incusdata (pty) ltd. 1995/04149/07
Coding · Matters
PO Box 94
Raslouw, 0109
tel : +27 12 666 2020
fax: +27 12 666 2021
www.incusdata.com

**Price:** R11,900.00 excl. VAT
**Duration:** 5 days
**Code:** DSNPT

# Design Patterns

## Description

A design pattern is an optimised, reusable design solution to a programming problem that occurs often. It is not specific to a particular programming language. Design patterns improve code, because they provide a tested solution to a problem, and make it easier to maintain the code. This course will give you a solid foundation in design patterns: how to classify them, how to identify the right pattern, and when to use or avoid using a design pattern.

## Objectives

After you have completed the Design Patterns course, you will be able to:

- Identify and use design patterns correctly.
- Identify the wrong use of a design pattern.
- Understand the Gang of Four (GoF) design patterns.
- Identify and classify behavioral, creational and structural patterns.

## Intended Audience

You should attend the Design Patterns course if:

- You are an experienced programmer and want to learn more about design patterns.
- You are a systems architect and need to know more about design patterns and anti-patterns.

## Prerequisites

Before you attend the Design Patterns course:

- You must have experience programming in an object-oriented language such as Java, C++, C# or VB.NET.
- You must have attended our Object-Oriented Analysis and Design Course or have experience in OO analysis and design.

## Course Contents

### *Revision of OO Concepts and Methodologies.*

- MVC paradigm.
- Classes and objects.
- Attributes and behaviours.
- Data encapsulation.
- Polymorphism, overloading and overriding.
- Inheritance and interfacing.
- Composition and aggregation.
- OO methodologies.
- Iterative, incremental development.
- Inception, Elaboration, Construction, Transition phases.
- Robustness analysis.

### *UML Revision*

- Use case diagrams and use case text.
- Activity diagrams.

- Class and object diagrams.
- Sequence and communication diagrams.
- State and timing diagrams.
- Component and deployment diagrams.
- Package diagrams.
- Visual modelling tools.

***Introduction to Design Patterns***

- Design patterns history.
- Design patterns as proven solutions to common design problems.
- The Gang of Four (GoF) patterns.
- Creational, structural and behavioural classifications.
- Design patterns vs design principles.

***Components of a Design Pattern***

- Name, Intent, Motivation, Applicability, Structure, Participants, Collaborators, Consequences, Implementation.

***Creational Patterns***

- Abstract Factory.
- Builder.
- Factory.
- Prototype.
- Singleton.

***Structural Patterns***

- Adaptor.
- Bridge.
- Composite.
- Decorator.
- Facade.
- Flyweight.
- Proxy.

***Behavioural Patterns***

- Chain of Responsibility.
- Interpreter.
- Command.
- Iterator.
- Mediator.
- Memento.
- Observer.
- State.
- Strategy.
- Template method.
- Visitor.

***Additional Patterns***

- Model-View-Controller (MVC).
- Simple Factory.
- Null Object.

***Architectural Issues***

- Relationship between design patterns and architecture.
- Applying patterns to an architectural solution.
- SOLID (Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion) design principles
- GRASP (General Responsibility Assignment Software Principles) guidelines.
- Coding styles.

*** The lecturer reserves the right to modify the contents of the course to suit the needs of the delegates.*