SE002: ***The Practice of Writing Excellent Code***

DURATION: 2 Days; Instructor-led

WHAT YOU WILL LEARN

Writing a computer program is challenging but writing the good program is even harder. Many of the software issues are avoidable or can be minimized if the programmers adopting good practices during coding.

The course covers some common techniques in producing good code for various high level languages.

Although some of the software engineering concepts will be introduced, but the course is focusing mainly on the context of programming.

Upon completion of this course, participants should be able to:

- Appreciate why good techniques are important in software construction
- Apply techniques learnt during coding
- Write better quality code

AUDIENCE

Novice C/C++, Java, C#, VB6/VB.NET Programmers

PREREQUISITES

REQUIRED PREREQUISITES:

- Basic programming concept

METHODOLOGY

This course will be conducted with heavy hands on exercises and discussions with minimum theory.

Examples from various languages depending on the audience language background will be provided to ensure the effective understanding of the course contents.

COURSE OUTLINES

Module 1– Brief Introduction to Software Quality

- Toward Good Code
- Various important -ties
- Software Metrics
- How programming practices influent software quality?

Module 2- The Basic Principles & Strategies

- Loose Coupling
- High Cohesion
- The principle of Information Hiding
- Single assignment of responsibility
- Prevention is Always Better than Cure
- Let system do more and let human do less

Module 3 - The Layout and Presentation of Source Code

- What's the Big Deal?
- Know Your Audience
- What Is Good Presentation?
- Brace Yourself
- Single point of reference
- Setting the Standard

Module 4- Giving Meaningful Things Meaningful Names

- Why Should We Name Well?
- What Do We Name?
- Name Games
- The Nuts and Bolts
- In a Nutshell
- General Dos and Don'ts

Module 5- Techniques for Writing "Self-Documenting" Code

- "No Documentation is the best Documentation"
- Writing code for others to understand
- Techniques for Self-Documenting Code
- Practical Self-Documentation Methodologies

Module 6– Error Handling

- Type of Errors
- The "Fail-First" Heuristic
- Is Exceptions a failure?
- When to use exceptions?
- Dealing with the Inevitable—Error Conditions in Code

Module 7– Defensive Programming Techniques

- What Is Defensive Programming?
- Assume the Worst
- Code Robustness
- Techniques for Defensive Programming
- Constraints

Module 8– Optimizing Programs and Writing Efficient Code

- What Is Optimization?
- What Makes Code Suboptimal?
- Why Not Optimize?
- Why Optimize?
- Optimization Techniques
- Writing Efficient Code

Module 9– How to Produce Good Software Designs

- Programming as Design
- What Do We Design?
- Good Software Design
- How to Design Code
- Design By Contract