Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975)

Category: Software Development & Programming | **Vendor:** Trivera Tech

Duration: 40.00 hours (5 days) **32.5 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format: Instructor Led - Online

Course Overview

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 .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.

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 .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.

Who Should Attend

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 / .Net Core 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

Prerequisites & Entry Requirements

General Prerequisites:

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

Learning Outcomes

Upon successful completion of this course, participants will be able to:

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

Detailed Course Outline

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

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 for each
- Thinking Like a Programmer
- Making Decisions
- Designing Loops

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

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

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

Understanding Namespaces

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

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
- Working with LINQ
- Making LINQ Queries
- Enumerating LINQ Query Results
- Working with Anonymous Types
- Using Extension Methods with LINQ

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

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

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

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

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

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

Building N-Tier Applications

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

, Module 1: Introduction to Core Programming Concepts

This module provides background and foundational information on how computers process information, discusses the different types of applications that a programmer might be creating, and then provides information on how code is compiled and interpreted by a computer.

After completing this module, students will be able to:

Describe computer data storage and processing concepts

Describe application types

Describe the lifecycle of an application

Describe code compilation

Lessons

Computer Data Storage and Processing

Application Types

Application Life-Cycle

Code Compilation

Lab : Thinking Like a Computer

Module 2: Core Programming Language Concepts

This module covers programming language syntax and the importance of using good syntax and following the syntax rules for the chosen language. This module also discusses the core data types and how to store these data types in computer memory by using variables and constants.

After completing this module, students will be able to:

Define syntax

Explain the different types of core data used in programs

Declare and use variables and constants in a computer program

Lessons

Syntax

Data Types

Variables and Constants

Lab : Working with Data Types

Module 3: Program Flow

This module covers how code is executed in a computer program, such as top to bottom, in structured programming and branching in code execution. The module teaches these concepts through the use of functions, decision structures, and looping constructs.

After completing this module, students will be able to:

Describe structured programming

Create and use functions in your code

Create and use decision structures

Create and use looping structures

Lessons

Introduction to Structured Programming Concepts

Introduction to Branching

Using Functions

Using Decision Structures

Introducing Repetition

Lab : Creating Functions, Decisions, and Looping

Module 4: Algorithms and Data Structures

This module introduces the concept of an algorithm by examining a daily routine such as a morning routine for getting up and going to work, outlining all the steps required including the decisions to be made as the routine progresses. The module also discusses how to translate these set of steps into pseudo code for evaluation of the algorithm that will be translated into actual code.

After completing this module, you will be able to:

Transfer problem statements into pseudo code

Create algorithms

Translate pseudo code into programming code

Create simple algorithms in code

Create data structures to store data

Lessons

Understand How to Write Pseudo Code

Algorithm Examples

Introduction to Data Structures

Lab : Working with Algorithms and Data Structures

Module 5: Error Handling and Debugging

This module helps students understand that errors are a part of programming and they must understand how to anticipate errors, handle those errors in code, and present a good user experience with a program. This module introduces structured exception handling as the mechanism to deal with errors. After completing this module, students will be able to:

Implement structured exception handling

Debug applications by using Visual Studio

Lessons

Introduction to Program Errors

Introduction to Structured Error Handling

Introduction to Debugging in Visual Studio

Lab : Implementing Debugging and Error Handling

Module 6: Introduction to Object-Oriented Programming

This module covers an introduction to the concepts related to object-oriented programming (OOP). The content has been split across two modules with this module focusing on basic OOP concepts that will provide sufficient knowledge to understand complex data structures starting with structs and then moving onto classes. This module helps the students gain an understanding of how to encapsulate data and related functionality within a class. After completing this module, students will be able to:

Create and use structure types

Create and use basic class files

Choose when to use a struct vs a class

Lessons

Introduction to Complex Structures

Introduction to Structs

Introduction to Classes

Introducing Encapsulation

Lab : Implementing Complex Data Structures

Module 7: More Object-Oriented Programming

This module teaches students about inheritance and polymorphism in classes and function overloading. Function overloading and polymorphism can go hand-in-hand as often times when you inherit from a class, you want to override or change the existing behavior to suit the needs of you class.

