Index

Contents

Immersive Course Structure .NET

.NET LOT provides exposure to the entire spectrum of .NET technologies. It focuses on Desktop as well as Web application development using .NET Technologies. The following table lists the proposed 40 Days Immersive course structure for .NET.

Sr. No.	Course	Duration	Remarks
1	Power Skills	5	
2	Programming Foundation	3	
3	.NET Framework 4.7 + C# 8.0	7	
4	Unit Testing and TDD	1	
5	Git	1	
6	RDBMS and SQL Server	4	
7	.NET Framework 4.7 + C# 8.0 +Unit Testing and TDD+ Git +RDBMS and SQL Server Assessment	0.5	Module Assessment
8	LINQ and Entity Framework	2.5	Introduction to ADO.Net
10	ASP.NET Web Forms	2	
11	ASP.NET MVC	4	
13	ASP.NET Web API	3	
15	Sprint Implementation	4	Project use case to be implemented using ASP.NET MVC, Web API and LINQ & Entity Framework
16	Sprint Evaluation	1	
17	L1 Preparation	1	
18	L1 Test	1	
Total Training Duration		40	

.NET Curriculum

Programming Foundation

Program Duration: 3 Days.

Table of contents

- Introduction to program development with pseudocode
- Good Programming Practices
- Algorithm Analysis and Design
- Data Structure and Algorithm
- Exception Handling

.NET Framework 4.7 and C# 8.0

Program Duration: 7 Days.

Table of contents

- Introduction to .NET Framework 4.7
    - What is .NET Platform?
    - What is .NET Framework
    - Net Framework Architecture
    - .NET Framework, Languages, and Tools
    - .NET Framework Major Components
    - Common Language Runtime (CLR)
    - Compilation and Execution in .NET
    - Understand the .NET Framework 4.7 stack
    - Introduction to .NET Core
        - .NET Core Overview
        - .NET Core Versions
        - .NET Core Compositions
        - Dependency Injection
- Introduction to C#
    - Features of C#
    - C# Compilation and Execution
    - General Structure of a C# Program
    - 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
- Using Microsoft Visual Studio Community
  - Overview of Visual Studio
  - Tracing, Debugging, Build
  - Using break points
  - Using break conditions
  - Using watch and output window
  - Creating multiple projects within one solution
- 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
    - Exception Logging- File or DB
- 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
    - Creating Generic class, Generic Methods, Interfaces, Delegates
    - Collection Initializers
    - Iterators
    - Constraints
- Delegates and Events
    - Introduction to Delegates
    - Implementing Delegates
    - Single & Multicasting Delegate
    - Creating callback method using Delegate
    - Events in C#
    - Generic delegates and generic Delegates, Generic Classes
    - Action, Func and Predicate Delegate
- 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
- Assemblies, Reflection & Attribute based programming
    - Introduction to .NET Reflection
    - Obtaining details about types from the assembly(Private, Public, Shared and Satellite)
    - Obtaining details about methods, properties and fields
- 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
- Good Programming Practices and Design Patterns
- New Features in C# 8.0
    - Readonly members
    - Default interface methods
    - Pattern matching enhancements:
    - Using declarations
    - Static local functions
    - Disposable ref structs
    - Nullable reference types
    - Asynchronous streams
    - Asynchronous disposable
    - Indices and ranges
    - Null-coalescing assignment
    - Unmanaged constructed types
    - Stackalloc in nested expressions
    - Enhancement of interpolated verbatim strings

Unit Testing and TDD

Program Duration: 1 Day.

Table of contents

- Microsoft Unit Testing Framework
    - Test Class & Test Methods
    - 3 A's of Testing – Arrange, Act and Assert
    - Working with Test Explorer
    - Selectively executing test cases
    - Naming Conventions for Test Class and Methods
    - Code Coverage
- Introduction to TDD
    - Advantages of TDD
    - Mocking using Custom Classes
    - Mocking with RhynoMock
    - Various Mock Objects such as - Dummy, Fake, Spy, Stub, Mock
    - Writing Test cases using TDD approach

Git

Program Duration: 1 Day.

