

10975: Introduction to Programming

Duration: 5 Days

Method: Instructor-Led Training (ILT)

Course Description

In this course, participants will learn the basics of computer programming through the use of Microsoft® Visual Studio® and either the Visual C#® or Visual Basic® programming languages. The focus will be on core programming concepts such as computer storage, data types, decision structures, and repetition by using loops. The course also covers an introduction to object-oriented programming with topics such as classes, encapsulation, inheritance, and polymorphism. Other topics that are also covered include exception handling, application security, performance, and memory management.

Target Audience

This course is intended for:

- Anyone who is new to software development and wants, or needs, to gain an understanding of programming fundamentals and object-oriented programming concepts.
- Persons who want to gain an understanding of the core programming fundamentals before moving on to more advanced courses such as 20483: *Programming in C#*.

Prerequisites

To attend this course, candidates must be able to:

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

Course Objectives

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

- Explain the application security concepts.
- Create and use classes in a computer program.
- Explain pseudocode and its role in programming.
- Explain computer number systems such as binary.
- Implement object-oriented programming concepts.
- Create and use variables and constants in programs.
- Explain how to create and use functions in a program.
- Identify the performance considerations for applications.
- Create and use repetition (loops) in a computer program.
- Implement encapsulation, inheritance, and polymorphism.
- Create and use decision structures in a computer program.
- Describe the Base Class Library (BCL) in the .NET Framework.
- Implement simple Input/Output (I/O) in a computer program.
- Explain core programming fundamentals such as computer storage and processing.
- Explain the basic computer data structures such as arrays, lists, stacks, and queues.
- Identify application errors and explain how to debug an application and handle errors.



Course Topics

Module 1: Introduction to Core Programming Concepts

- Computer Data Storage and Processing
- Application Types
- Application Life-Cycle
- Code Compilation

Module 2: Core Programming Language Concepts

- Syntax
- Data Types
- Variables and Constants

Module 3: Program Flow

- Introduction to Structured Programming Concepts
- Introduction to Branching
- Using Functions
- Using Decision Structures
- Introducing Repetition

Module 4: Algorithms and Data Structures

- Understand How to Write Pseudo Code
- Algorithm Examples
- Introduction to Data Structures

Module 5: Error Handling and Debugging

- Introduction to Program Errors
- Introduction to Structured Error Handling
- Introduction to Debugging in Visual Studio

Module 6: Introduction to Object-Oriented Programming

- Introduction to Complex Structures
- Introduction to Structs
- Introduction to Classes
- Introducing Encapsulation

Module 7: More Object-Oriented Programming

- Introduction to Inheritance
- Introduction to Polymorphism
- Introduction to the .NET Framework and the Base Class Library

Module 8: Introduction to Application Security

- Authentication and Authorization
- Code Permissions on Computers
- Introducing Code Signing

Module 9: Core I/O Programming

- Using Console I/O
- Using File I/O

Module 10: Application Performance and Memory Management

- Value Types vs Reference Types
- Converting Types
- The Garbage Collector

LABS INCLUDED

