

**Course Description:** Design Patterns are proven solutions to recurring problems in object-oriented software systems. This course covers sixteen design patterns and includes detailed programming exercises to allow students to practice implementing selected patterns.

**Who Should Attend:** This course is for system architects, designers and programmers working on or preparing for a software project using an object-oriented design.

**Prerequisites:** Programming experience in Java or C++ and some familiarity with object-oriented concepts is required.

**Benefits of Attendance:** Upon completion of this course, students will be able to:

- Describe the purpose of design patterns.
- Understand the ways that design patterns are documented and classified.
- Use the Singleton Pattern to provide controlled access to the sole instance of a class.
- Use the Composite Pattern to represent whole-part hierarchies of objects.
- Use the Factory Method Pattern to eliminate the need to 'hard-code' specific class names.
- Use the Observer Pattern to minimize coupling between domain and interface objects.
- Use the Template Method Pattern to implement the common parts of an operation.
- Use the Strategy Pattern to configure a class with one of many alternate behaviors.
- Use the Iterator Pattern to separate the traversal mechanism from an aggregate object and to support concurrent traversals on the same object.
- Use other creational patterns to help make systems independent of how its objects are created.

## Course Outline:

### Introduction

What is a Design Pattern?  
Reasons to study Design Patterns  
History of Design Patterns  
Cataloging Design Patterns  
Patterns covered in this course  
Design Patterns "Themes"

### The Singleton Pattern

Overview  
C++ Implementation  
Java Implementation  
Java API Example

### The Composite Pattern

Overview  
Structure  
C++ Implementation  
Java Implementation  
Java API Example  
C++ Source Code Example  
Java Source Code Example

### The Factory Method Pattern

Overview  
Structure  
C++ Implementation  
Java Implementation  
Java API Example  
C++ Source Code Example  
Java Source Code Example

### The Observer Pattern

Overview  
Structure  
C++ Implementation  
Java Implementation  
Java API Example  
C++ Source Code Example  
Java Source Code Example

### The Template Method Pattern

Overview  
Structure  
C++ Implementation  
Java Implementation  
Java API Example  
C++ Source Code Example  
Java Source Code Example

### The Iterator Pattern

Overview  
Structure  
C++ Implementation  
Java Implementation  
C++ Standard Template Library  
Java API Example  
C++ Source Code Example  
Java Source Code Example

### The Strategy Pattern

Overview  
Structure  
C++ Implementation  
Java Implementation  
Java API Example  
Java Source Code Example

### Other Creational Patterns

The Abstract Factory Pattern  
The Prototype Pattern  
The Builder Pattern  
Creational Patterns - Summary

### Other Structural Patterns

The Adapter Pattern  
The Bridge Pattern  
Decorator Pattern  
Structural Patterns - Summary

### Other Behavioral Patterns

The Memento Pattern  
The Chain of Responsibility Pattern  
The Visitor Pattern  
Behavioral Patterns - Summary

### Appendix A: References And Web Resources

### Appendix B: Exercises - C++

### Appendix C: Exercises - Java

### Appendix D: Antipatterns