

## Programming (Object Oriented) in C# 2005

**Descripción:** Este curso de 35 a 40 horas es una introducción práctica a la programación en C# 2005, al aprendizaje del Visual Studio 2005 y al uso de los servicios proporcionados por .NET. El curso hace énfasis en el lenguaje C# y en cómo construir aplicaciones C# desde una perspectiva de la programación orientada a objetos.

**Audiencia:** Programadores que necesitan iniciarse en el lenguaje C# 2005 (C#.NET) para diseñar y desarrollar en aplicaciones .NET. Es recomendable para quienes en un futuro desean aprender a hacer aplicaciones Web con ASP.NET y C# 2005 o para quienes desean, después, hacer aplicaciones Windows con Windows Forms.

**Prerrequisitos:** Experiencia en Programación en algún lenguaje de alto nivel como Cobol, Perl o C.

### Contenido

#### 1. NET: WHAT YOU NEED TO KNOW

- Getting Started
- .NET: What Is Really Happening
- .NET Programming in a Nutshell
- Viewing the Assembly
- Viewing Intermediate Language
- Understanding .NET
- Visual Studio 2005
- Visual C# 2005 Express Edition
- Creating a Console Application
- Saving Your Project
- Adding a C# File
- Using the Visual Studio Text Editor
- IntelliSense
- Build and Run the Project
- Pausing the Output
- Visual C# and GUI Programs
- .NET Documentation

#### 11. ARRAYS AND INDEXERS

- Arrays
- One Dimensional Arrays
- System.Array
- Interfaces for System.Array
- Random Number Generation
- Constructors for Random
- Next Methods
- Jagged Arrays
- Rectangular Arrays
- Arrays As Collections
- Bank Case Study: Step 1
- Account Class
- Bank Class
- TestBank Class
- Atm Class
- Running the Case Study
- Indexers
- Using the Indexer

## 2. FIRST C# PROGRAMS

- Hello, World
- Compiling, Running (Command Line)
- Program Structure
- Namespaces
- Variables
- Expressions
- Calculations Using C#
- More about Output in C#
- Input in C#
- More about Classes
- InputWrapper Class
- Echo Program
- Using InputWrapper
- Compiling Multiple Files
- Multiple Files in Visual Studio
- The .NET Framework

## 3. DATA TYPES IN C#

- Strong Typing
- Typing in C#
- Typing in C++
- Typing in Visual Basic 6
- C# Types
- Integer Types
- Integer Type Range
- Integer Literals
- Floating Point Types
- Floating Point Literals
- IEEE Standard for Floating Point
- Decimal Type
- Decimal Literals
- Character Type
- Character Literals
- string
- Escape Characters
- Boolean Type
- Implicit Conversions
- Explicit Conversions
- Nullable Types

## 4. OPERATORS AND EXPRESSIONS

- Operator Cardinality
- Arithmetic Operators
- Multiplication
- Division
- Additive Operators
- Increment and Decrement
- Relational Operators
- Conditional Logical Operators
- Bitwise Operators
- Bitwise Logical Operators
- Bitwise Shift Operators
- Assignment Operators
- Expressions
- Precedence
- Associativity
- Checking

## 12. INHERITANCE

- Inheritance Fundamentals
- Inheritance in C#
- Single Inheritance
- Root Class – Object
- Access Control
- Public Class Accessibility
- Internal Class Accessibility
- Member Accessibility
- Member Accessibility Qualifiers
- Method Hiding
- Method Hiding and Overriding
- Initialization
- Initialization Fundamentals
- Default Constructor
- Overloaded Constructors
- Invoking Base Class Constructors
- Bank Case Study: Step 2
- Bank Case Study Analysis
- Account
- CheckingAccount
- SavingsAccount
- TestAccount
- Running the Case Study

## 13. VIRTUAL METHODS AND POLYMORPHISM

- Introduction to Polymorphism
- Abstract and Sealed Classes
- Virtual Methods and Dynamic Binding
- Type Conversions in Inheritance
- Converting Down the Hierarchy
- Converting Up the Hierarchy
- Virtual Methods
- Virtual Method Cost
- Method Overriding
- The Fragile Base Class Problem
- override Keyword
- Polymorphism
- Polymorphism Using "Type Tags"
- Polymorphism Using Virtual
- Abstract Classes
- Keyword: abstract
- Sealed Classes
- Heterogeneous Collections
- Bank Case Study: Step
- Case Study Classes
- Run the Case Study
- Account
- CheckingAccount, SavingsAccount
- Bank and Atm
- TestBank

## 5. CONTROL STRUCTURES

- If Tests
- Blocks
- Loops
- while Loop
- do/while Loops
- for Loops
- Arrays
- foreach Loop
- break
- continue
- goto
- Structured Programming
- Multiple Methods
- switch
- switch in C# and C/C++

## 6. OBJECT-ORIENTED PROGRAMMING

- Objects
- Objects in the Real World
- Object Models
- Reusable Software Components
- Objects in Software
- State and Behavior
- Abstraction
- Encapsulation
- Classes
- Inheritance Concept
- Relationships among Classes
- Polymorphism
- Object-Oriented Analysis and Design
- Use Cases
- CRC Cards and UML

## 7. CLASSES

- Classes as Structured Data
- Classes and Objects
- References
- Instantiating and Using an Object
- Assigning Object References
- Garbage Collection
- Methods
- Public and Private
- Abstraction
- Encapsulation
- Initialization
- Initialization with Constructors
- Default Constructor
- this
- Static Fields and Methods
- Static Methods
- Static Constructor
- Constant and Readonly Fields

