GoF Design Patterns:
Elements of Reusable Object-Oriented Software

COURSE NAME	GoF Design Patterns – Elements of reusable object oriented software
Course Description	The gang of four book "*Design Patterns: Elements of Reusable Object Oriented Software*" 1994 (ISBN 0-201-63361-2) is a fundamental pillar in object oriented software design and programming. Since it's inception it has become a catalog of common vocabulary, and object structure / interactions that programmers had found useful. It assists developers in recognizing problems that fall into familiar categories and it encourages less re-inventing, more flexible and familiar code by offering common solutions to common problems. This training program follows the book by describing, enhancing and demonstrating recurring solutions to common problems in object oriented software design. In this training program we use the **Java programming language** to explain, demonstrate and practice conceptsl from the book. • This training program explains the recommended object oriented design techniques of: ◦ **Interfaces vs. Implementations** – "*Program to an interface – not implementation*" ◦ **Composition instead of Inheritance** – "*Favor object composition over inheritance*" ◦ **Parameterized Types** (Generics) ◦ **Aggregation and Association** • This training program enhances the 23 classic design patterns using the original book UML diagrams and additional hands on code exercises and examples, using the original grouping of design patterns into the 3 following categories: ◦ **Creational Patterns** - Class / objects instantiation and creation using inheritance and delegation ◦ **Structural Patterns** - Using inheritance to compose interfaces and define ways to compose objects to obtain new functionality ◦ **Behavioral Patterns** - communication between objects. • This training program is academic and is intended for the seasoned, as well as the novice programmer, developer, designer and architect alike.

רח' החילזון רמת גן מיקוד טלפון: - פקס -
Http://www.iitc.co.il -

Target Population	object oriented programmers, developers, designers and architects who wish to learn anew or enhance their understanding of fundamental object and class relationships and to be able to provide common solutions to common problems, to do less re-inventing and to adopt or enhance familiarity with the associated design patterns vocabulary.
Pre-requisites	Familiarity with the java programming language and basic object oriented experience. This class does not require extensive development experience – but it is not a substitute for learning OOP.
Course Objectives	• participants will learn to recognize common object structures and interactions • participants will learn to recognize problems that fall into familiar categories • participants will learn to do less re-inventing and become familiar with common solutions and trade-offs associated with them • participants will learn common vocabulary associated with design patterns
Course Topics	Module 1: Design Patters Overview • Motivation • Basic Design Concepts ◦ separate abstract requirements – into an interface ◦ isolate the minimum factor of change – D.R.Y ◦ often we favor composition over inheritance • Design Patterns Taxonomy ◦ **Creational Patterns** – strategies for creating new objects ◦ **Structural Patterns** – how objects may be grouped into more complex structures ◦ **Behavioral Patterns** – define the flow of communication between objects Module 2: Creational Design Patterns ◦ **Factory** (Method) – a central point for instance creation ◦ **Abstract Factory** – select between several possible factories where each factory generates it's own family of objects ▪ abstract factory lab exercise ◦ **Builder** – separate the construction algorithm from the internal representation to construct complex objects with multiple parts ▪ builder lab exercise ◦ **Prototype** – clone an existing object-creation ▪ prototype lab exercise ◦ **Singleton** – A class of which there can be only one instance ▪ using static instances as singltons ▪ using the volatile key word with the double checked lock synchronization pattern ▪ singleton lab exercise

	Module 3: Structural Design Patterns ○ **Adapter** – force a class to conform to a new interfaces ○ **Bridge** – separate abstraction from implementation so the two can vary independently ▪ bridge lab exercise ○ **Composite** – an object may consist of other objects recursively ▪ composite lab exercise ○ **Decorator** – dynamically add functionality to objects ▪ decorator lab exercise ○ **Flyweight** – remove state variables and replace with parameters to share objects in order to save allocations ○ **Facade** – simplify the access to complex systems ○ **Proxy** – access to objects can be controlled by another object with the same interfaces ○ **Java's built-in dynamic proxy** – the invocation handler ○ **Discussion** – similar motives and differences in design patterns so far Module 3: Behavioral Design Patterns ○ **Observer** – a way for an object to notify others when it changes ○ **Mediator** – classes communicate through a mediator for simplification and loose coupling ○ **Memento** – capture and externalize an object's state so it can be restored later ○ **Chain of Responsibility** – pass a request through a chim of objects until it encounters the most appropriate handler ○ **Template** – abstract definition of an algorithm ▪ template lab exercise ○ **Interpreter** – how to include language elements in a program ○ **Strategy** – encapsulate an algorithm in a class ○ **Visitor** – allow a visitor class to perform operations on a visited class ○ **State** – class delegates action to internal State variable so it behaves differently at different states ▪ state lab exercise ○ **Command** – execute pieces of code, oblivious to implementation ▪ command lab exercise ○ **Iterator** – traverse a collection of data within a class ▪ iterator lab exercise ○ **Interpreter** - parse a language expression into a matching tree ○ **Conclusion**
Course Duration	32 hours
Instructor	Tomer Silverman

רח' החילזון רמת גן מיקוד טלפון: - פקס -
Http://www.iitc.co.il -