# Index

## Contents

## IMMERSIVE COURSE STRUCTURE .NET WITH AZURE LOT

.NET with Azure LOT provides exposure to the entire spectrum of .NET/Azure technologies. It focuses on Desktop as well as Cloud application development using .NET Technologies. The following table lists the proposed course structure .NET/Azure LOT.

| Sr. No. | Course | Duration | Remarks |
|---|---|---|---|
| 1 | Discover | 0 | |
| 2 | Soft Skills Part 1 | 1 | Soft Skills Part 1 |
| 3 | .NET Framework Fundamentals | 7 | |
| 4 | .NET Core Basics | 3 | |
| 5 | Git | 1 | |
| 6 | Soft Skills Part 2 | 1 | Soft Skills Part 2 (Saturday) |
| 7 | .NET Core Fundamentals Test | 0.5 | Module Test (MCQ + Coding) |
| 8 | RDBMS and SQL Server | 2.5 | Sprint 1 Implementation |
| 9 | LINQ and Entity Framework Core | 2 | |
| 10 | ASP.NET Core Web API | 3 | |
| 11 | Soft Skills Part 3 | 1 | Soft Skills Part 3 |
| 12 | WEBAPI + SQL Server + LINQ and EF MCQ Test (1 hr on Day1 of Sprint) & Sprint 1 Implementation | 3 | Sprint 1 Implementation + MCQ |
| 13 | Sprint 1 Evaluation | 1 | |
| 14 | Cloud Computing & Microsoft Azure Fundamentals | 6 | |
| 15 | Soft Skills Part 4 | 1 | Soft Skills Part 4 (Evaluation) |
| 16 | Cloud Networking and Storage | 5 | Sprint 2 Implementation |
| 17 | Azure PaaS Services | 6 | |
| 18 | MCQ Test (1 hr on Day 1) + Azure Function App + Azure Service Bus + Azure Logic App Sprint 2 Implementation | 2 | Sprint 2 Implementation + MCQ |
| 19 | Sprint 2 Evaluation | 1 | |
| 20 | L1 Test | 1 | |
| | Total Training Duration | 48 | |

# .NET with Azure Curriculum

# .NET Framework Fundamental

**Program Duration**: 7 Days.

**Table of contents**

    - Customizing Visual Studio Settings – Extensions, NUGet Package, Environmental Settings
- 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
    - Inheritance in C#
    - Method Overloading
    - Method Overriding
    - Operator Overloading
    - Method Hiding
    - Access modifiers : private, pubic, protected, internal, protected internal, new
    - Anonymous types
    - Abstract classes
    - Sealed classes
    - Creating Interfaces
    - Implementing Interface inheritance
    - Declaring properties within Interfaces
    - Namespaces
    - Creating and using Generic classes
    - Indexers & Properties
    - Auto Implemented properties
    - Static Classes
    - Property Accessors
    - Partial types
    - Extension methods
    - Object Initializer
- Evaluating Regular Expressions in C#
    - RegEx Class
    - Forming Regular Expression
    - Methods for Regular Expression
- 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
- Garbage Collection in C#
    - Role of a Garbage Collector
    - Garbage Collection Algorithm
    - Finalize vs Dispose
- Collections & Generics
    - System.Collections Namespace

    - Collection Interfaces
    - Collection Classes
    - The collection API
    - Working with Generics
    - Collection Initializers
    - Iterators
    - Constraints
- File I/O and Serialization
    - Persisting object state to a stream
    - Various classes used for File handling
    - Using StreamReader, StreamWritter
    - Using BinaryReader, BinaryWriter
    - Using File, FileInfo, Directory, DirectoryInfo
    - Serialization modes: Binary, SOAP, XML
    - JSON serialization using DataContractJsonSerializer
    - Performing various serializations,
    - Runtime serialization (Deep)
    - Marking a class serializable
    - Serialization and inherited classes
    - Customizing Serialization by Using attributes
    - Implementing ISerializable interface
- Threading, Parallel and Async programming with C#
    - Appdomain vs Process vs Thread
    - Process vs Thread
    - Creating and running a thread
    - Thread.Sleep() method
    - Parallelization Overview
    - Task Parallel Library
    - Threads Vs. Tasks
    - Parallel Extensions
    - Task Based Asynchronous Model
    - Async and Await
    - Using Locks
- C# New Features
    - Using Static
    - String Interpolation
    - Await in catch/finally
    - Exception filters
    - Pattern Matching
    - Tuples
    - Generalized async Return Types
- Utilizing Class Libraries & Console Project
    - The .NET Console Application Project Type
    - References and Importing Namespaces
    - Instancing Classes
- IOC Container
    - Dependency Injection – Only Concept and Simple Demo

# .Net Core Basics

**Table of contents**

**Program Duration**: 3 Days.

**Understand .NET Core**

- Understand .NET Core architecture and Advantages
- Understand .NET Core middleware & Configuration
- Understand and Use Dependency Injection in .NET Core
- Handle Errors in .NET Core

