**Training Mode**

* **Duration**: ~2.5 hours per day
* **Hands-on** included in each relevant module
* **Targeted for**: CRM associates with basic OOP knowledge

**Day 1 – .NET Core Introduction & IDE Basics**

* Overview of .NET Core/.NET Framework
* .NET architecture, runtime, MSIL, assemblies, CLS/CTS
* Introduction to Visual Studio 2019/2022
* Project templates, Solution Explorer, folder structure
* Hands-on: Creating your first C# project

**Day 2 – Programming Fundamentals in C#**

* Data types, variables, literals, naming conventions
* Operators (math, logical, assignment, conditional)
* Control flow: if, switch, loops (for, while, do)
* Type conversions: casting, boxing/unboxing, Convert, Parse
* Hands-on: Simple calculator and decision-making exercises

**Day 3 – Functions, Arrays & Intro to OOP**

* Functions: definition, scope, parameters, return types
* Arrays: single and multi-dimensional, System.Array
* OOP Basics: Classes, objects, constructors, encapsulation
* Hands-on: Define classes for a Library or Shopping Cart

**Day 4 – Advanced OOP Concepts**

* Inheritance: types, base and derived classes
* Polymorphism: overloading, overriding, virtual/abstract methods
* Interfaces vs Abstract classes
* Design Patterns Overview: SOLID, DI basics
* Hands-on: Shape or Vehicle inheritance model

**Day 5 – Exception Handling & Reusability**

* Error types, try-catch-finally, custom exceptions
* Object class methods: Equals, ToString, GetHashCode
* Generics: class, method, constraints
* Dependency Injection & Inversion of Control
* Hands-on: Implement custom exceptions & logging

**Day 6 – Delegates, Lambda & Collections**

* Delegates: declaration, invocation, multicast
* Built-in delegates: Action<>, Func<>, Predicate<>
* Anonymous methods and lambda expressions
* Generic Collections: List<T>, Dictionary<TKey, TValue>
* Hands-on: Employee filtering using LINQ & delegates

**Day 7 – LINQ Fundamentals**

* LINQ syntax: Query vs Method syntax
* Filtering, selecting, ordering, grouping
* Aggregate operations (Sum, Count, Average)
* Deferred execution and var keyword
* Hands-on: Data manipulation with LINQ

**Day 8 – File Handling & Serialization**

* File I/O: File, FileInfo, StreamReader, StreamWriter
* Async file access
* Serialization: JSON, XML, Binary using System.Text.Json, Newtonsoft.Json
* Hands-on: Save and load user profiles from files

**Day 9 – Multithreading & ADO.NET**

* Threads: Thread class, Join, Lock, Monitor
* Tasks and Parallel Library
* ADO.NET basics: SqlConnection, SqlCommand, Parameters
* Hands-on: Multi-threaded file operations or logging

**Day 10 – ASP.NET Core Architecture & Middleware**

* Setting up ASP.NET Core projects (web or API)
* Hosting models: Kestrel, IIS
* Middleware pipeline and request flow
* Registering and using services (DI)
* Hands-on: Build a minimal ASP.NET Core web app

**Day 11 – Configuration & Razor/MVC/Web API Basics**

* Configuration sources: appsettings.json, environment variables
* Options pattern
* Razor Pages vs MVC vs Web API
* Logging: ILogger, log levels
* Hands-on: Basic Razor Page form and logging setup

**Day 12 – Routing, Views & Tag Helpers**

* MVC routing: attribute vs conventional
* ViewImports, Layouts, Bootstrap integration
* TagHelpers: form, image, environment
* Hands-on: Build a product catalog page

**Day 13 – Error Handling & State Management**

* Exception filters, developer error pages, status codes
* Session, cookies, TempData, ViewBag/ViewData
* Query strings, hidden fields
* Hands-on: Build a login form with session management

**Day 14 – Models, EF Core & Data Validation**

* Introduction to Entity Framework Core
* DbContext, models, code-first approach
* Data annotations and validation
* Hands-on: Build a Contact form with validation and database save

**Day 15 – Web API, Testing & Deployment**

* REST principles, building a CRUD API
* Attribute routing, response codes, async/await
* Exception handling in APIs, Postman testing
* Authentication, Authorization, JWT basics
* Git basics and deployment overview (Azure, IIS)
* Hands-on: Build and test a complete REST API with Postman