

.NET Core 10.0 Combo Package Includes

1. CSharp 14.0
2. ASP.NET Core 10.0
3. SQL Server 2022
4. ASP.NET Core Web API

CSharp 14.0 Programming

Start learning how to program using the C# programming language and become a professional software developer from a beginner/novice level.

About the Course & Importance of C#

C# is a popular and powerful object-oriented programming language developed by Microsoft. It is widely used to build desktop applications, web applications, Web APIs, cloud solutions, mobile apps, games, enterprise software, and microservices. Whether you are a beginner or an experienced developer, learning C# is a valuable step in your software development journey.

This course is designed for students, fresh graduates, working professionals, and developers who want to build a strong foundation in programming using the latest C# 14.0 features. It starts with the basics and gradually moves into advanced topics such as Object-Oriented Programming (OOP), Exception Handling, Collections, LINQ, Delegates, Events, Generics, Asynchronous Programming, File Handling, Reflection, and modern C# language features.

With its clean syntax, excellent tooling in Visual Studio, and seamless integration with the .NET ecosystem, C# makes software development simple and effective. By the end of this course, you will have the confidence and practical knowledge needed to write professional C# applications and prepare for technologies like ASP.NET Core, Web API, Microservices, Azure, MAUI, and Blazor.

Computer programming is really fun in general, and programming with C# Language using Visual Studio is even better & simpler!

Learning C# 14.0 :

There are 4 phases to learn C# 14.0.

1. Phase I – Introduction to .NET Core & Language Fundamentals
2. Phase II – Application Development and C# 14.0 new features
3. Phase III (Files , Database's, XML & Cloud Storage)
4. Phase IV (Distributed Programming)

Pre-Requisites:	
NA.	Duration : 45 days
Course Materials & Textbooks : 1. Running Notes in class, 2. Pdfs, 3. HandOuts, 4. Certification Material in Pdfs will be provided for this course.	
Reference Websites: www.dotnetgurukul.com	
Instructor: Praveen Kumar M Learning from Praveen Sir, is always different and you will experience it within few sessions – you attend and work regularly as per given guidance then you will be best in any team that you work in any company – assured.	

ASP.NETCore Course Syllabus by Praveen

1. Phase I – Introduction to .NET Core & Language Fundamentals

-  Available Free on YouTube

Topic	Key Concepts Covered
.NET Core & Languages	• Getting Started with .NET: From .NET Framework to .NET Core • Brief history of Microsoft & evolution of programming (COM/DCOM, C, C++, VB, VC++) • Overview of .NET Framework • CLR, FCL, Assemblies (EXE/DLL) • CLR Execution Model (C# → IL → CLR → Execution) • Features: Interoperability, Security, Scalability • Introduction to .NET Core (Cross-platform, Open Source, Modular) • .NET Core vs .NET Framework
Detailed Exploration of .NET Core Components	• CoreCLR – Runtime Execution & Memory Management • CoreFX – Base Class Libraries • RyuJIT – JIT Compilation & Performance Benefits • .NET Standard – API Compatibility • Unified .NET Ecosystem (.NET 5+) • .NET CLI (dotnet new, build, run, publish) • NuGet Package Management • Garbage Collector (GC) & Generational GC • Key Features of .NET Core
.NET Evolution and Versions	• .NET Framework Versions & Compatibility • Supported C# & Visual Studio Versions • Legacy vs Modern .NET Development • .NET Core Versions (1.0 → .NET 8) • Evolution to Unified .NET (.NET 5+) • C# Versions & Feature Highlights (LINQ, async/await, pattern matching, records) • Visual Studio Evolution & Compatibility
C# Data Types and Memory Management	• Built-in C# Data Types • Global, Stack & Heap Memory • Stack vs Heap Allocation • Common Type System (CTS) • Common Language Specification (CLS)
Value Types, Reference Types & Type Casting	• Value Types vs Reference Types • Memory Behavior & Copy Semantics • Stack & Heap Storage Structures • Implicit & Explicit Type Casting • Parse() & TryParse() • Nullable Casting & as Keyword • Checked & Unchecked Blocks
Working with Arrays – Basics	• Introduction to Arrays • Declaration & Initialization • Default Values & Collection Initializers • Accessing Array Elements • Iteration using for, while & foreach • Array Properties & Methods (Length, Sort, Reverse, IndexOf)
Working with Arrays – Advanced	• Single-Dimensional Arrays • Multidimensional Arrays • Jagged Arrays • Accessing & Iterating 2D and Jagged Arrays
Understanding Methods and Parameters in C#	• Introduction to Methods • Method Signature & Definition • Return Types & Return Values • Arguments vs Parameters • Caller vs Called Methods • Pass by Value vs Pass by Reference • ref, out, and in Keywords

