



COURSE INSTRUCTOR

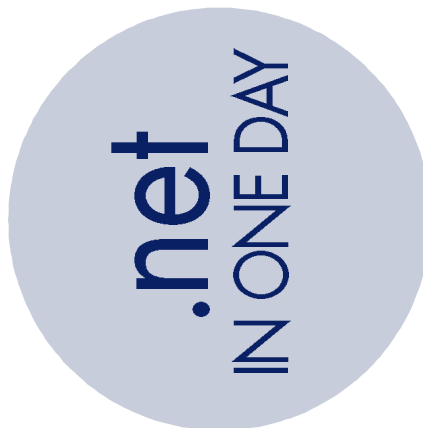
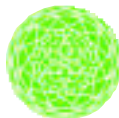
This course will be presented by Dr. Bertrand Meyer, CTO of Interactive Software Engineering and Professor at Monash University. Bertrand Meyer is best known for his best-seller *Object-Oriented Software Construction* (2nd edition,

Prentice Hall, 1997), recipient of the Jolt Award and one of the seminal works on modern software technology. He is the author of 8 other books on software engineering, programming languages and object-oriented development, including *Reusable Software*, *Eiffel: The Language and Object Success*, and of numerous widely cited articles. He is a columnist for *Software Development* magazine, the *Journal of Object-Oriented Programming* and was Department Editor for *IEEE Computer*. He is the series chair for the international TOOLS conferences (USA, Europe, Pacific), devoted to component and object technology. Bertrand is also the editor of the Prentice Hall Object and Component Technology Series. As head of ISE's development team, he has led the design of tools and environments used routinely by major corporations worldwide for their mission-critical applications.

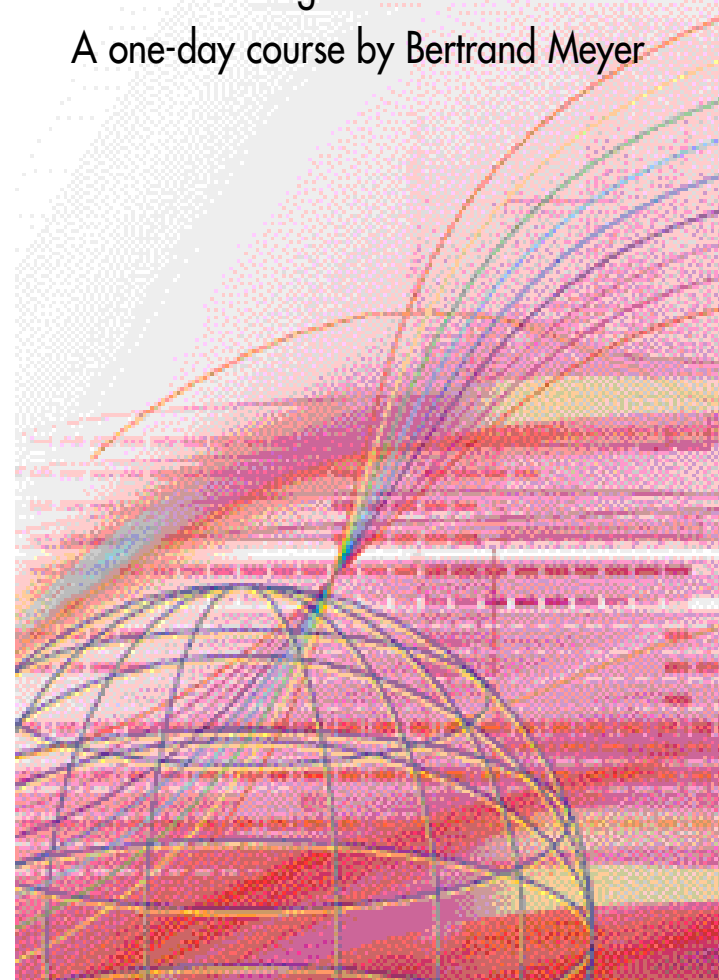
COURSE MATERIAL.

