

Introduction to Programming & Coding | C# .Net Basics for Non-Developers (TTCODE101-N)

Modality: Virtual Classroom

Duration: 5 Days

SUBSCRIPTION: No

About this course:

Introduction to Programming & Coding | C# .Net Basics for Non-Developers is a skills-focused, hands-on coding course that teaches students the fundamentals of programming object oriented (OO) applications with C#.Net to a basic level, using sound coding skills and best practices for OO development. This course is presented in a way that enables interested students from any background to embrace the fundamentals of coding as well as an introduction to C#.Net and Visual Studio, in a gentle paced environment that focuses on coding basics, moving at the comfort level of the attendees.

Students are introduced to the application development cycle, structure of programs, and specific language syntax. The course also contains "Thinking Like a Programmer" sections that provide students insight on how to develop common algorithms. The course covers console and file I/O, string and character manipulation, managing data using collections and fundamental object-oriented programming concepts. Error handling techniques are also emphasized. The course also introduces how to access databases using ADO.NET and illustrates how to build user interfaces using Windows Forms. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Students will leave this course armed with the required skills to begin their journey as a .Net programmer using modern coding skills and technologies. This course provides students with a solid foundation for continued .Net programming training and skills development.

The average salary of a .NET Software Developer / Programmer is **\$65,791** per year.

Course Objective:

Working within an engaging, hands-on learning environment led by our highly-experienced coding mentors, students will learn how to:

- Using Visual Studio to create C# applications
- Working with the .NET data types
- Creating variables with the proper scope and using operators to build complex expressions
- Using control structures such as if, while and for
- Using procedures to build complex applications
- Designing and using classes
- Using arrays and .NET collections
- Throwing and trapping exceptions using the C# try and catch statements

- Using ADO.NET to access databases
- Working with files and directories
- Building simple Windows Forms applications
- Building and using derived classes
- Defining and implementing interfaces
- Performing advanced string manipulation

Audience:

This course is intended for anyone who is new to software development and wants, or needs, to gain an understanding of the fundamentals of coding and basics of .Net and object-oriented programming concepts or anyone who wants to have a basic understanding of and learn how to code C# .Net applications and syntax.

Attendees might include:

- Anyone who wants exposure to basic coding skills, or who wants to begin the process of becoming an OO application developer
- Technical team members from non-development roles, re-skilling to move into software and application development roles within an organization
- Recent college graduates looking to apply their college experience to programming skills in a professional environment, or perhaps needing to learn the best practices and standards for programming within their new organization
- Technical managers tasked with overseeing programming teams, or development projects, where basic coding knowledge and exposure will be useful in project oversight or communications needs

Prerequisite:

Before attending this course, students must have:

- Ability to use computers to start programs, open and save files, navigate application menus and interfaces
- Ability to understand logical concepts such as comparisons
- Understand number theory
- Ability to create, understand, and follow structured directions or step-by-step procedures
- Ability to understand and apply abstract concepts to concrete examples

Course Outline:

Module 1: Application Development Fundamentals

- Overview of Programming Concepts
- Understanding the Structure of a C# Program
- Understanding Data Types
- Working with Variables
- Reading From and Writing to the Console
- Overview of the .NET Framework

- Using Visual Studio
- Thinking Like a Programmer
- Designing Algorithms
- Finding Patterns in Code

Module 2: Managing the Flow of an Application

- Conditional Constructs
- Working with if/else Constructs
- Working with switch
- Looping Constructs
- Working with while Constructs
- Working with for and foreach
- Thinking Like a Programmer
- Making Decisions
- Designing Loops

Module 3: Working with Data

- Using Integer Data Types
- Using Floating Point Data Types
- Using Characters and Strings
- Using Dates
- Using Booleans
- Working with Constants and Literals

Module 4: Using Procedures to Modularize Code

- Defining and Calling Subroutines
- Defining and Calling Functions
- Understanding Variable Scope
- Overloading Procedures
- Passing Parameters
- Understanding by value vs by reference Parameters
- Understanding the Call Stack
- Thinking Like a Programmer
- Finding the Procedures
- Refactoring
- Improving Productivity with Snippets

Module 5: Object-Oriented Programming

- Understanding Object-Oriented Concepts
- Designing Classes
- Coding Properties and Methods
- Initializing Objects with Constructors
- Overloading Constructors
- Declaring and Instantiating Objects

- Calling Properties and Methods
- Understanding Value Types vs. Reference Types
- Working with Shared Data Members and Methods
- Thinking Like a Programmer
- Finding your Classes

Module 6: Understanding Namespaces

- Understanding the Role of Namespaces
- Understanding .NET Namespaces
- Defining Custom Namespaces
- Referencing Members in a Namespace
- Using the using Statement

Module 7: Working with Data Collections

- Understanding Arrays
- Declaring and Instantiating Arrays
- Iterating through Arrays
- Working with System.Array Methods
- Copying, Sorting, Searching and Resizing
- Passing Arrays to Methods
- Working with param array Parameters
- Working with Command-line Arguments
- Understanding .NET Collections
- Managing Data Using List
- Managing Data Using Dictionary<TKey, TValue>
- Working with LINQ
- Making LINQ Queries
- Enumerating LINQ Query Results
- Working with Anonymous Types
- Using Extension Methods with LINQ

Module 8: Building Inheritance Hierarchies

- Understanding Inheritance
- Building Derived Classes
- Understanding Constructors in Derived Classes
- Defining and Using Protected Class Members
- Understanding Polymorphism
- Defining Overridable Methods
- Overriding Methods
- Understanding Abstract Classes
- Defining Abstract Classes
- Inheriting from Abstract Classes

Module 9: Building GUIs with Windows Forms

- Overview of Windows Forms
- Designing Forms
- Working with Controls
- Using Labels and Textboxes
- Using Buttons
- Using Checkboxes and Radio Buttons
- Using Menus
- Using List Controls
- Handling Events

Module 10: Accessing Databases Using ADO.NET

- Understanding the ADO.NET Object Model
- Opening Connections
- Executing Queries Using Commands
- Iterating Through Results Using DataReaders
- Calling Stored Procedures
- Passing Parameters to Stored Procedures
- Working with DataSets
- Binding Data to Controls

Module 11: Exception Handling

- Understanding Exception Handling
- Using try/catch to Handle Exceptions
- Working with the Exception Class
- Understanding Exception Propagation
- Using finally to Manage Cleanup Processing
- Throwing Exceptions

Module 12: Working with Files, Directories and Streams

- Using the System.IO Namespace
- Discovering Drives
- Working with Directories
- Working with Files
- Parsing a File Path
- Understanding Streams
- **Working with FileStream**
- **Reading and Writing Text Files**
- **Understanding other Types of Streams**

Module 13: Working with Strings

- Working with the String Class
- Working with String Literals and Escape Sequences
- Understanding String Manipulation Performance Issues

- Working with the StringBuilder Class
- Formatting Output with String.Format

Module 14: Building N-Tier Applications

- Building Large Scale Applications
- Designing N-Tier Applications
- Building .NET Assemblies
- Referencing Assemblies