2. Phase II – Application Development and C# 14.0 new features



Premium Phase

Topic	Key Concepts Covered
Introduction to Object-Oriented Programming (OOP)	Class & Object, Properties, Methods, Access Modifiers, Encapsulation, Inheritance, Polymorphism, Abstraction (Overview), Procedural vs OOP, Creating Classes & Objects
OOP Principles – Deep Dive	Encapsulation (BankAccount), Inheritance (Vehicle, Car, Bike), Polymorphism (Shape & Draw), Abstraction (API Example)
Interfaces	Interface Fundamentals, Structure & Rules, Implementation Steps, Reusability, Testability, Dependency Injection (DI), Polymorphism, Separation of Concerns (SoC)
Abstract Classes	Abstract Classes, Abstract vs Non-Abstract Members, Rules, Reusability, Encapsulation, Polymorphism, Banking Example
Constructor Basics	Constructors, new Keyword, Constructor Execution Order, Inheritance Flow, Constructor Overloading, Constructor Chaining (this), Access Modifiers
Constructor Types	Default, Parameterized, Copy, Static & Private Constructors, Static Constructor Behavior, Constructor Exceptions, Best Practices
Destructors & Resource Management	Destructors, Garbage Collection Lifecycle, GC.Collect(), WaitForPendingFinalizers(), IDisposable, Dispose Pattern, using Statement
C# Class Types	Abstract Classes, Sealed Classes, Partial Classes, Static Classes, Real-world Use Cases
C# Method Types	Instance Methods, Static Methods, Extension Methods, Async Methods, Anonymous Methods
Delegates & Events	Delegates, Multicast Delegates, Events vs Delegates, Event Declaration & Raising, UI and File Download Scenarios
Properties in C#	Properties vs Fields, Read-only, Write-only, Read-Write Properties, Auto-Implemented Properties, Encapsulation Best Practices
Fields in C#	Instance Fields, Static Fields, Readonly Fields, Constants (const), Backing Fields
SOLID Principles – Part 1	OOP Recap, Importance of Design Principles, SRP, OCP, LSP, ISP, DIP, Monolithic vs SOLID Architecture
SOLID Principles – Part 2 (DIP & DI)	Dependency Inversion Principle, Dependency Injection, Constructor/Method/Property Injection, Built-in DI Container, IoC Containers (Autofac, Ninject), Docker & Kubernetes Overview
Design Patterns	Introduction to GoF Patterns, Singleton, Factory, Repository, Unit of Work, Microservices Architecture, Ocelot API Gateway
Exception Handling	Exceptions vs Errors, try-catch-finally, throw Keyword, using Statement, Custom Exceptions, Database Exception Handling
Async Programming & Multithreading	Concurrency Concepts, Threads, ThreadPool, Parallel Programming, Task-based Async Programming, Choosing Async Models
Reflection	Reflection Basics, Metadata Inspection, Type Class, Runtime Type Information, Dynamic Object Creation, Reflection Use Cases

3. Phase III (Files , Database' s, XML & Cloud Storage)

Premium Phase

Topic	Key Concepts Covered
Introduction to Generics	Recap of OOP Building Blocks (Interfaces, Classes, Methods, Properties, Fields), Why Generics?, Type Safety, Code Reusability, Performance (Avoid Boxing/Unboxing), Generic Classes (MyList<T>), Generic Interfaces (IRepository<T>)
Introduction to Collections & Types	What is a Collection?, Collections vs Arrays, Types of Collections (Index-based, Key-based, Sequential, Specialized), Generic vs Non-Generic Collections, Collection Characteristics, Access Patterns, Duplicates, Thread Safety, Common Interfaces (IEnumerable, ICollection, IDictionary, IList)
Deep Dive: IEnumerable, ICollection & IQueryable	Collection Interface Recap, Exploring Interfaces using Visual Studio Object Explorer, IEnumerable<T> & ICollection<T>, yield Keyword, IQueryable<T> vs IEnumerable<T>, EF Core & LINQ Query Execution, Deferred Execution
Iteration Patterns in C#	Iteration Fundamentals, for vs foreach, IEnumerable Basics, IEnumerator Mechanics, GetEnumerator(), MoveNext(), Current, Stateless vs Stateful Iteration
IComparable vs IComparer	Importance of Sorting, Sorting Primitive Types & Custom Objects, IComparable<T>, IComparer<T>, Multiple Sorting Strategies
Key-Based Collections	Hashtable, Dictionary, SortedDictionary, SortedList, Internal Storage (Hash Tables, Red-Black Trees, Sorted Arrays), Key-based Access, Ordering, Duplicates, Exceptions, GetHashCode()
Sequential / Priority-Based Collections	Stack (LIFO), Push, Pop, Peek, Queue (FIFO), Enqueue, Dequeue, Peek, Real-world Use Cases (Undo, Browser History, Order Processing)
Specialized Collections	ObservableCollection<T>, CollectionChanged Event, UI Data Binding, HashSet<T>, Duplicate Prevention, Set Operations, HybridDictionary, String Collections

