Course Outline

**Module 01: Basics of Computer Architecture Hardware and software**

 Role of CPU, Memory

 Role of Operating System in Detail

Programming Language Journey (Advantage and Limitation) Features in Detail

 Machine Language

 Assembly language

 Procedural Language-C language

 Object-Oriented Language-C++

 GUI Based Programming Language-Visual Basic

 Java and .Net Language-C# Comparison in brief

Introduction to Development Models

 Application Development Models

 The Desktop models

 The Client-Server model (Tier Architecture till N-Tier)

 The Web models

**Module 02: Programming Fundamentals and OOPS Concept**

 Pseudo Code Algorithm and Flow Chart

 Programming Constructs

 Software Engineering Principles

 Discussing Object Oriented Approach

 Procedural Programming vs Object Oriented Programming

 OOPS Fundamentals and Pillars Discussion

 Introducing GIT Repository

 Creating Git Repository

**Module 03: Overview of the Microsoft .NET Framework**

 Introduction to the .NET Platform and .NET Framework

 Framework Class Library, ADO.NET Library and ASP.NET Library

 Understanding the Common Type System (CTS)

 Introduction to the Need for the Common Language Runtime (CLR)

 Components of CLR and Roles

 JIT Compiler

 Type Checker

 Exception Manager

 Security Checker

Com Marshaler

 Thread Support

 Garbage Collector

 Code manager

 Class Loader

 Managed Code Vs. Unmanaged Code

 Understanding the Just-In-Time (JIT) Compilation Process

 MSIL Code

 Metadata- The Self Explanatory Files

 Extracting IL Code and Viewing Metadata using ILDASM Tool

 .NET Assemblies Explanation

 Difference between .NET Exe File and Other Exe, DLL Description

 Overview of the .NET-Based Languages

 Comparison of the .NET-Based Languages

 The .NET Framework – Versions

**Module 04: Using Microsoft Visual Studio .NET 2019**

 Overview of Visual Studio .NET IDE Features

 Properties Window

 Tool Box

 Solution Explorer

 Server Explorer

 Object Browser

 Editor Browser

 Creating a Console Application Project

 Creating Windows Application Project

 Compiling Running and Debugging Application

 Folder Structure and File Types Created by Application

**Module 05: C# Language Fundamentals**

 Structure of a C# Program

 Basic Input/Output Operations

 Commenting a Program

 Recommended Practices

**Module 06: Using Value-Type Variables in C#**

 Naming Variables

 Best Practices for Naming Conventions

 Using Built-In Data Types

 Creating User-Defined Data Types

 Converting Data Types

 Typecasting

 Boxing and Un-boxing data types

**Module 07: C# Statements**

 Introduction to Statements

 Using Selection Statements

 Using Iteration Statements

 Using Jump Statements

 Using Conditional Statements

 Applications Based on All Statements

**Module 08: Essentials of Object-Oriented Programming in C#**

 Understanding Namespaces

 Understanding Scope Resolution

 Defining Classes

 Instantiating and Working with Objects

 Difference between Abstraction and Encapsulation

 Understanding and Implementing Encapsulation

 Defining Object-Oriented Systems

**Module 09: String and Arrays in C#**

 String Handling

 The String and String Builder Class

 Different Methods and Properties of String and String Builder Class

 Arrays

 Overview of Arrays

 Creating and using Single Dimension and Multi Dimension Arrays

 Jagged Arrays

 Using foreach Loop

 Param Keyword

Module 10: Methods and Parameters using C#

 Using Methods

 Using Parameters

 Passing value type parameters

 Passing Reference types (string, Array, object) as parameters

 Passing Parameters using Ref and Out keyword

 Static and Instance Members

 Explaining Constant and Read-only

**Module 11: Creating Objects in C#**

 Using Constructors

 Using Initializer Lists

 Initializing Data

**Module 12: Properties and Indexers in C#**

 Data Fields

 Properties

 Using Indexers

 Compare and Contrast between Properties and Indexers

**Module 13: Inheritance in C#**

 Deriving classes

 Understanding Type Hierarchy

 Hiding Base Class Member in Derived Class

 Implementing Multi Level Inheritance

 Using base keyword

 Using Static Classes

**Module 14: Access Modifiers and Constructor**

 Access Modifiers in C#

 Default Accessibility Level for Class, Methods and Structures

 Constructor Execution Sequence in Inheritance Scenario

 Default Constructor and Parameterized Constructor

 Constructor in Structure in C#

 Calling Base Class Constructor in Derived Class Constructor

 Using this keyword to Call One Constructor by another Constructor

 Discussing Public, Private and Protected Constructor

 Understanding Static Constructor

 Differentiating Static Constructor and Instance Constructor by call Mechanism

 Implementing Singleton Design Pattern and Understanding Static Class

**Module 15: Polymorphism in C#**

 Polymorphism Using Methods

 Overloading a Method

 Overriding Virtual Method

 Abstract Class and Abstract Method

 Interface Implementation

 Interface Inheritance and Implementation

 Using Sealed Class and Sealed Method

 Discussed Inheritance and Interface Implementation in Structure

 Discussion to Differentiate Virtual Method, Abstract Method and Interface

**Module 16: Operators and Equality Comparison**

 Introduction to operators

 Operator overloading

 Equality Comparison Operators and Methods

 Comparing Value Equality

 Comparing Reference Equality

 Using ==, Equals, ReferenceEquals, CompareTo

 Comparison by GetHashCode Method

