C# Programming Syllabus

Overview: Today C# is considered to be the most popular and modern Programming language. It belongs to "C" family and inherently has lots of things carried from C programming language. It is the ideal choice of all .net developers for the reason that Microsoft has developed C# with features of popular languages to develop different types of .net applications. It has SIMPLICITY of Java, POWER of C++ and PRODUCTIVITY of VB.

Module 1:- Introduction to .Net Framework

The Microsoft .NET Framework is a platform for building, deploying, and running Web Services and applications. It consist of components such as common language runtime (CLR) and the .NET Framework class library, which includes classes, interfaces, and value types that support wide range of technologies.

- ϖ What is a .Net Framework and components in the .Net Framework
- ϖ Different .Net Framework versions and their Dependency
- ϖ The core of the .Net Framework and the Types of .Net Applications that we can develop
- ϖ What are Base class Libraries and What is a Namespace
- ϖ How the Compilation process and Execution Process is done
- ϖ What is Portable Executable and its extensions
- ϖ What is MSIL and Why MSIL instructions are Platform Independent Instructions
- ϖ What is Metadata and which type of Information does the Metadata Stores
- ϖ What is CLR and What are the Components in CLR Module

Module 2:- VS.Net and Entry point Method

Visual Studio .NET is Microsoft's visual programming environment for creating Web services based on use of the Extensible Markup Language (XML).It comes with the .NET Framework, including the Common Language Runtime, and includes several programming languages including Visual Basic, Visual C++, and Visual C#.

- ϖ Importance of Command Line arguments and How to pass values for arguments through Command prompt and through command Line arguments in the visual studio
- ϖ Different Entry point methods and Significance of the Return value in main
- ϖ How to resolve ambiguity of Main method ϖ How to develop an application without using Visual studio .Net Module

Module 3:- C# Language Syntax

C# is an object-oriented programming language. In Object-Oriented Programming methodology, a program consists of various objects that interact with each other by means of actions. In this Module

we concentrated on Introduction to C#, its Evolution and its versions History along with that We understand

- ϖ Why we need a programming Language
- ϖ What are the Data Types we have in C# and How to declare a Variable
- ϖ How Data Types are Categorized into Value Type and Reference Type
- ϖ What is Implicit Casting and Explicit casting and How to handle Overflow checks
- ϖ .Difference between string and string Builder ϖ What is Boxing
- ϖ What is Unboxing
- ϖ What is Type Inference ϖ What are constants and Enums ϖ What are the Operators we have in C#
- ϖ How the if, while, do while, switch condition will works
- ϖ What is the difference between for and foreach and where to use for loop and where to use foreach loop
- ϖ What is single dimension Array, multi dimension Array
- ϖ What is method overloading
- ϖ What are optional parameters and What will happen When we not provide any value for the parameter
- ϖ What are Named Arguments
- ϖ What is params Parameter ϖ How to Pass argument by value, ref and out
- ϖ How to improve our Programming skills and logical skills to become a extraordinary programmer

Module 4:- OOPS – Concepts Object-oriented programming

(OOPs) is the core ingredient of the .NET framework. OOPS is so important that, before embarking on the road to .NET, you must understand its basic principles and terminology to write even a simple program.

- ϖ Introduction to OOPS and its principles
- ϖ What is a class
- ϖ What is an object
- ϖ What is a component
- ϖ What is Encapsulation and Data Abstraction
- ϖ What is an inheritance and advantages of inheritance
- ϖ What is a polymorphism

Module 5:- OOPs - Programming Encapsulation

The need of encapsulation is to protect or prevent the code (data) from accidental corruption due to the silly little errors that we are all prone to make. In the program development and data is packed closely to the functions that operate on it and protects it from accidental modification from outside functions. ϖ How to create a WindowsForms application

- ϖ How to create a class and How to declare field members in it
- ϖ How to Design GUI using Controls in the ToolBox

- How button click event works
- How Garbage collector will destroy the objects and What are the generations in Garbage Collector ϖ What is an instance Method and What is the use of this keyword inside a method
- What are properties and What does a get and set block do
- What is the difference between constructor and Destructor
- Where the static members allocate memory
- When the memory is allocated for static members ϖ How to access a static member
- What is the role of Static constructor and How it executes
- When to declare a class as static

Module 6:- OOPs – Inheritance

One of the most important concepts in object-oriented programming is inheritance. Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and speeds up implementation time.

- What is Protected keyword and How to bypass it through child class
- How to casting the reference types ϖ What does a "is" operator do
- What does "as" operator do
- What does "??" operator do
- What is static Binding and Dynamic Binding
- How to override a method
- What is an abstract class , abstract method
- When to declare a class as abstract
- What is the difference between abstract class and concrete class
- When to declare a method using new keyword
- What is a system. object class
- What are the methods in the object class

Module 7:- OOPS - Interface and Polymorphism

The word polymorphism means having many forms. In object-oriented programming paradigm, polymorphism is often expressed as 'one interface, multiple functions'. An interface is defined as a syntactical contract that all the classes inheriting the interface should follow.

- What is an interface
- How does multiple inheritance is working with interfaces
- How to solve if two interfaces having same method name
- What is publicly implemented and Explicitly implemented
- Why does the .net doesn't support multiple inheritance using classes
- How to implement an interface by inheriting it

Module 8:- Collections and Generics

Collection classes are specialized classes for data storage and retrieval. These classes provide support for stacks, queues, lists, and hash tables. Most collection classes implement the same interfaces. Generics allow you to delay the specification of the data type of programming elements in a class or a method, until it is actually used in the program.