The module also provides an introduction to the base class library in the .NET Framework so that students can start to think about the existence of functionality in other class files and how they can search the .NET Framework to find this functionality and take advantage of it. After completing this module, students will be able to:

Use inheritance in OOP

Implement polymorphism in your classes

Describe how the base class library is constructed

Find class information by using the Object Browser

Lessons

Introduction to Inheritance

Introduction to Polymorphism

Introduction to the .NET Framework and the Base Class Library

Lab : Implementing Inheritance

Lab : Implementing Polymorphism

Module 8: Introduction to Application Security

This module helps students think about security in their applications. This module introduces the concepts of authentication for users and also introduces the concept of permissions for the code that is running on a computer. It explains that operating systems might prevent certain aspects of the program from executing, such as saving a file to a directory to which the user running the app might not have permission to write. The module briefly covers code signing and why programmers might

want to consider using code signing. After completing this module, students will be able to:

Describe how authorization and authentication work

Describe how to apply access permissions for executing code on a computer

Explain how code signing works

Lessons

Authentication and Authorization

Code Permissions on Computers

Introducing Code Signing

Module 9: Core I/O Programming

This module introduces some core input/output (I/O) concepts that programmers will use while creating applications. Starting with console I/O, this module introduces input and output to the Console window.

The module also talks about reading and writing files, which is an important concept to know because applications work with the files on the disk systems on computers. After completing this module, students will be able to:

Read input from a console

Output data to the console

Read and write text files

Lessons

Using Console I/O

Using File I/O

Lab : Core I/O Programming

Module 10: Application Performance and Memory Management

This module enables students understand that memory on a computer is a finite resource. It talks about how good application design and good coding discipline with memory conservation and

memory management will help programmers learn to develop applications that users will like. This is because these applications will be fast, responsive, and do not negatively impact other applications.

After completing this module, students will be able to:

Implement value and reference types correctly in an application

Convert between value types and reference types

Use the garbage collector

Lessons

Value Types vs Reference Types

Converting Types

The Garbage Collector

Lab : Using Value Types and Reference Types

Additional Course Details

Nexus Humans Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975) training program is a workshop that presents an invigorating mix of sessions, lessons, and masterclasses meticulously crafted to propel your learning expedition forward.

This immersive bootcamp-style experience boasts interactive lectures, hands-on labs, and collaborative hackathons, all strategically designed to fortify fundamental concepts.

Guided by seasoned coaches, each session offers priceless insights and practical skills crucial for honing your expertise. Whether you're stepping into the realm of professional skills or a seasoned professional, this comprehensive course ensures you're equipped with the knowledge and prowess necessary for success.

While we feel this is the best course for the Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975) course and one of our Top 10 we encourage you to read the course outline to make sure it is the right content for you.

Additionally, private sessions, closed classes or dedicated events are available both live online and at our training centres in Dublin and London, as well as at your offices anywhere in the UK, Ireland or across EMEA.

Frequently Asked Questions

Q: What delivery options are available for Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975)?

We offer multiple delivery formats:

- Live Instructor-Led Classroom Online (Virtual/Live Online)
- Traditional Instructor-Led Classroom Training (ILT)
- On-site delivery at your offices anywhere in United Kingdom
- Private dedicated courses customized for your team

Q: How many CPD hours does this course provide?

The 5-day Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975) course provides up to 32.5 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975) training?

The training takes place over 5 day(s), with each day lasting approximately 40.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975)?

Yes, we provide corporate training, dedicated training, and closed classes for Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975). Training can take place anywhere in United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online allowing teams from across United Kingdom or internationally to attend.

Q: What related terms do people search for?

Popular related searches include: C# / .Net

Q: Why choose Nexus Human for Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975)?

Nexus Human is recognized as one of the leading training providers. Our trainers have won multiple awards including:

- Small Firms Best Trainer Award
- National Training Partner of the Year (Ireland) - Multiple Years
- Global Top 30 Instructor Awards (2012, 2019, 2021)
- Tech Excellence Award Nominations
- Learning Performance Institute (LPI) External Training Provider Sponsor 2024

Q: Are there any discount codes available?

Yes! Use discount code **PENPAL5** when booking your Introduction to Programming in C# / .Net Core Basics for Non-Developers (TTCN10975) training. Please note that only one discount code can be used per booking and cannot be combined with other special offers.

Nexus Human

Professional Training & Development

Email: info@nexushuman.com

Website: www.nexushuman.com

Phone: +353 1 889 8700 (Ireland) | +44 20 8142 8961 (UK)

© 2026 Nexus Human. All rights reserved. This brochure was generated on 14/11/2026.