Table of contents

- 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: 4 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
- Introduction to SQL Server
  - What is SQL Server?
  - SQL Server Components
  - SQL Server Authentication Modes
  - SQL Server Services
  - SQL Server Version History
  - SQL Server Editions
- 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
- 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 & Functions
    - Stored Procedure
    - Implementing Stored Procedure
    - Exception handling using TRY-CATCH
    - Creating User Defined Functions
- Implementing Triggers
    - Introduction to Triggers
- Transactions
    - Introduction to Transactions
- SQL Server Profiler
    - Introduction to SQL Server Profiler
- NoSQL Database
    - Brief History of NoSQL Databases
    - NoSQL Database Features
    - Types of NoSQL Database
    - Difference between RDBMS and NoSQL
    - Why NoSQL?
    - When should NoSQL be Used?
    - Demo
- Azure SQL Database:
    - Introduction to Azure SQL Database
    - Demo

LINQ and Entity Framework

Program Duration: 2.5 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
  - 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

ASP.Net Web Forms

Program Duration: 2 Days.

Table of contents

- Introduction to ASP.NET
  - Basic ASP.NET concepts, ASP.NET Framework
  - Web Forms and their features
  - Life Cycle of a Web Form
  - Advantages of ASP.NET over ASP
  - Concept of Code Behind, Web Configuration File
  - Page Life Cycle
  - ASP.NET 4.5 features
- Validation in ASP.NET
  - Validation Controls
  - Custom Validator
  - Validation Summary
  - Validation Group
  - Validation: Best Practices and Guidelines
- Master Pages
  - Why Master Pages
  - Working with Master Pages
  - Coding a Master Page and Content Page
  - Nesting of Master Pages
  - Best Practices & Guidelines
- Data Driven Pages
  - Using DataSource controls provided in ASP.NET
  - Using GridView, FormView, Repeater and DetailsView Controls
- State Management
  - HTTP Basics
  - State Management
  - Client-side State Management - ViewState, Hidden Fields, QueryString, Cookies
  - Server-side State Management - Session Object, Application Object
  - Global.asax
  - State Management: Best Practices and Guidelines
- Authentication: Overview
  - Token Based Authentication

ASP.NET MVC

Program Duration: 4 Days.

Table of contents

- Fundamentals of ASP.NET MVC 5
    - ASP.NET MVC Framework
    - ASP.NET vs. ASP.NET MVC
    - ASP.NET MVC Model
- Exploring Controllers
    - Working with Controllers
    - Routing
    - Attribute Routing
    - Action Methods
    - Action Filters, Types of Action Filters
    - Passing Data from controller to view
- Working with Views
    - ASP.NET MVC Razor View Engines
    - HTML Helpers
    - Working with Layout
- Exploring Models & working with data
    - Model Binding
    - Data Annotation
- Entity Framework 6 Features with Asp. Net Scaffolding
    - Code First Model
    - Multiple Migrations
    - Scaffolding
- Web Optimization with ASP.NET MVC
    - Web Optimization
    - Bundling & Minification
    - Configuring bundles
- Using AJAX in ASP.NET MVC
    - Introduction to Ajax
    - AJAX in ASP.NET MVC
    - AJAX Helpers
    - Partial Page rendering
    - AJAX using jQuery
- Using ASP.NET Identity in ASP.NET MVC
    - What is ASP.NET Identity?
    - Features of ASP.NET Identity
    - Claim Based Authentication
    - External Authentication

ASP.NET Web API

Program Duration: 3 Days.

Table of contents

- Introduction to ASP.NET Web API
    - Web API features
    - SOAP- based WebServices
    - HTTP Web Services
    - Web API Introduction
    - Web API Routing
    - Web API Parameter Biding
    - Content Negotiation
    - Working with swagger and Postman Utility