VerticalDivers®

CLEAN DESIGN WORKSHOP

Clean Design

for

Tesco

3 Days

www.verticaldivers.com

Abstract

course outline

The hands-on course with agility is aimed at architects and developers of OO and functional systems, You will learn to design, apply visual modeling in an agile modeling approach, and a suite of related advanced design topics. . You will learn how to keep your designs clean and simple by applying those design principles through various refactoring techniques. In this intensive seminar there is some lecture time, but the majority is spent in small teams at vast whiteboard spaces while the coach rotates and works with each team, guiding them through the case study problems. The course proceeds through a series of case studies within which you learn design skills. This course approaches software design from three perspectives: the software engineering principles that enable development of quality software, modeling of software elements using the Unified Modeling Language (UML), and reusing design models that are accepted best practices.The assignments will apply an agile design first methodology.

Abstract

course outline

Object-oriented (OO) design and Functional design are a critical skill of successful developers to support design agility. Various studies indicate that after initial release of a software system, at least 50% of effort and cost is spent in modification. To save money, it is skillful and economic to take a long-term view of product development and invest in simple, clear and extensible designs that reduce these costs.

You will leave this workshop with deeper skill to apply OO, functional tactics and create quality designs that have agility – that support reduced modification costs and increased comprehensibility, and hence support business agility. As you participate in our hands-on environment, you will learn how to approach design from the outside-in; one test case at a time, driving the system to better and simpler design while it continues to pass all its tests.

Outline

3

DAYS

course outline

Objective

- Understand Difference between Architecture and Design
- Modeling using UML
- Understand the importance of Design Principles
- Know how to do a Design Review
- Understand code Metrics
- Learn approaches to Refactor brown field code.
- Understand when not to use Patterns.
- Learn Design Anti Patterns.
- Learn How to reduce Coupling
- Learn How to make system Modular
- How to make code Unit testable
- Communicate with other developers using the language of patterns.
- Improve the design of new and existing software.
- Combine different patterns so that they work together in a software design
- Understand some of the design contained in such frameworks as.NET and Java
- Combine Object Oriented Patterns with Functional Patterns.

Outline

3

DAYS

course outline

Prerequisite

The working assumption for this course is that registered students are proficient in programming. Our goal is to move beyond programming-specific concerns to a point where one can think through a design for a software application.

Audience

Developers, Sr. Programmers, Team Leads, etc…

Delivery Method

Problem based Learning
Hands on Design Session
No power points

Outline

3
DAYS

course outline

Day 1

Session 1 - Introduction **2hrs**

- Design vs Architecture
- OOP vs Procedural
- OOP vs Functional

Session 2 - Documenting Design **2hrs**

- UML Class Diagram Quick Tour
- UML Sequence Diagram Quick Tour

Session 3 - Refactoring Inheritance **2hrs**

- Inheritance vs Composition
- DRY principle
- LSP principle
- Runtime Inheritance

Session 4 - OO Structures **2hrs**

- Linked List of Objects
- Hierarchy of Objects
- Heterogeneous Family

Outline

3
DAYS

course outline

Day 2

Session 1 - Refactoring bidirectional Coupling **2hrs**

- Efferent Coupling
- Afferent Coupling
- many to many coupling

Session 2 - Refactoring tight Coupling **2hrs**

- Removing instantiation coupling
- Removing method call coupling
- Inversion of Control

Session 3 - Refactoring Callback **2hrs**

- Functional Interface
- Function Object
- Broadcast

Session 4 - Refactoring Flags **2hrs**

- Cyclomatic Complexity
- Unit testability
- Open Closed Principle

Outline

3

DAYS

course outline

Day 3

Session 1 - Refactoring Non Cohesive Code

- Single Responsibility principle
- Indirection
- Dependancy Injection

Session 2 - Refactoring type check

- Runtime Type Identification
- Refactoring Down cast

Session 3 - Refactoring God class

- Functional Interface
- Function Object
- Lookup
- Decision Tables

Session 4 - Dual Dispatching

- Single Dispatch
- Dual Dispatch
- Multi Dispatch

VerticalDivers®
a division of Colossal Software Technologies
No 237, 1st floor, 7th cross, 1st stage
Indiranagar, Bangalore - 560038