

Hands-on course , 5  
day(s)  
Ref : PSH

### Pre-requisites

Programming experience and knowledge of computer languages. Experience with the C language would be an asset for this course.

### Next sessions

# C# Programming

## OBJECTIVES

*This course covers the C# language with a deep coverage of the object oriented programming concepts. During the course participants will develop object oriented applications using C# and Visual Studio 2008/2010 within the .NET Framework.*

### 1) Introduction

### 2) Language syntax

### 3) Object Oriented Programming

### 4) Class development and objects creation

### 5) Interfaces

### 6) Exception handling

### 7) Assemblies

### 8) Application development with .NET technologies

## Workshop

*The practical exercises have been designed to illustrate all the elements of the language and to implement the concepts of object-oriented design: all the exercises contain an analysis/design stage followed by a programming stage.*

## 1) Introduction

- Principle and architecture of the .NET environment.
- Main components: language, CLR, CTS, ...
- Benefits of the MSIL language.
- Execution model in .NET: managed execution benefits.
- Visual Studio. Type of projects. Help and documentation.

### Workshop

*Development of a Windows application using the C# language. The steps necessary to construct, compile and run a program are covered in detail.*

## 2) Language syntax

- Basic understanding of the C# language.
- Values, operators and variables.
- Reference types : classes and interfaces.
- Exception handling. Program structure.
- Program debugging.

### Workshop

*Programs development in C#.*

## 3) Object Oriented Programming

- Encapsulation and abstraction.
- Classes and objects. Inheritance. Polymorphism.
- Multiple interfaces implementation.
- Introduction to the modeling language UML 2.0.

## 4) Class development and objects creation

- Class and object definition.
- Class members: methods and properties.
- Static members. Overloads of methods.
- Object's life cycle and the garbage collector in action.
- Benefits of a typed language.
- Application structure through Namespaces.
- Inheritance: which members are inherited from a derived class ?
- Abstract class. Generic collections and classes.

### Workshop

*Development of a C# application highlighting the main tasks of a professional .NET developer.*

## 5) Interfaces

- Definition. Explicit and implicit implementations.
- The role of the interface in inheritance.

### Workshop

*Development of an object oriented C# application using the design by contract paradigm.*

## 6) Exception handling

- Principle. Best practices in exception handling.

- Creating a custom exception class. Libraries.

**Workshop**

*Exception handling implementation.*

## 7) Assemblies

- Definition. Organizing a project through assemblies.
- Creating shared assemblies. Assembly deployment.
- Use of the Global Assembly Cache (GAC).

**Workshop**

*Creation of shared assemblies. Distribution to the users' desktop and GAC*

## 8) Application development with .NET technologies

- Evolution of the data access model in .NET Framework.
- ADO.NET technology for data handling.
- Web development with the ASP.NET namespaces.
- Service Oriented applications using Web Services.