- What are the Types of collections and What is IEnumerable, ICollection, IList, IDictionary
- What is ArrayList, HashTable, SortedList, Queue, Stack
- How to iterate using IEnumerable
- How sort using IComparer and IComparable
- What are the advantages of Generics and How they work at Runtime
- What are Generic methods and Generic collection classes
- What is List and Dictionary

Module 9:- Exception Handling

This features help you deal with any unexpected or exceptional situations that occur When a program is running. Exception handling uses the try, catch, and finally keywords to try actions that may not succeed, to handle failures When you decide that it is reasonable to do so, and to clean up resources afterward.

- What is an Exception and types of Exceptions
- How to handle Exception using try and catch blocks ϖ How to throw an Exception using throw ex and throw
- What is finally Block
- How to define custom Exception class

Module 10:- IO Streams

C# includes following standard IO (Input/Output) classes to read/write from different sources like a file, memory, network, isolated storage, etc.The System.IO namespace has various classes that are used for performing numerous operations with files, such as creating and deleting files, reading from or writing to a file, closing a file etc.

- What is a Stream and Types of Streams
- What are standard IO streams
- How Files can be Handled using FileMode, FileAccess, FileShare
- What is Binary Reader and Binary Writer
- How to work with File System
- What is Serialization and Deserialization

Module 11:- Developing GUI Application Using WinForms

Windows Forms is a graphical (GUI) class library included as a part of Microsoft .NET Framework,[1] providing a platform to write rich client applications for desktop, laptop etc. A Windows forms application will normally have a collection of controls such as labels, textboxes, list boxes, etcWhat are Windows Forms and How they bring Rich GUI to the Application

- What are the controls that have in the WindowsForms
- What are the important properties of the controls
- What are the important Events that each control have
- What are the Container controls
- What are Graphical Objects
- What are GDI objects
- What is MenuStrip, ContextMenuStrip, ToolStrip And StatusStrip
- How to work with Model Dialog
- How to develop a Notepad Application
- What is Modeless dialog Box
- What is Multiple Document Interface
- What is Form Inheritance
- How to Add Login Facility to the Application
- How to work with the Resource files
- What is NotifyIcon Control
- What is Timer control
- How to Drag and Drop the Files
- What is a Treeview
- What is a ListView

Module 12:- ADO.NET

Part1 - Managed Provider Objects ADO.NET provides consistent access to data sources such as SQL Server and XML, and to data sources exposed through OLE DB and ODBC. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, handle, and update the data that they contain.

What is a Manage Provider and important objects in it?

- How to Install SQL server and Management Studio
- How to establish a connection to Database
- What is Connection Pooling
- How to insert, Update, Delete the data in the Database from the Application
- How to Fetch Data from the Database using Select command
- How to implement Login to the Application using Database
- What is MultipleActiveResultSets
- What is Parameterized Prepared Statement
- How to write stored procedures in Backend
- How to Execute storedprocedures from front end Application
- What are the Transactions ϖ How to Manage the Transactions using Transaction Scope
- What is Asynchronous Execution of SQL Commands
- How to write Provider independent code
- What is utility class

Module 13:- ADO.NET Part2 –

DataSet Object Model DataSet is tabular representation of data. Tabular representation means it represents data into row and column format. This class is counted in a disconnected architecture in .NET Framework. Which means it is found in the "System.Data" namespace. The Dataset can hold records of more than one Database tables or DataTables.

- ϖ What are DataAdapter events
- ϖ How to handle Concurrency issues if multiple users performing operations on same Data
- ϖ How to sort and filter the data using DataView
- ϖ What are the constrains in the DataTable and How to Add the constrain to the DataTable
- ϖ What is a DataRelation object
- ϖ How to create DataSet/ DataTable Dynamically without using DataAdapter
- ϖ What is Typed Dataset

Module 14:- Windows Services

Windows service is a computer program, which will run in the background.When some action has to be performed at a particular time, or need to be performed continuously in specific time interval without user interaction then the solution is Windows service.

ϖ What is a Windows service ϖ How to create a new windows service Template

ϖ How to Install/Deploy windows service in the OS

ϖ How to Launch a Windows Service

ϖ How to develop an Application for controlling the Service

Module 15:- Delegates & Events

A Delegate is an abstraction of one or more function pointers. The .NET has implemented the concept of function pointers in the form of delegates. With delegates, you can treat a function as data. Delegates allow functions to be passed as parameters, returned from a function as a value and stored in an array.

- ϖ What is a Delegate
- ϖ How to create a chat application using Delegates ϖ How to raise an event using Delegates
- ϖ What are Anonymous Methods

Module 16:- User Control and Custom Control

A control which can reuse the Components in the Applications. The control can be defined in both Xaml and Code-Behind.An UserInterface element that have a distinct behavior which is said as CustomControl.

- ϖ What is User Control
- ϖ What is composite control

- How to inherit the User Control
- What is a custom control

Module 17:- MultiThreading

Every application runs with at least one thread. So what is a thread? A thread is nothing more than a process.he process performs sets of sequential steps, each step executing a line of code. Because the steps are sequential, each step takes a given amount of time.

- What is process and Thread
- What is difference between MultiThreading and Multitasking
- What is scheduling and types of scheduling
- How to set the Thread priority ϖ How to suspend, Resume, Interrupt, Abort and get the status of Thread
- What is cross Thread operation
- What is Thread pool
- What is Thread Synchronization
- What is critical section
- What is Mutex
- What is Semaphore
- What is Task parallel programming
- What is Async Programming

Step-by-Step procedure for building the project from ground up

- Complete Source Code
- Database Script with Sample data
- Instructions to Setup the Project on a Development box
- Instruction to Deploy the project on Production Box / Microsoft Azure

TANMAYA