## 14. FORMATTING AND CONVERSION

- Introduction to Formatting
- ToString
- ToString in Your Own Class
- Using Placeholders
- Format Strings
- Simple Placeholders
- Controlling Width
- Format String
- Currency
- String.Format
- PadLeft and PadRight
- Bank Case Study: Step 4
- Type Conversions
- Conversion of Built-In Types
- Conversion of User-Defined Types

## 15. EXCEPTIONS

- Introduction to Exceptions
- Exception Fundamentals
- .NET Exception Handling
- Exception Flow of Control
- Context and Stack Unwinding
- System.Exception
- User-Defined Exception Classes
- Structured Exception Handling
- Finally Block
- Bank Case Study: Step 5
- Inner Exceptions
- Checked Integer Arithmetic

## 16. INTERFACES

- Interfaces in C#
- Interface Inheritance
- Programming with Interfaces
- Implementing Interfaces
- Using an Interface
- Dynamic Use of Interfaces
- is Operator
- as Operator
- Bank Case Study: Step 6
- Common Interfaces in Case Study –IAccount
- Apparent Redundancy
- IStatement
- IStatement Methods
- IChecking
- ISavings
- The Implementation
- SavingsAccount
- The Client
- Resolving Ambiguity
- Access Modifier
- Explicit Interfaces Test Program

## 8. MORE ABOUT TYPES

- Overview of Types in C#
- Structures
- Uninitialized Variables
- Copying a Structure
- Hotel.cs
- HotelCopy.cs
- Results of Hotel Copy
- Classes and Structs
- Enumeration Types
- Reference Types
- Class Types
- Object
- String
- Arrays
- Default Values
- Boxing and Unboxing

## 9. METHODS, PROPERTIES, AND OPERATORS

- Static and Instance Methods
- Method Parameters
- No “Freestanding” Functions in C#
- Classes with All Static Methods
- Parameter Passing
- Parameter Terminology
- Value Parameters
- Reference Parameters
- Output Parameters
- Structure Parameters
- Class Parameters
- Method Overloading
- Modifiers as Part of the Signature
- Variable Length Parameter Lists
- Properties
- Operator Overloading
- Operator Overloading in the Class Library

## 10. CHARACTERS AND STRINGS

- Characters
- Character Codes
- ASCII and Unicode
- Escape Sequences
- Strings
- String Class
- String Literals and Initialization
- Concatenation
- Index
- Relational Operators
- String Equality
- String Comparisons
- String Comparison
- String Input
- String Methods and Properties
- StringBuilder Class
- StringBuilder Equality
- Command Line Arguments
- Command Line Arguments in the IDE
- Command Loops
- Splitting a String

## 17 .NET INTERFACES AND COLLECTIONS

- Collections
- Count and Capacity
- foreach Loop
- Array Notation
- Adding to the List
- Remove Method
- RemoveAt Method
- Collection Interfaces
- IEnumerable and IEnumerator
- ICollection
- IList
- A Collection of User-Defined Objects
- Duplicate Objects
- A Correction to AccountList (Step 1)
- Bank Case Study: Step 7
- Copy Semantics and ICloneable
- Copy Semantics in C#
- Shallow Copy and Deep Copy
- Reference Copy
- Memberwise Clone
- Using ICloneable
- Comparing Objects
- Sorting an Array
- Anatomy of Array.Sort
- Using the is Operator
- The Use of Dynamic Type Checking
- Implementing IComparable
- Running the Program
- Complete Solution
- Writing Generic Code
- Using a Class of object
- Generic Types
- Generic Syntax in C#
- Generic Client Code
- System.Collections.Generic

## 18. DELEGATES AND EVENTS

- Overview of Delegates and Events
- Callbacks and Delegates
- Usage of Delegates
- Declaring a Delegate
- Defining a Method
- Creating a Delegate Object
- Calling a Delegate
- A Random Array
- Anonymous Methods
- Combining Delegate Objects
- Account.cs
- DelegateAccount.cs
- Events
- Events in C# and .NET
- Client Side Event Code

## 19. INTRODUCTION TO WINDOWS FORMS

- Creating a Windows Forms App
- Partial Classes
- Windows Forms Event Handling
- Add Events for a Control
- Events Documentation
- Closing a Form
- ListBox Control

### **Notas:**

- \* Este curso es parte de nuestro Diplomado Web Programming with C# 2005 & ASP.NET 2005.
- \* El curso utiliza .NET 2.0
- \* El curso incluye tópicos de Windows Forms y ADO.NET suficientes para aprender lo básico de estos temas pero se centra más que nada en el lenguaje C# ya que hay otros cursos más completos de Windows Forms y ADO.NET.

### **Duración aproximada:**

35-40 horas

### **Lugar:**

Altadena 26. Col. Nápoles, México, D .F.

### **Incluye:**

Material del curso, mochila, estacionamiento y servicio de cafetería.

### **Nota:**

El material que se entrega está en inglés técnico.

### **Formas y condiciones de pago:**

Para la inscripción, aplicación de los descuentos y aseguramiento de que el diplomado se impartirá en las fechas pactadas se requiere el pago anticipado 2 semanas antes de la fecha de inicio con depósito o transferencia bancaria a la siguiente cuenta:

- ❖ Depósito en Banamex cuenta 4923239 Suc. 575 a nombre de Desarrollo y Capacitación en Internet, S. A. de C. V. (CLABE en caso de transferencia electrónica vía Internet 002180057549232394)
- ❖ O cheque a nombre de Desarrollo y Capacitación en Internet, S. A. de C. V.

# DCInternet