 Overriding Methods and Operators for Equality Comparison

 Overriding To String Method

**Module 17: Exception Handling**

 Checked and Unchecked Statements

 Try, Catch and Finally

 Creating Custom Exception

 Exception Handling Best Practices

 Do’s and Don’ts of Exception Handling

**Module 18: Collection Classes in C#**

 Understanding Collection

 Using Different Collections viz. ArrayList, Stack, Queue, SortedList

 Understanding Different Interface viz. IEnumerable,

 IEnumerator, IComparable, IComparer, IList, IEquatable

 Hashing Mechanism

 Generic Collection Classes

 Performance Improvement using Generic Collection over Non-Generic version

Module 19: Delegates and Events in C#

 Creating and using Delegates

 Multicast Delegates

 Anonymous Method

 When to Use Delegates, Events and Interfaces

 Covariance and Contravariance in Delegates

 Generic Delegates

 Comparing C++ Templates and C# Delegates

 Implementing Polymorphism Using Delegates

 Defining and using Events

 Creating Custom Events and Using it

 Passing Event Arguments

**Module 20: Creating Windows Applications**

 Creating a Windows Form

 Windows Form Controls

 Writing Code for Control Events

 Understanding Delegates and Events Implemented in windows Forms

 Writing a Common Method Called for Click Event of Multiple Buttons

**Module 21: Destroying Objects and Resource Management in C#**

 Objects and Memory

 Using Destructors

 Destroying Objects

 Programming for the Garbage Collector

 Implementing the IDisposable Interface

 Understanding and Implementing Object Pooling

 Cloning Objects implementing Shallow and Deep Copies

**Module 22: File Handling**

 FileSystemInfo Base Class, FileInfo Class and their Members

 Streams

 Reader/Writer

 Basic File IO

**Module 23: Serialization**

 Serialization Scenarios

 Serialization Attributes

 Object Graph

 Serialization Process

 Serialization Example

 Deserialization Example

**Module 24: Threading**

 Threading & Synchronization

 Life cycle of a thread

 Different Thread Methods and Properties

 Synchronizing critical data using Synchronization objects

 Thread Pool

**Module 25: Language Enhancements in C# 2.0**

 Static Classes

 Property Accessors

 External Aliases

 Nullable types

 Iterators

 Partial types

 Generics

**Module 26: Language Enhancements in C# 3.0**

 Implicitly typed local variables

 Anonymous Types

 Extension Methods

 Object and Collection Initializer

 Lambda Expressions

 Query Expressions

 Expression Trees

**Module 27: Language Enhancement in C# 4.0, 5.0 and .NET 4.5**

 Named and Optional Parameters

 Co and Contra variance

 Dynamic Typing and Late Binding

 Parallelization Overview

 Task Parallel Library

 Threads Vs. Tasks

 Parallel Extensions in .NET 4.5

 Async and await in C# 5.0

**Module 28: C# 6.0 Features**

 using Static.

 Auto property initializer.

 Dictionary Initializer.

 nameof Expression.

 New way for Exception filters.

 await in catch and finally block.

 Null – Conditional Operator.

 Expression – Bodied Methods

 Easily format strings – String interpolation

**Module 29: C# 7.0 Features**

 Tuples

 Deconstruction

 Non-'NULL' able reference type

 Minimizing Out variables

 Patterns Matching

 Readability Improvements with Literals

 Local functions

 Expression-bodied members

 Throw Expressions

**Module 30: .NET Remoting**

 Understanding Application Domain and Remoting Architecture

 Accessing .NET Components Across Application Domain

 .NET Remoting architecture

 Creation of Proxy Objects by the CLR

 Using the Channel Services to Transport Remote

 Object Across Application Domains

 Using TCP Channel, Using HTTP Channel

 Formatter for Creating Message and Encoding it

 Soap Formatter

 Binary Formatter

 Activation Model

 Server Activated

o Creating a Single Call Object

o Creating a Singleton Object

 Client Activated

o Managing Lifetime with Lease Manager

 Hosting .NET Remote Component

 Using Framework Classes

 Using Configuration Files

**Module 31: Design Patterns and Principles**

 Understanding SOLID Design Principles

 Understanding Design Patterns and its real use

 Difference between Design Principles and Patterns

 Types of Design Patterns (Creational, Behavioural, Structural)

 Knowing and Implementing Few Important Design Patterns (Singleton, Factory,

Abstract Factory)

 Design Patterns implemented in .NET Framework

**Module 32: Dependency Injection using Unity Framework**

 Dependency Inversion Principle

 Inversion of Control

 Dependency Injection

 Creating a console application

 Adding Reference to Microsoft Unity Framework

 Adding BL class

 Adding DL class

 Adding Interface

 How to configure the Unity Container

 Running and debugging an application

 Final output

 Dependency Injection Pros and Cons.

**Module 33: Unit Testing with TDD approach**

 What is Unit Testing

 Advantages of Unit Testing

 Unit Testing C# Code using tools like Nunit

 MS Test Testing Framework

 Working with MS Test

 Microsoft Test Framework (Mock Framework)

 TDD Approach of Software Delivery

 Understanding SDLC Process Models (Waterfall Model, Agile Process Model)

**Module 37: SQL Server Overview**

 What is SQL Server

 Advantages of SQL Server 2012

 SQL Server architecture

 SQL Server security Model

 SQL Server System databases

**Module 38: SQL Server Tools**

 Server Tools

 SQL Server manager

 SQL Server Agent

 Server Network Utility

 Client Tools

 SQL Enterprise Manager

 SQL Query Analyzer

 Client Network Utility