4. Phase IV (Distributed Programming)

Topic	Key Concepts Covered
Introduction to ADO.NET & Database Connectivity Basics	Importance of Data-Driven Applications, Why ADO.NET Still Matters, Evolution of Microsoft Data Access (DAO, RDO, ADO → ADO.NET), ADO.NET Architecture (Connected vs Disconnected), .NET Data Providers (SqlClient, OleDb, Odbc), Connection Lifecycle & Best Practices, ADO.NET vs EF Core
SQL Connection Pooling & Database Connection Best Practices	Connection Pooling Architecture, Benefits of Pooling, Connection String Settings (Pooling, Min/Max Pool Size, Timeout), Connection Lifecycle, Performance Optimization

Executing SQL Commands Using ADO.NET	SqlCommand Overview, ExecuteNonQuery(), ExecuteReader(), ExecuteScalar(), ExecuteXmlReader(), Async Methods (ExecuteReaderAsync(), ExecuteNonQueryAsync()), CRUD Operations
Connected Data Access with SqlDataReader	SqlDataReader Fundamentals, Forward-Only Read-Only Result Sets, Read(), HasRows, GetValue(), Access by Column Index & Name, Mapping Data to Objects, Async Data Reader
Disconnected Data Access – DataSet & DataTable	Connected vs Disconnected Architecture, DataSet, DataTable, DataAdapter, DataRow, RowState, In-Memory CRUD Operations, Persisting Changes, AcceptChanges(), Error Handling
Disconnected Data Access – Multiple Tables & Relationships	Working with Multiple Tables, DataRelation, Parent-Child Relationships, Foreign Key & Unique Constraints, Master-Detail Data Binding, DataView for Sorting & Filtering
Advanced DataSet Handling & Serialization	DataSet Serialization, XML, JSON & CSV Formats, WriteXml(), ReadXml(), Newtonsoft.Json, Data Exchange Best Practices
Parameterized Queries & Stored Procedures	SQL Injection Attacks, Parameterized Queries, Add() vs AddWithValue(), Stored Procedures, Database Security Best Practices
Understanding & Using MARS (Multiple Active Result Sets)	Limitations of SqlDataReader, Multiple Active Result Sets (MARS), Enabling MARS, Real-world Parent-Child Query Scenarios
Transactions in ADO.NET	Importance of Transactions, ACID Properties, SqlTransaction, BeginTransaction(), Commit(), Rollback(), Error Handling
Asynchronous Programming in ADO.NET	Synchronous vs Asynchronous Database Calls, async/await Recap, Async ADO.NET APIs, SqlBulkCopy.WriteToServerAsync(), UI Responsiveness
Bulk Data Loading with SqlBulkCopy	Introduction to SqlBulkCopy, Destination Table Requirements, Supported Data Sources, Column Mapping, High-Performance Bulk Inserts
Stored Procedure Input, OUTPUT & RETURN Values	Stored Procedure Fundamentals, Input Parameters, OUTPUT Parameters, RETURN Values, Executing Stored Procedures from ADO.NET

ASP.NET Core 10.0 Programming

Start learning how to program ASP.NET MVC Core applications using the C# programming language and become a professional from beginner/novice user.

About the Course & Importance of ASP.NET Core

All .NET Beginner(s)/Professional(s) who are keen to develop modern, light weight and cloud based web applications should go for this course. ASP.NET Web form and ASP.NET MVC available because of its age is considered to be very matured for web application development, it is because of the popularity of ASP.NET Core Framework and many exclusive features of ASP.NET Core that today it is dominating over Web Forms and MVC are first choice for large sized enterprise web application development.

