

Course code:	C/OOBP
Course title:	Best practices of object-oriented programming in C++ language
Days:	3

Description:

Course intended for:

The training is offered to programmers, designers and architects, who would like to acquire, broaden or systematize their knowledge on the best practices, principles and patterns of object-oriented programming in C++.

Course objective:

The training objective is to get familiar with and to train the best practices, principles and patterns of object-oriented programming in C++, including:

- SOLID principles
- Selected GRASP patterns
- Selected patterns of the Gang of Four
- Selected idioms of programming in C++

Requirements:

The participants must be familiar with programming in C++.

Course parameters:

3*8 hours (3*7 net hours) of lectures and workshops (with a visible emphasis on workshops).

Group size: no more than 8-10 participants.

Course curriculum:



1. SOLID principles

- The Single Responsibility Principle
 - Introduction, definition, responsibilities
 - Problem-oriented workshop
 - Refactoring principles
 - Selected patterns of the Gang of Four (GoF) in the context of the principle
- The Open/Closed Principle
 - Introduction, definition, responsibilities
 - Problem-oriented workshop
 - Refactoring principles
 - Selected patterns of the Gang of Four (GoF) in the context of the principle
- The Liskov Substitution Principle
 - Introduction, definition, responsibilities
 - Problem-oriented workshop
 - Refactoring principles
 - Selected patterns of the Gang of Four (GoF) in the context of the principle
- The Interface Segregation Principle
 - Introduction, definition, responsibilities
 - Problem-oriented workshop
 - Refactoring principles
 - Selected patterns of the Gang of Four (GoF) in the context of the principle



- The Dependency Inversion Principle
 - Introduction, definition, responsibilities
 - Problem-oriented workshop
 - Refactoring principles
 - Selected patterns of the Gang of Four (GoF) in the context of the principle

2. Gang of Four (GoF) patterns

- Building patterns
 - Patterns: Singleton, Factory Method, Abstract Factory, Prototype, Builder
 - Problem-oriented workshop
- Structural patterns
 - Patterns: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy
 - Problem-oriented workshop
- Behavioral patterns
 - Patterns: Chain of Responsibility, Command, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor
 - Problem-oriented workshop

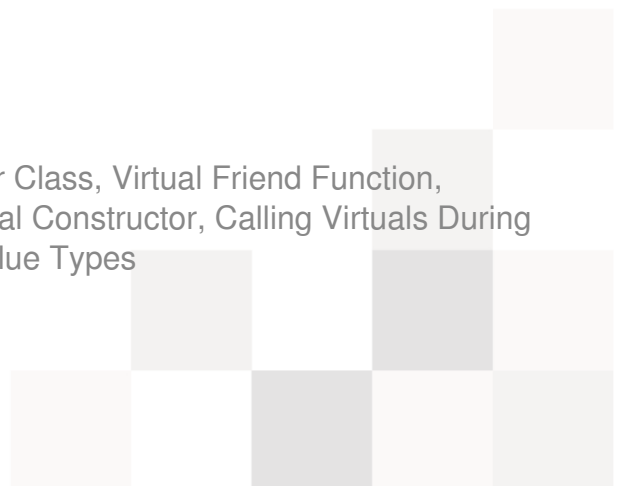
3. GRASP patterns

- Low Coupling, High Cohesion
- Information Expert, Creator, Controller
- Polymorphism, Pure Fabrication
- Indirection, Protected Variations
- Problem-oriented workshop



4. Idioms of programming in C++

- Management of resources
 - Idioms: Release Return, Move Constructor, Resource Acquisition Is Initialization, Scope Guard
 - Problem-oriented workshop
- Memory management
 - Idioms: Intrusive Reference Counting, Non-intrusive Reference Counting, Const auto_ptr, Checked Delete, Concrete Data Type, Copy and Swap
 - Problem-oriented workshop
- Optimization of memory and processing
 - Idioms: Shrink to Fit, Clear and Minimize, Non-throwing Swap , Erase-Remove, Boost Mutant, Computational Constructor, Copy on Write, Empty Base Optimization
 - Problem-oriented workshop
- Types and type security
 - Idioms: Type Safe Enum, Type Selection, Type Generator, Traits, Capability Query, Coercion by Member Template, Mixin from Above, Int to Type
 - Problem-oriented workshop
- Construction and initialization
 - Idioms: Construction Tracker, Construct on First Use, Base from Member, Runtime Static Initialization Idiom
 - Problem-oriented workshop
- Polymorphism
 - Idioms: Interface Class, Inner Class, Virtual Friend Function, Polymorphic Exception, Virtual Constructor, Calling Virtuals During Initialization, Polymorphic Value Types



- Problem-oriented workshop

