This course explores proven real-world techniques to meet the biggest challenge in the software development community - building quality systems which fulfill your requirements, and delivering them on time. The focus of the course is to give you the practical skills that are most critical in building well designed software systems. Written exercises are used throughout the course to enhance your understanding of the principles discussed during the lectures. This course explores the most common object-oriented design patterns (Gang of Four) and how to use these patterns to develop solid, robust, and reusable software development applications.
The course covers the patterns in the three core areas of Creational, Structural, and Behavioral and is hands-on with design projects and programming labs. A basic understanding of Java is beneficial to understanding the code samples presented throughout the course. There are no coding exercises presented in this course.

Course Objectives:

- Explore and understand basic Design Pattern concepts.
- Use Design Patterns effectively to build robust, well designed, reusable systems.
- Gain familiarity with the GOF Patterns.

Audience: Application developers, programmers, system designers, and project manager developers who need to improve the systems development through the use of design patterns.

Prerequisites: Some experience in object-oriented thinking/programming, professional experience with object-oriented technologies and UML diagrams, and a basic knowledge of Java.

Number of Days: 2 days

1. Introduction
- What's our World?
- OK – So Just What is a Design Pattern?
- Design Patterns are not Esoteric
- Why Use Patterns?
- The Adapter Pattern
- Reviewing Interfaces & Abstract Classes
- Interface Types
- Interface Definitions
- Abstract Methods
- Abstract Classes
- Using Abstract Classes
- Important Principal of OO Design

2. The Iterator Pattern
- Patterns: Traversing a Collection
- A Simple ArrayList
- Using Our ArrayList
- Using Our Simple Collection
- Another Design for Collection Traversal
- Using Our New Collection
- Differences in Traversing Our Collection
- Why is This Important?
- Why is This a Design Pattern?
- We Will Expand on Our Design

3. Design Patterns – Background
- Design Patterns Arise From Architecture
- Christopher Alexander
- The TimelessWay
- A Core Principle of His Books
- Patterns in *A Pattern Language*
- Sitting Circle (185)
- Different Chairs (251)
- Patterns Evolution in Software
- OOPSLA 88