Computer programming is really fun in general, and programming with C# Language using Visual

Studio is even better & simpler!

Learning ASP.NET Core :

There are 4 phases to learn ASP.NET Core 10.0.

Phase 1. Introduction to ASP.NET Core Fundamentals

Phase 2 .Developing ASP.NET Core Applications

Phase 3.Middleware and Security for ASP.NET Core Applications

Phase 4.Advanced Features and Deployment

Pre-Requisites:
C# basic Knowledge
Duration : 45 days
Course Materials & Textbooks : 1. Running Notes in class, 2. Pdfs, 3. HandOuts, 4. Certification Material in Pdfs will be provided for this course.
Reference Websites: www.dotnetgurukul.com
Instructor: Praveen Kumar M
Learning from Praveen Sir, is always different and you will experience it within few sessions – you attend and work regularly as per given guidance then you will be best in any team that you work in any company – assured.

Phase I – Introduction to ASP.NET Core - Available Free on YouTube

Topic	Key Concepts / Coverage	Hands-on
ASP.NET Core Introduction – Client–Server Model	Client-side: HTML, CSS, JavaScript; Server-side: IIS Express/IIS/Kestrel; HTTP.sys; launchSettings.json; Dev vs Prod Ports; HTTP Request/Response; GET/POST/PUT/DELETE; Browser Lifecycle; DevTools; Rendering Model; Controller→Model→View; Razor; HTML/JSON/XML Output	HTML, Razor & Request/Response Demo
Client-Side Technologies: HTML & JavaScript Basics	Web evolution; HTML & JavaScript; Client-side limitations; Static UI; Security; Debugging; No Strong Typing; Dynamic UI; Type Inconsistency; Need for Server-side , Transition to ASP.NET Core	JavaScript Demos
Server-Side Technologies Comparison	Classic ASP→ASP.NET→MVC→ASP.NET Core; Web Forms; Code-behind; Lifecycle; DAL/BAL; MVC; SoC; Web API; Cross-platform; Razor Pages; Minimal APIs	Hands-on Lab
ASP.NET Core Fundamentals	Framework; MVC+Razor+APIs; Project Structure; Program.cs; appsettings.json; Controllers; Models; Views; wwwroot; Pipeline; Dependency Injection , Service registration & usage	Project Structure & DI Demo
Routing, Controllers & IActionResult	Routing; MVC vs API; Conventional & Attribute Routing; REST; IActionResult; Ok/NotFound/BadRequest; Json/File/Redirect; ActionResult<T>	Hands-on Lab
Razor Pages Fundamentals	Razor Pages; MVC vs Page Model; Pages Folder; PageModel; Pipeline; DI	Hands-on Demo
Razor Pages Routing & Return Types	Routing; @page; Route Parameters; Query Strings; IActionResult; Page; Redirect; Json; File; Content	Hands-on Lab
ASP.NET Core Execution Pipeline	Middleware; app.Use/app.Run; Custom Middleware; UseRouting; UseEndpoints; MVC & Razor Execution Flow	Pipeline Demo

Phase II – Developing ASP.NET Core Applications Premium Phase

Topic	Key Concepts / Coverage	Hands-on
Designing Models & Relationships in ASP.NET Core MVC	EF Core overview, Navigation Properties, Relationship Types (1:1, 1:M, M:M), Designing Models from SQL Server Tables, Customer–Address, Customer–Orders & Order–Products Relationships, DbContext & DbSet Configuration, Controllers for Relational Data, Displaying Related Data in Views	Build relational models, configure EF Core relationships, create controllers & views, navigation flow demo
ViewModels, DTOs & AutoMapper	Purpose of DTOs & ViewModels, Domain Models vs DTOs vs ViewModels, CustomerRequestDto & CustomerResponseDto, UI-specific ViewModels, Data Shaping, AutoMapper Benefits, Mapping Profiles Configuration	Create DTOs, configure AutoMapper, perform end-to-end object mapping
Model Binding in ASP.NET Core	Model Binding Fundamentals, Binding Sources (Route, Query, Form, Body), [FromForm], [FromQuery], [FromBody], [FromRoute], Request Processing Flow	Implement model binding using multiple binding sources, real-world form & API examples
Filters in ASP.NET Core – Part 1	Filter Pipeline Overview, Authorization Filters, Global/Controller/Action Filters, Role-based Authorization, Resource Filters, Custom Resource Filters	Implement authorization & resource filters with practical demonstrations

