Introduction to Programming (MS-10975)

Modality: Virtual Classroom

Duration: 5 Days

SATV Value: 5

About this course:

This is an excellent course for beginners with little to no prior programming skills. The students who enroll get to learn the fundamentals of programming along with core concepts. Basic learning in this course includes loops, functions, inheritance, classes, polymorphism, and encapsulation. Students also get to dive deeper into programming and learn the storage of data, data handling, data types, exception handling, memory allocation/management, and design structures as well.

Microsoft Visual Studio 2013 is the main IDE that will be used and students get to learn all these programming skills in either Visual Basic or C#.

Course Objective:

The objective of this course is to provide knowledge and skills on the following.

- Basic programming knowledge and fundamentals
- Loops, functions, and algorithms
- Computer storage and processing
- Variables, constants and number systems
- Explain in-depth pseudo code
- Provide knowledge on arrays, lists, queues, and stacks
- Concept of classes and their importance
- Object-oriented programming concepts
- Some basic level security concepts in application development
- Explain the base class library in .NET
- Error and exception handling, and error classification
- Performance enhancement in applications

Audience:

This course has a very vast audience pool of different ages. Students of high school, college or universities can opt-in to get a basic programming skillset that can be further enhanced. The course is not limited to just school students, many people who are looking for a career change can get started quickly with this course and in very little time they can find calling themselves developers (or junior developers).

Anyone willing to get started with computer application development can enroll and gain sufficient knowledge to advance their skills further and then can take more intermediate-level courses like (20483B - Programming in C#).

Prerequisite:

Programming is usually mistaken to be something very hard that people might not understand or might need some extra skills to learn. In reality, it is not, you only need these few things to get started.

- Know-how of how a personal computer works, able to use the Windows environment
- A creative and eager-to-learn mindset with constructive thinking
- Ability to learn new things and excel, and also question a lot as well (how things work, what other ways can the same task be done with fewer resources, etc)

Course Outline:

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 Life-Cycle
- Code Compilation

Lab : Thinking Like 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

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

Lab : Working with Data Types

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

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.

Lessons

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

Lab : Creating Functions, Decisions, and Looping

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

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.

Lessons

- Understand How to Write Pseudo Code
- Algorithm Examples

- Introduction to Data Structures

Lab : Working with Algorithms and Data Structures

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

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.

Lessons

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

Lab : Implementing Debugging and Error Handling

After completing this module, students will be able to:

- Implement structured exception handling
- Debug applications by using Visual Studio 2013

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.

Lessons

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

Lab : Implementing Complex Data Structures

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

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 your 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.

Lessons

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

Lab : Implementing Inheritance

Lab : Implementing Polymorphism

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

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.

Lessons

- Authentication and Authorization
- Code Permissions on Computers
- Introducing 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

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.

Lessons

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

Lab : Core I/O Programming

After completing this module, students will be able to:

- Read input from a console
- Output data to the console
- Read and write text files

Module 10: Application Performance and Memory Management

This module enable 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 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.

Lessons

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

Lab : Using Value Types and Reference Types

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