VerticalDivers®

CLEAN CODE WORKSHOP

Clean Code

2 Days

www.verticaldivers.com

Abstract

course outline

Anyone can write code a computer can understand, but professional developers write code *humans* can understand. Clean code is a reader-focused development style that produces software that's easy to write, read and maintain.
This Clean Code workshop alternates between lectures and exercises so that you can experience, first-hand, the practices and disciplines of the following topics:

Objective

Upon completion of this course, students should be able to

- Professionalism, Craftsmanship, and Clean Code.
- Choosing Meaningful Names
- Writing Clean Functions
- The Proper Use of Comments
- Coding Style and Formatting
- Object Oriented Programming vs. Procedural Programming vs functional programming.
- Error Handling Boundaries between Subsystems
- Writing Clean Classes Systems

Abstract

course outline

- Plan vs. Emergence
- Concurrency
- Smells and Heuristics.
- Develop an attitude of professional craftsmanship.
- Tell the difference between good and bad code.
- Create and write good code and transform bad code into good code.
- Create good names, good functions, good objects, and good classes.
- Format code for maximum readability.
- Implement complete error handling without obscuring code logic.
- Apply effective unit testing and refactoring techniques.
- Spot the smells in your code
- Refactor your code using safe and fast refactoring techniques
- Write clean code that is more readable and maintainable
- Break long methods into smaller, more maintainable ones
- Simplify nested / complex conditional blocks
- Design the proper signature for your methods
- Remove duplicated code
- Refactor long switch/case blocks into an object-oriented design using polymorphism

Outline

2
DAYS

course outline

Why Code Quality and Code Design **1hr**

- Activities in software development
- Code Design
- Functions, Classes, Modules
- Total Cost of Owning a Mess
- Technical Debt and Refactoring
- What is a Good Design

Clean behaviour **3hrs**

- Meaningful Names
- The Boy Scout Rule
- Coding Standards
- Comments
- Error Handling
- Refactoring

Object Oriented Programming **4hrs**

- The Newspaper Metaphor
- Solid principles
- GRASP principles

Outline

2
DAYS

course outline

- Smells Within Classes
- Coupling Code smells
- Inversion of Control
- Structural Code smells
- Behavioural Code smells
- Anti patterns
- Refactoring

Functional Programming **4hrs**

- Pure Functions
- Side effects
- Referential transparency
- Internal Iterator vs External Iterator
- Declarative Programming
- Refactoring

Reactive Programming

- Streams
- Observables

Outline

2

DAYS

course outline

- hot vs cold observables
- functional reactive programming

Design by contract

- Pre Conditions
- Post Conditions
- Invariants
- Interface Contracts

Test Driven Development (TDD) **4hrs**

- Three Laws of TDD
- Reduce Coupling using Mock Objects
- Static and Dynamic Mock objects
- Using Static Analysis Tools
- Code Coverage
- Cyclometic Complexity

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