Learning to Program with C#

Duration: 35 hours

Prerequisites: Familiarity with computers.

Course Description: This hands on C# programming course provides an introduction to programming using the C# language. 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 Learn

- 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

Overview

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 `foreach`
- 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<TKey, TValue>`
- 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

Accessing Databases Using ADO.NET

- Understanding the ADO.NET Object Model
- Opening Connections

- Working with Controls
  - Using Labels and Textboxes
  - Using Buttons
  - Using Checkboxes and Radio Buttons
  - Using Menus
  - Using List Controls
- Handling Events

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 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`

- Executing Queries Using Commands
- Iterating Through Results Using `DataReaders`
- Calling Stored Procedures
- Passing Parameters to Stored Procedures
- Working with `DataSets`
- Binding Data to Controls

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

Building N-Tier Applications

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

Related Bootcamp

Track	Duration	Price
Microsoft .NET Developer: C#	161 hours	$4,475

Contact Us

Address: 6 Hemlock Drive Chelmsford, MA 01824

Phone: 978.250.4983

Mon - Thur: 9 am - 5 pm EST
Fri: 9 am - 4 pm EST

E-mail: info@developer-bootcamp.com

Copyright© 2025 Developer Bootcamp