The material distributed to participants includes more than 200 slides, as well as supporting articles.

Interactive Software Engineering
270 Stork Rd, 2nd Floor
Santa Barbara, CA 93117 USA
Phone: (805) 685-1006 Fax: (805) 685-6869



The Multi-Language Platform
for the Age of the Internet
A one-day course by Bertrand Meyer



.net IN ONE DAY

The Multi-Language Platform for the Age of the Internet—A one-day course by Bertrand Meyer

In July of 2000 Microsoft introduced the .NET framework, the most important development since the introduction of Windows in 1991. The result of a \$2 billion investment, .NET is a revolutionary multi-language platform integrating all aspects of application development and closely integrating the Web at every step.

This information-packed one-day course covers the essentials of .NET, including both the “big picture” and a review of all major aspects of the technology. Presented by Dr. Bertrand Meyer, one of the pioneers of modern software technology, it is based on more than one year’s advance exposure to .NET prior to the technology’s official release. Dr. Meyer’s team at Interactive Software Engineering and Monash University worked with Microsoft to integrate ISE’s technology with .NET, culminating in a joint appearance with Bill Gates at the Microsoft Professional Development Conference that first introduced .NET.

WHO SHOULD ATTEND

This course presents a compact, in-depth survey of the .NET technology over one day. It is intended both for managers (VPs of technology, CTOs, project leaders, Web content managers, e-commerce strategy leaders) and for software developers who want to know about the most important technology offering in many years. The .NET technology will affect everyone; no one can afford to miss it.

COURSE OUTLINE

1 Overview

- .NET in 15 minutes: an overview of the technology and its contributions.

2 The challenges: if .NET is the answer, what are the questions?

- Background: the Internet and the evolution of software development. E-commerce and its demands. Requirements of mission-critical applications. Exposing the business model.
- Object Technology: contribution and challenges. O-O languages, GUI tools, databases.
- Component-based development: COM, CORBA, Enterprise Java Beans. The notion of Interface Description Language. Contributions and limitations of pre-.NET component approaches.
- Programming for the Web and e-commerce: CGI scripts, Active Server Pages, Java Server pages. Advantages and drawbacks.
- The state of multi-language interoperability. Approaches to portable application development. Graphics, database issues.

3 .NET: The Vision and the Platform

- The .NET architecture: runtime, framework, platform, web services.
- The .NET runtime: architecture and goals. Comparison with the Java Virtual Machine.
- MSIL: the intermediate language. Security issues and the concept of verifiability. How critical is it to produce verifiable code?
- Organizing and extending your components: assemblies and metadata. An application: equipping components with contracts.

4 The .NET object model and type system

- Classes, methods, fields, properties and events
- .NET types: reference and value types, array types, arrays.

- Inheritance concepts: multiple interface inheritance, covariance.

- Encapsulating behavior: delegates
- C#: a language for programming .NET
- C# versus Java

- .NET mechanisms and the dominant languages: commonalities and mismatches. How easy is it to map an existing language into the .NET model?

5 Language interoperability

- Available languages and degree of interoperability. Cross-language inheritance; cross-language debugging.
- The Common Language System: both a consumer and an extender be? Levels of compliance.
- Advantages and challenges of CLS compliance
- Examples: combining components from various languages.
- From a common runtime to a common development environment: Visual Studio.NET and the concept of multi-language, pluggable environment. GUI, browsing, debugging.

6 Frameworks and applications

- Web and Win Forms
- Remoting and threading capabilities
- ASP+: Active Server Pages +. Building advanced Web sites for e-commerce.
- Web services, SOAP and Building Block Services.
- Database access and manipulation: ADO+

7 Summary and perspective

- .NET and the competition
- The significance of .NET
- Future developments
- Corporate strategies: getting ready for .NET