

# SOFTWARE DESIGN PATTERNS



**Goals:** Maximize the reusability and the correctness of the code by using consolidated patterns.

**Target:** Developers with a good knowledge of an object oriented language.

**Format:** 40 hrs. (40% Theory – 60% Coding)

**Notes :** The course could be taught both for Java or C++.

For most important patterns, aside the theory, example of code are provided and commented. Workshops could be arranged.

## SYLLABUS

### Introduction

#### Contents

- Recap of basic OO Principles
- Recap of UML notation
- General Software Design Pattern Purpose
- Design Pattern Classification

### Most Common Design Patterns

#### Contents

- Abstract Factory
- Singleton
- Composite
- Decorator
- Observer
- Strategy
- Template Method

### Other Design Patterns

#### Contents

- Chain of Responsibility
- Facade
- Iterator
- Proxy
- Adapter
- Prototype