Filters in ASP.NET Core – Part 2	Action Filters, Logging & Validation, Exception Filters, Centralized Error Handling, Result Filters, Response Modification	Build custom filters and integrate multiple filters into a complete application
Razor View Engine: Standard vs Strongly Typed Helpers	Razor Syntax Fundamentals, Razor Directives, HTML Helpers Overview, Standard HTML Helpers, Strongly Typed HTML Helpers, IntelliSense & Compile-time Safety	Create Razor views using standard and strongly typed HTML helpers
Digging into HTML Helpers (Forms & Controls)	Form Helpers, Input Controls (TextBoxFor, CheckBoxFor, etc.), Model Binding with Helpers, ViewBag & ViewData, Display Helpers, Editor Helpers	Build data entry forms and implement complete form submission workflow
Passing Data from Controller to View	Data Passing Techniques, Model-based Approach, ViewData vs ViewBag, TempData, Cross-request Data Flow	Implement different controller-to-view communication techniques
Tag Helpers in ASP.NET Razor Forms	Introduction to Tag Helpers, HTML Helpers vs Tag Helpers, Advantages, asp-for, asp-action, asp-controller, Built-in Tag Helpers	Develop forms using Razor Tag Helpers with model binding
Advanced Tag Helpers & Validation	Form Tag Helpers, Input Controls, TextArea, CheckBox, Radio Buttons, DropDown Lists, Data Annotations, Validation Rules, Validation Tag Helpers, Client-side Validation	Build a validated registration form with client-side validation
Navigation & Rendering Tag Helpers	Anchor Tag Helper, Routing, Query Strings, Page Handlers, Partial Tag Helper, Environment Tag Helper, Cache Tag Helper, Image Tag Helper	Create reusable UI components and implement navigation & caching
Layouts, ViewStart & ViewImports	Layout Pages, RenderBody, RenderSection, _ViewStart.cshtml, _ViewImports.cshtml, Namespace Management, Tag Helper Registration	Create consistent application layouts and shared Razor configurations

Phase 3: Middleware and Security for ASP.NET Core Applications Premium

Topic	Key Concepts / Coverage	Hands-on
Securing ASP.NET Core Applications using Microsoft Identity (Individual Accounts)	Authentication vs Authorization, Login Workflow, Authentication Options, Individual Accounts, ASP.NET Core Identity Packages, ApplicationDbContext, Identity Database Creation	Create MVC project with Individual Accounts, implement User Registration & Login
Securing Applications using Microsoft Identity Platform & Windows Authentication	Microsoft Identity Platform, Azure Active Directory (AAD), Tenant ID, Client ID, Client Secret, App Registration, Redirect URI, Callback Paths, Authentication Flow, Windows Authentication	Configure Azure AD authentication and Windows Authentication
Role & Policy Authorization with Middleware Introduction	Role-Based Authorization, Policy-Based Authorization, UserManager, RoleManager, Authorize Attribute, Middleware Fundamentals, Request Pipeline, HttpContext.Items, Authorization Middleware	Implement Admin vs Non-Admin access using Roles, Policies & Middleware
ASP.NET Core Identity – Default Authentication, Roles & Claims Caching	Identity Authentication Flow, SignInManager, UserManager, Claims-Based Authentication, Authentication Cookies, Session Behavior, Claims & Roles Caching, Default vs Manual Role Checks	Configure Identity authentication and inspect cached claims & roles
Securing Web APIs using JWT Authentication	Cookie vs Token Authentication, JWT Fundamentals, JWT Structure (Header, Payload, Signature), Token Generation, JWT Validation Middleware, Securing APIs with Authorize Attribute	Build Login API, generate JWT tokens, secure API endpoints using Postman
Implementing OAuth 2.0 Authentication in ASP.NET Core Web API	OAuth 2.0, OpenID Connect, Authorization Code Flow, Access Token vs ID Token, Google OAuth, Authorization & Token Endpoints, OAuth Scopes, ID Token Decoding,	Integrate Google OAuth 2.0 authentication into Web API

