

## Curso C# 2010 Object-Oriented Programming in C#® (VS 2010)

**Descripción:** Este curso es ideal para quien desea iniciarse en la programación en C#. El principal objetivo de este curso es presentar a C# como lenguaje de programación para desarrollos basados en la plataforma .NET. Está actualizado a las versiones de Visual Studio® 2010 y .NET 4.0, productos en los cuales se presentan nuevas tecnologías tales como tipos de datos dinámicos, argumentos nombrados y opcionales, y el uso de variancia en interfaces genéricas. Este curso está diseñado para ser base en la formación de programadores que no cuentan con experiencia en lenguajes orientados a objetos tales como C++ o Java™. Es ideal para programadores que vienen de lenguajes tales como Visual Basic® 6.0 o COBOL que desean aprender C#. Si bien el objetivo del curso es comprender la sintaxis general del lenguaje, un aspecto sumamente importante es la parte de programación orientada objetos, y este curso hace hincapié en esta área por ser fundamental para otros temas de especialización en tecnologías .NET tales como Windows Forms, ASP.NET, ADO.NET, LINQ y AJAX, etc.

Además, que el curso ofrece los conocimientos necesarios para que puedas crear tus propios de objetos de negocio y saques el verdadero provecho de un lenguaje orientado a objetos. Frecuentemente es difícil, para los programadores formados en lenguajes procedurales, pensar en términos de objetos por eso la importancia del enfoque que este curso tiene en esa área en particular. Por lo tanto aprenderás conceptos fundamentales y especializados de la programación orientada a objetos tales como clases, herencia, polimorfismo e interfaces pasando también por elementos que C# agrega y potencian esta área tales como propiedades auto implementadas, indexadores, delegados y eventos, entre otros más.

Con todo lo anterior conocer C# es contar con un lenguaje altamente competitivo para desarrollar aplicaciones orientadas a objetos y basadas en componentes. Además de presentar todas esas características del lenguaje, el curso también presenta cómo podemos aprovechar e integrar en nuestras aplicaciones el amplio repertorio de servicios proporcionados por el Framework .NET y así desarrollar diferentes tipos de soluciones, por ejemplo, de escritorio o basadas en Web entre muchas más. Esta colaboración del lenguaje con el Framework .NET se muestra con una introducción al desarrollo de aplicaciones de escritorio usando Windows Forms, que si bien es un nivel introductorio, permite mostrar cual es el alcance de la tecnología .NET.

**Audiencia:** Programadores que necesiten aprender C# como lenguaje de programación para aplicaciones basadas en la plataforma .NET.

**Prerrequisitos:** El estudiante deberá tener experiencia en programación en lenguajes de alto nivel, no necesariamente orientados a objetos.

## 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 2010
- Creating a Console Application
- Adding a C# file
- Using the Visual Studio Text Editor
- IntelliSense
- Build and Run the Project
- Pausing the Output
- Visual C# and GUI Program
- .NET Documentation

### 2. First C# Programs

- Hello, World
- Compiling, Running (Command Line)
- Program Structure
- Namespaces
- Variables
- Expressions
- Assignment
- 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

### 11. Arrays and Indexers

- Arrays
- One Dimensional Arrays
- System.Array
- Random Number Generation
- 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

### 12. Inheritance

- Inheritance Fundamentals
- Inheritance in C#
- 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

### 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
- Short-Circuit Evaluation
- Ternary Conditional Operators
- Bitwise Operators
- Bitwise Logical Operators
- Bitwise Shift Operators
- Assignment Operators
- Expressions
- Precedence
- Associativity
- Checking

### 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
- Sealed Classes
- Heterogeneous Collections
- Bank Case Study: Step 3
- Case Study Classes
- Run the Case Study
- Account
- CheckingAccount, SavingsAccount
- Bank and Atm
- TestBank

### 14. Formatting and Conversion

- Introduction to Formatting
- ToString
- ToString in Your Own Class
- Using Placeholders
- Format String
- Controlling Width
- Format String
- Currency
- String.Format
- PadLeft and PadRight
- Bank Case Study: Step 4
- Type Conversions
- Conversion of Built-In 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
- Structure Exception Handling
- Family Block
- Bank Case Study: Step 5
- Inner Exceptions
- Checked Integer Arithmetic

## 5. Control Structures

- If Test
- Blocks
- Loops
- while Loop
- do while Loops
- for Loops
- Arrays
- foreach Loop
- break
- continue
- goto
- Structure 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 Concepts
- Relationships among Classes
- Polymorphism
- Object-Oriented Analysis and Design
- Use Cases
- CRC Cards and UML

## 7. Classes

- Classes as Structure 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

## 8. More about Types

- Overview of Types in C#

## 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

## 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

- Structures
- Uninitialized Variables
- Copying a Structure
- Hotel.cs
- HotelCopy.cs
- Classes and Structs
- Enumeration Types
- Reference Types
- Class Types
- object
- string
- Arrays
- Default Values
- Boxing and Unboxing
- Implicitly Types Variables

## 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
- Auto-Implemented 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 Input
- String Methods and Properties
- StringBuilder Class
- StringBuilder Equality
- Command Line Arguments
- Command Line Arguments in the IDE
- Command Loops
- Splitting a String

- Object Initializers
- Collection Initializers
- Anonymous Types

## 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
- Lambda Expressions
- Named Method
- Events
- Events in C# and .NET

## 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

## 20. New Features in C#

- dynamic Type
- dynamic versus object
- Behavior of object
- Behavior of dynamic
- Names Arguments
- Optional Arguments
- Book Class
- Using Optional Arguments
- Automating Office with C# 4.0
- Automating Excel
- Automating Word
- Variance in Generic Interfaces
- Variance with IComparer<T>
- Interfaces with Variance Support

**Incluye:**

- ❖ Un manual original importado en inglés.
- ❖ Diploma de participación por parte de nuestra empresa.
- ❖ Estacionamiento.
- ❖ Servicio de cafetería.

**Duración: 35 horas****Formas de pago:**

Este pago debe realizarse en una sola exhibición de las siguientes maneras:

1. 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
2. Cheque a nombre de Desarrollo y Capacitación en Internet, S. A. de C. V.

**Clave SWIFT para traspasos internacionales: BNMXXMM**

# DCInternet