principle
- SRP Lab
- Open/Close Principle
- OCP Lab
- OCP Lab - Refactoring Employee
- OCP - Refactoring Company
- Liskov's Substitution Principle
- LSP Lab

- Interface Segregation Principle
- Interface Segregation Lab
- Dependency Inversion Principle
- DIP Lab