

Visual Basic.NET, Programming

Hands-on course of 5 days - 35h

Ref.: VBN - Price 2024: CHF2 970 (excl. taxes)

This course will teach you how to develop VB.NET applications with Framework 2.0 by applying object programming principles. It will give you a complete mastery of the language, including using libraries and interfacing with other languages.

HANDS-ON WORK

Half of the course is done through workshops.

THE PROGRAMME

last updated: 01/2018

1) Introduction to the .NET platform

- The principles and architecture of the .NET environment.
- Object distribution architecture in .NET, executing remote objects and components.
- Structure of a VB.NET program. The namespace notion.
- Presentation of Framework .NET.
- Development environment and tools.
- MSIL language: the principles of the intermediary language, and JIT (Just In Time) compilation.

Hands-on work : Example of the minimum VB.NET program. Managed mode execution. Using the Visual Studio.

2) Basic syntax: data, expressions and instructions

- Values, operators, manipulating variables.
- Data types: Common Type System.
- Complex data: tables. Handling tables, creating and manipulating them.
- Flow control instructions: loops, test, flow control.
- Functions.
- Program sequencing, the notion of exceptions.
- Some minor new features: Continue, IsNot, Using, property visibility, support for non-signed types, Global key word, Try
- IsTrue and IsFalse operators.
- Generating documentation.

Hands-on work : Writing basic programs in VB.NET.

3) Object-Oriented Programming

- Classes and objects: real world object modelling.
- Notions of attributes, methods and properties.
- Inheritance. Polymorphism.
- Implementing multiple interfaces.
- Representing the object model.

4) Classes and objects in VB.NET

- Defining classes. Defining objects.
- Defining the content of the class: methods and attributes. Overloading methods and operators.
- The life cycle of objects: constructor, destructor.

PARTICIPANTS

The course is aimed at experienced professional developers who want to learn VB.NET development.

PREREQUISITES

Good knowledge in programming. Basic knowledge of the object principles. Necessary experience of software development.

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more. Participants also complete a placement test before and after the course to measure the skills they've developed.

TEACHING AIDS AND TECHNICAL RESOURCES

- The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.
- At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.
- A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@ORSYS.fr to review your request and its feasibility.

- Typing in VB.NET: type handling and conversion.
- Properties. Definition. "read only" properties.
- Visibility of the members of a class: properties and methods. Using namespaces.
- Deriving and inheriting classes. Controlling access during derivation.
- System.Object base class.
- Manipulating attributes: the principle of metadata. Class, method, field attributes.
- Manipulating and handling tables. Handling memory. Using the garbage collector.
- Foreach structure and indexer: principle and utilisation.
- Partial classes. Customised event management. Generic classes. Cancellable types.
- Covariance and contravariance of delegates.

Hands-on work : Writing base classes. Manipulating methods and properties. Deriving a class. Using attributes, indexers and tables.

5) Interfaces

- Principle. Declaring and utilising an interface.
- Interfaces and inheritance. The principle of multiple inheritance through interfaces.

Hands-on work : Writing programs implementing interfaces.

6) Exception handling

- The principle of handling exceptions and events.
- The structure of inter-language exception handling: triggering an exception in one language and processing it in another
- Using delegates: static and dynamic delegates for handling events.

Hands-on work : Handling exceptions in VB.NET. Example of inter-language exceptions. Utilising delegates.

7) Assemblies

- The notion of assembly. Creating an assembly. Organising a project through assemblies.
- Private assembly and shared assembly: signature, role of the GAC (Global Assembly Cache), implementation in the GAC. Ass
- Assembly and multi-language support.

Hands-on work : Organising development with assemblies. Creating a public assembly. Using sn (Strong Name) for the signature, implementation in GAC.

8) Framework .NET's base classes and object

- The principle of Framework.
- CTSs: base classes (Int, String).
- General class: .NET System, Threading.
- Data structure manipulation class: Math, ArrayList.
- Processing chains, date and time.
- Dynamic tables. Regular expressions.
- Implementing type conversion through classes. Creating the type through an instance.

Hands-on work : Using base classes. Using types as classes of the .NET environment.

9) Handling inputs/outputs

- The hierarchy of classes.
- FileStream and StreamReader/StreamWriter.
- Manipulating the file system.
- Asynchronous Inputs/Outputs.

Hands-on work : Using .NET classes for inputting/outputting.

10) Applications developed with Framework .NET

- The essential elements of Framework: user interface with Windows.Form, ADO.NET, ASP.NET, Web services.
- Multi-target development: .NET peripherals, development targets.
- The architecture of applications with Web Services.
- Example of programs using .NET's essential components.

- Other areas of .NET: integrating Web services.

Hands-on work : Example of an application with a graphics interface. Using Windows.Forms.

DATES

Contact us