	Google Cloud Console	
ASP.NET Core Middleware Pipeline – HTTPS & HSTS	Middleware Pipeline, Middleware Execution Order, HTTP vs HTTPS, HTTPS Redirection, HSTS Concepts, Strict-Transport-Security Headers	Configure HTTPS, HSTS and understand middleware execution flow
ASP.NET Core Middleware Pipeline – CORS & Anti-Forgery	Same-Origin Policy, CORS Fundamentals, CORS Policies, AllowAnyOrigin, AllowAnyMethod, AllowCredentials, CSRF Attacks, Anti-Forgery Tokens, ValidateAntiForgeryToken	Configure CORS policies and implement Anti-Forgery protection
Custom Middleware in ASP.NET Core	Request Processing Pipeline, Inline Middleware, Class-Based Middleware, Reusable Middleware, Real-World Middleware Scenarios	Build custom inline and reusable middleware components
Introduction to State Management in ASP.NET Core	Stateless HTTP, State Management Fundamentals, Client-Side vs Server-Side State Management	Demonstrate various state management techniques
Client-Side State Management in ASP.NET Core	Cookies, Cookie Expiration & Persistence, Query Strings, Hidden Fields, Client-Side Data Storage	Implement Cookies, Query Strings and Hidden Fields in MVC applications
Server-Side State Management in ASP.NET Core	ViewData, ViewBag, TempData, Session State, Session Lifecycle, Application State, MemoryCache	Implement Session Management, TempData and Server-Side State Handling
Caching in ASP.NET Core	Caching Fundamentals, IMemoryCache, Output Caching, Distributed Caching, Redis Concepts, IDistributedCache	Implement Memory Cache, Output Cache and Distributed Caching demonstrations

Phase 4: Advanced Features and Deployment Premium Phase

Topic	Key Concepts / Coverage	Hands-on
Creating RESTful APIs in ASP.NET Core	Web API Fundamentals, REST vs Non-REST, REST Principles (Resources, HTTP Verbs, Statelessness), HTTP Status Codes, JSON Responses, RESTful API Design, Routing, Controllers, CRUD Operations using [ApiController], Swagger & Postman	Build RESTful CRUD APIs, document APIs with Swagger, test endpoints using Postman
API Versioning & JSON Formatting	API Versioning Fundamentals, URL-based Versioning, Header-based Versioning, Configuring API Versioning, JSON Serialization Options, camelCase, Null Handling, Indented Formatting, Custom Date Formatting	Implement API Versioning and customize JSON serialization settings
Consuming APIs (HttpClient & Fetch API)	API Consumer Architecture, Client-Server Communication, HttpClient in .NET, JSON Parsing, JavaScript Fetch API, Frontend Integration	Consume Web APIs using HttpClient and Fetch API in an end-to-end application
Real-Time Communication with SignalR	Traditional HTTP Limitations, Real-Time Communication, SignalR Architecture, WebSockets, SignalR Hub, Client-Server Messaging, Browser Integration	Build a real-time messaging application using SignalR
Exception Handling & Logging with ILogger and Serilog	Exception Handling Review, Production Logging, ILogger Interface, Log Levels (Information, Warning, Error, Critical), Serilog Integration, Console Logging, File Logging, Database Logging	Configure Serilog and implement centralized application logging
Unit Testing in ASP.NET Core (xUnit & NUnit)	Unit Testing Fundamentals, Automated Testing Benefits, xUnit vs NUnit, Assertions, AAA (Arrange-Act-Assert) Pattern, Test Execution	Write and execute unit tests using xUnit and NUnit
Mocking with Moq Framework	Mocking Fundamentals, Moq Framework, Mock Objects, Mock Interfaces, Fake Responses, Method Verification, Success & Failure Scenario Testing	Create mock dependencies and test services using the Moq framework

SQL Server 2022 Programming

Start learning Microsoft SQL Server from the ground up and become a professional SQL Developer and Database Programmer from a beginner/novice level.

About the Course & Importance of SQL Server

Microsoft SQL Server is one of the world's leading Relational Database Management Systems (RDBMS), used by organizations to store, manage, and analyze important business data. It is widely used in enterprise applications, banking, healthcare, retail, e-commerce, manufacturing, government, and cloud-based solutions. Whether you are a beginner or an experienced professional, learning SQL Server is an essential skill for building reliable and high-performance data-driven applications.

