



The Art of Software Design and Development – Academic Edition

Overview:

Building software products and services is more an art. It requires skill as also a disciplined approach to build quality software. Many teams realize the pitfalls of the software they developed only well after it is in production when it becomes painful to maintain. Building robust software has more to do with intention, understanding the wisdom behind it & of course implementation, rather than purely knowing the technology/tools.

This program gleans on some of the time tested wisdom accumulated over the years by designers and developers of successful operating systems, languages, tools as also people who have delivered good software products and services. The program also has many snippets of production code which would have rather been designed/written differently. That said it's not about teaching OO programming, design patterns etc. It not only discusses about the best things to do, but also things one would rather avoid.

Target Audience:

- Students who have learnt/are learning a programming language

Pre-requisite

- Familiarity with programming and operating systems

Takeaways:

- Solid understanding of the software development principles & practices not found in textbooks

Delivery Method:

Instructor lead with case studies & code as applicable.

Program Contents:

- Need
- Principles
- Methodology
- Style and conventions
 - Naming, consistency, conventions, syntax...*
- Goofing up code
 - Logic, popular irritants, language semantics, readability, commenting...
- Designing API's
 - Tips what to do and avoid, the many aspects of elegant API's...
- Debugging
 - Myth & facts, design for debugging, how to deal with bugs, unreproducible bugs...
- Testing
 - Unit/System tests, automation, stress test, good practices...
- Performance



- Measurement, tuning – what to optimize, some guidelines for achieving speed, efficiency & proper resource usage...
- Portability
 - Today's computing world, features and definitions, nature of data, language issues, assumptions, known pitfalls...
- Miscellaneous
 - Things not covered elsewhere

Duration: 6 - 8 hours

** Ellipsis (...) at the end of line indicates that there's more to it than just what's mentioned*