

55337AC Introduction to Programming

Duration: 5 Days

Course Description:

This five-day course introduces students to the basics of computer programming through the use of Microsoft Visual Studio 2022 and the Visual C# and Visual Basic programming languages. It covers core programming concepts such as computer storage, data types, decision structures, repetition by using loops, and an introduction to object-oriented programming.

This course is designed to provide a comprehensive introduction to the core concepts of programming. It begins with an exploration of how computers process information, the types of applications that programmers might create, and how code is compiled and interpreted by a computer. Students will learn about computer data storage, application types, the application lifecycle, and code compilation. By the end of the first module, students will be able to explain core programming fundamentals, create and use variables and constants in programs, and create and use decision structures and loops in a computer program.

The second module delves into the syntax of programming languages and the importance of using good syntax. It also discusses the core data types and how to store these data types in computer memory using variables and constants. Students will learn to define syntax, explain the different types of core data used in programs, and declare and use variables and constants in a computer program.

The third module covers how code is executed in a computer program, including structured programming and branching in code execution. Students will learn these concepts using functions, decision structures, and looping constructs. They will be able to describe structured programming, create and use functions in their code, and create and use decision structures and looping structures.

The fourth module introduces the concept of an algorithm and how to translate these set of steps into pseudocode for evaluation of the algorithm that will be translated into actual code. Students will learn to transfer problem statements into pseudo code, create algorithms, translate pseudo code into programming code, create simple algorithms in code, and create data structures to store data.

The fifth module helps students understand that errors are a part of programming. They will learn how to anticipate errors, handle those errors in code, and present a good user experience. This module introduces structured exception handling as a mechanism to deal with errors. Students will learn to implement structured exception handling and debug applications by using Visual Studio 2022.

The sixth and seventh modules cover an introduction to the concepts related to object-oriented programming (OOP). Students will gain an understanding of how to encapsulate data and related functionality within a class. They will learn about inheritance and polymorphism in classes and function overloading. The module also introduces the base class library of .NET so that students can start to think about the existence of functionality in other class files and how they can search the .NET libraries to find this functionality and take advantage of it.

The eighth module helps students think about security in their applications. It introduces the concepts of authentication and authorization for users, and also introduces the concept of permissions for running code. 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 it.

The ninth 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 from and writing to the filesystem. Students will learn to read input from a console, output data to the console, and read and write text files.

The final module enables students to understand that memory on a computer is a finite resource. It talks about how good application design and good coding discipline with memory management will help programmers learn to develop applications that are fast, responsive, and do not negatively impact other applications. Students will learn to implement value and reference types correctly in an application, convert between value types and reference types, and use the garbage collector.

Prerequisites:

Before attending this course, students should 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 Objectives:

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

Course Outlines:

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.

Lessons

- Computer Data Storage and Processing
- Application Types
- Application Lifecycle
- Code Compilation

REGISTER NOW!

training@trends.com.ph
(+632) 8863-2123
www.trendssacademy.com.ph

COURSE OUTLINE

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.

Lessons

- Syntax
- Data Types
- Variables and Constants

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 using functions, decision structures, and looping constructs.

Lessons

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

Module 4: Algorithms and Data Structures

This module introduces the concept of an algorithm by 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 pseudocode for evaluation of the algorithm that will be translated into actual code.

Lessons

- Understand How to Write Pseudocode
- Algorithm Examples
- Introduction to 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. This module introduces structured exception handling as a mechanism to deal with errors.

Lessons

- Introduction to Program Errors
- Introduction to Structured Error Handling
- Introduction to Debugging

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 on to classes. This module helps the students gain an understanding of how to encapsulate data and related functionality within a class.

Lessons

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

Module 7: More Object-Oriented Programming

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

The module also introduces the base class library of .NET so that students can start to think about the existence of functionality in other class files and how they can search the .NET libraries to find this functionality and take advantage of it.

Lessons

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

Module 8: Introduction to Application Security

This module helps students think about security in their applications. It introduces the concepts of authentication and authorization for users, and also introduces the concept of permissions for running code. 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 it.

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 from and writing to the filesystem.

Lessons

- Using Console I/O
- Using File I/O
- Read and write text files

Module 10: Application Performance and Memory Management

This module enables students to understand that memory on a computer is a finite resource. It talks about how good application design and good coding discipline with memory management will help programmers learn to develop applications that are fast, responsive, and do not negatively impact other applications.

Lessons

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

REGISTER NOW!

training@trends.com.ph
 (+632) 8863-2123
 www.trendssacademy.com.ph