This course is designed for students, fresh graduates, working professionals, software developers, database administrators, testers, and data enthusiasts who want to build a strong foundation in Microsoft SQL Server 2022. It starts with database fundamentals and gradually moves into advanced topics such as Database Design, Normalization, SQL Queries, Joins, Functions, Stored Procedures, Views, Indexes, Transactions, Triggers, CTEs, Window Functions, Error Handling, Performance Tuning, Query Optimization, and SQL Server Security through practical examples.

By the end of this course, you will have the confidence and practical knowledge needed to design relational databases, write optimized SQL queries, develop database objects, improve performance, and work effectively with real-world enterprise systems.

Learning SQL Server 2022 :

There are 4 phases to learn SQL Server 2022.

Phase I - Introduction to Databases

Phase II – Querying Data with Transact-SQL

Phase III (Developing SQL Databases)

Phase IV (Updating Your Skills to SQL Server 2022)

Pre-Requisites:
NA
Duration : 45 days
Course Materials & Textbooks : 1. Running Notes in class, 2. Pdfs, 3. HandOuts, 4. Certification Material in Pdfs will be provided for this course.
Reference Websites: www.dotnetgurukul.com
Instructor: Praveen Kumar M Learning from Praveen Sir, is always different and you will experience it within few sessions – you attend and work regularly as per given guidance then you will be best in any team that you work in any company – assured.

ASP.NETCore Course Syllabus by Praveen

Phase I - Introduction to Databases - Available Free on YouTube

Topic	Key Concepts Covered
Data & Information Fundamentals	What is Data, Forms of Data, Measurement of Data, Structured vs Unstructured Data, What is Information, What are Databases, Why Companies Need Databases, Flat File Database Disadvantages, Evolution of Databases, ACID Properties
Database Types	Relational Databases (RDBMS), Hierarchical Databases, Network Databases, NoSQL Databases (Document & Object-Oriented), Column Store Databases, Cloud Databases
RDBMS Fundamentals & Relationships	RDBMS Concepts, Primary Keys & Foreign Keys, RDBMS Characteristics, SQL (T-SQL) Overview, Relationship Types (One-to-One, One-to-Many, Many-to-One, Many-to-Many, Self-Referencing)
Normalization & Database Models	Database Normalization, Normal Forms, Denormalization, OLTP vs OLAP Database Models
SQL Server 2022 Installation & Architecture	SQL Server Installation (Basic & Custom), Editions & Versions, Default vs Named Instances, Windows & SQL Authentication, SQL Server Components (Database Engine, SQL Agent, SSRS, SSAS, SSIS, DQS), SQL Server Management Studio (SSMS), SQL Server Tools
Datatypes, Constraints & Type Conversions	SQL Server Data Types, Primary Key, Foreign Key, UNIQUE, CHECK, DEFAULT, NOT NULL Constraints, Type Conversion Functions (CAST, CONVERT, TRY_CAST, TRY_CONVERT)

Phase II – Querying Data with Transact-SQL - Premium Phase

Topic	Key Concepts Covered
T-SQL Fundamentals & Control Flow	Data Type Functions (String, Date/Time, Mathematical, System), T-SQL Elements (Predicates, Operators, Expressions), Control Flow Statements (IF...ELSE, WHILE, CASE, GOTO, BEGIN...END, TRY...CATCH, RETURN), Variables (Declaration, Assignment, Variable Types)
Set Theory & SQL Query Processing	Set Theory Concepts (Union, Intersection, Difference, Cartesian Product, Subsets), Applying Set Theory in SQL Server, SELECT Statement Structure, SQL Query Processing, Logical Order of Execution
Writing SELECT Queries & CASE Expressions	SELECT Statements, Coding Standards & Best Practices, Scalar Expressions, DISTINCT, CASE Expressions (Simple, Searched, Nested), Column & Table Aliases, Alias Scope
Joins & Join Types	Querying Multiple Tables, Join Fundamentals, Inner Join, Left Join, Right Join, Full Outer Join, Cross Join, Self Join, Join Processing, CustomerOrderDB Schema & ER Diagrams
Multi-Table Joins & Row Count Analysis	Multi-Table Joins, Join Syntax & Best Practices, Primary Key & Foreign Key Relationships, Row Count Analysis, Inner/Outer Join Scenarios, Working with Large Databases

