**C# 7.0**

**Contents:**

* Introduction to 4.6 .NET framework
  + What is .NET Platform?
  + What is .NET Framework
  + .NET Framework, Languages, and Tools
  + .NET Framework Major Components
  + Common Language Runtime (CLR)
  + Compilation and Execution in .NET
  + Understand the .NET Framework 4.6 stack
* Introduction to C# 7.0
  + Features of C#
  + C# Compilation and Execution
  + General Structure of a C# Program
  + Overview of Visual Studio
  + Creating and Using a DLL
* Data Types and Arrays in C#
  + Data Types in C#
  + Value Types and Reference Types
  + Boxing and UnBoxing
  + Single Dimensional, Multi-Dimensional & Jagged arrays
  + Nullable Types
  + Implicitly Typed Local variables
  + Var vs dynamic
  + Is and as operator
  + Ref vs out keywords
  + The ‘object’ base class in .net
  + Equals() vs ==
  + String vs StringBuilder
  + Various String class methods
  + Default parameters, named parameters
  + Parse() vs TryParse() vs Convert Class methods
* OOP with C#
  + Structures and enums
  + The architecture of a class in C#
  + Instance, Class & Reference variables
  + Access Modifier
  + Abstract Classes
  + Constructors, Destructors, The GC
  + .NET Base class library
  + OOP with C# - Continued from Day 2
  + Inheritance in C#
  + Method Overloading
  + Method Overriding
  + Operator Overloading
  + Method Hiding
  + Access modifies: private, pubic, protected, internal, protected internal, new
  + Abstract classes
  + Sealed classes
  + Creating Interfaces
  + Implementing Interface inheritance
  + Declaring properties within Interfaces
  + Namespaces
  + Creating and using Generic classes
* Exception Handling
  + Exceptions in C#
  + Exception class hierarchy
  + Try block
  + Multiple catch blocks
  + Finally block
  + Purpose of throw keyword
  + Purpose of inner exception
  + Creating Custom Exception
* Overview of Garbage Collection in C#
* Collections
  + System.Collections Namespace
  + Collection Interfaces
  + Collection Classes
  + The collection API
* Introduction to Debugging
  + Tracing, Debugging, Build
  + Using break points
  + Using break conditions
  + Using watch and output window
  + Creating multiple projects within one solution