# Git

**Table of contents**

**Program Duration**: 1 Day.

- Getting Started with Git
    - Install the Git Tools
    - Clone an Existing Repository
    - Add Files to a Repository
    - Edit Files in a Git Repository
    - Create and Merge Branches
    - Rewrite History in a Git Repository
    - Resolve Merge Conflicts

# RDBMS –SQL Server

**Program Duration**: 2.5 Days.

**Table of contents**

- Introduction to RDBMS
    - Introduction to databases
    - Data Models in Database
    - Properties of RDBMS
    - Normalization
    - CODD's Relational Database Rules
    - Data Integrity
    - T-SQL Language
- Working with Data Types, Tables & Data Integrity covering DDL, DML, DCL statements
    - Working with Data Types (Only Basics of Data Types)
    - Working with Schema
    - Working with Tables
    - Implementing Data Integrity
- Beginning with Transact-SQL
    - Transact-SQL
    - System Functions
    - Advanced T-SQL Queries`

    - Advanced T-SQL Statements
    - Other T-SQL Statements
    - Set Operators
    - Transact-SQL
    - System Functions
    - Advanced T-SQL Queries
    - Advanced T-SQL Statements
    - Other T-SQL Statements
- Working with Joins and Subqueries
    - What are Joins?
    - Types of joins
    - Subqueries
- Database Objects: Indexes and Views
    - Introduction to Index in SQL Server
    - Introduction to Views in SQL Server
- Stored Procedures
    - Stored Procedure
    - Implementing Stored Procedure
    - Exception handling using TRY-CATCH
    - Trigger & Cursors Basics

# LINQ and Entity Framework Core

**Program Duration**: 2 Days.

**Table of contents**

- Language Integrated Query
    - Introduction, LINQ Syntax
    - Query Operators
    - Select, from, Where
    - ofType
    - OrderBy
    - ThenBy
    - GroupBy, into
    - Select
    - SelectMany
    - Take, TakeWhile
    - First
    - FirstOrDefault
    - Single
    - SingleOrDefault
    - Aggregate functions Sum, Min, Max, Average, Count
    - Distinct
    - Intersect
    - Except
    - Join

- LINQ projection
- Deferred execution vs immediate execution
- Let keyword
- LINQ to Object
- LINQ to DataTable

- Entity Framework Core
  - Overview of ORM Products
  - Entity Framework introduction
  - Using Database first Approach
  - Using Model First approach
  - Using Code First approach
  - Using LINQ to Entities to perform CRUD operations
  - SQL Query Logging
  - Migration & Database Update
  - Eager Loading Vs Explicit Loading Vs Lazy Loading
  - Raw SQL And Stored Procedures

# ASP.NET Core Web API

**Program Duration**: 3 Days.

**Table of contents**

- Understand REST Service using .NET Core
- Understand Entity Framework Core
- Introduction to .Net Core WebAPI
  - Introduction to Web Service with Demo
  - Introduction to WCF Service with Demo
  - Introduction to Web API
  - Difference between Web Service, WCF Service and Web API
  - Web API features
  - HTTP Web Services
  - Web API Introduction
  - Middleware
  - Web API Routing
  - Configuring WebApi
  - Web API Parameter Biding
  - Action Return Type
  - WebApi Filters
  - Content Negotiation
  - Create CRUD WebApi
  - Consume WebApi
  - MSTest / XUnit
    - Basic Unit Test
  - Create CRUD WebApi
    - Working with swagger
    - Postman Utility
- Understand and Implement Repository, Unit of Work
- Publish WebApi on IIS

# Cloud Computing & Azure Fundamentals

**Program Duration**: 6 Days.

**Table of contents**

- Cloud Computing & Microsoft Azure Fundamentals
- Microsoft Azure Portal Overview (Utilization, Cost)
- Overview of Cloud Computing (Benefit, Capex/Opex)
- Overview of Public and Private Cloud & Hybrid
- Core Azure Service
- Availability Zones, Availability Set, Resource Groups, Azure Resource Manager
- Security & Governance Overview
- Identity, Azure Active Directory, Users & Groups
- Subscriptions and Accounts, Azure Policy, Role-based Access Control (RBAC)
- Azure Portal and Cloud Shell & Azure PowerShell and CLI

# Cloud Networking & Storage

**Program Duration**: 5 Days.

**Table of contents**

- Azure Virtual Machine
- Overview Virtual Networking
- Azure Load Balancer
- VPN Gateway, Azure Application Gateway, Azure CDN, Azure Traffic Manager
- Azure Storage Services
- Structured Data, Semi Structure Data, Unstructured Data
- Azure Database Services, Azure SQL Server

# Azure PaaS Services

**Program Duration**: 6 Days.

**Table of contents**

- Azure App Service
- Overview of Azure Web App
- Overview of Azure Function App
- Overview of Disconnected Architecture
- Overview of Azure Service Bus
- Overview of Azure Logic App
- Overview of Azure Data Factory
- Overview of Azure Analysis Service
- Azure Function App + Azure Service Bus + Azure Logic App