Advanced Join Concepts	Conditional Joins using CASE, Joins with Subqueries, Derived Tables, Joins vs Subqueries, Choosing the Appropriate Query Technique
Sorting Data	ORDER BY Clause, ASC & DESC Sorting, Sorting by Multiple Columns, Sorting by Expressions, NULL Handling, Aliases in ORDER BY
Filtering & Pagination	WHERE Clause, Comparison Operators, AND/OR Conditions, Logical Evaluation, OFFSET/FETCH Pagination, GROUP BY, HAVING, Filtering with Functions, COLLATE, Filtering using Subqueries
Aggregate Functions	Aggregate Functions (SUM, AVG, COUNT, MIN, MAX), NULL Handling, DISTINCT with Aggregates, COALESCE, Aggregate Function Errors
GROUP BY & HAVING in Depth	GROUP BY Fundamentals, Single & Multiple Column Grouping, NULL Handling, Multiple Aggregate Functions, HAVING vs WHERE, Logical Query Processing, GROUP BY Errors
Subqueries – Fundamentals	Introduction to Subqueries, Logical Execution Flow, Self-Contained & Correlated Subqueries, Scalar, Multi-Row & Multi-Column Subqueries, Using Subqueries in SELECT, FROM & WHERE Clauses, Common Error Scenarios
Subqueries – Advanced & CRUD Operations	Correlated Subqueries, Scalar, Multi-Row & Multi-Column Correlated Subqueries, CRUD Operations using Subqueries, Joins vs Subqueries, Performance Considerations

Phase III – Developing SQL Server Databases - 📖 Premium Phase

Topic	Key Concepts Covered
Working with Tables	Temporary Tables (Local #, Global ##), Table Variables vs Scalar Variables, Common Table Expressions (CTEs), Recursive CTEs, Identity Columns, Unique Constraints, IDENTITY_INSERT, SCOPE_IDENTITY(), @@IDENTITY, IDENT_CURRENT()
Working with Views – Fundamentals & Security	Introduction to Views, Tables vs Views, Internal vs External Users, Security & Data Abstraction, Simple & Complex Views, Updatable vs Non-Updatable Views, View Options (SCHEMABINDING, ENCRYPTION, CHECK OPTION, VIEW_METADATA), View Permissions
Working with Views – Advanced & Performance	Indexed (Materialized) Views, Partitioned Views, Performance Impact, View Optimization, When to Avoid Views, Real-World Use Cases
Stored Procedures – Basics	Introduction to Stored Procedures, Benefits over Ad-hoc Queries, Parameterized Stored Procedures, Input & Output Parameters, Creating, Executing, Modifying & Dropping Stored Procedures
Stored Procedures – Best Practices & Debugging	Coding Best Practices, SET NOCOUNT ON, Avoiding SELECT *, Schema Usage, Cursors vs WHILE Loops, Error Handling with TRY...CATCH, ERROR_MESSAGE(), ERROR_NUMBER(), ERROR_SEVERITY(), ERROR_STATE(), ERROR_LINE(), Debugging using PRINT, RAISERROR, SSMS Debugger, SQL Profiler, Transaction Management
Transactions & Transaction Management	Transactions, BEGIN TRANSACTION, COMMIT, ROLLBACK, TRY...CATCH, @@ERROR, @@TRANCOUNT, Savepoints (SAVE TRANSACTION), Nested Transactions, Transaction Count Handling
Functions – Scalar & System Functions	Introduction to Functions, Built-in vs User-Defined Functions, Scalar UDFs, System Functions (String, Date, Math, Aggregate), Creating & Executing Scalar Functions

Functions – Table-Valued Functions & Comparisons	Inline TVFs, Multi-Statement TVFs, Inline vs Multi-Statement Functions, Functions vs Stored Procedures, Functions vs Views, Performance Considerations, Best Practices
Triggers – Fundamentals	Introduction to Triggers, DML Triggers (AFTER, INSTEAD OF), DDL Triggers, Trigger Execution Order, INSERTED & DELETED Tables, Triggers vs Stored Procedures
Triggers – Advanced & Best Practices	Auditing & Logging, Cascading Updates & Deletes, Error Handling & Rollbacks, Triggers vs Constraints, Best Practices, Trigger Limitations, ALTER TRIGGER, DROP TRIGGER
Indexes – Fundamentals	Introduction to Indexes, Clustered Indexes, Non-Clustered Indexes, Clustered vs Non-Clustered, Benefits & Drawbacks, Heap vs B-Tree Storage
Indexes – Performance & Maintenance	Primary Key vs Clustered Index, Unique Key vs Non-Clustered Index, Index Impact on SELECT, INSERT, UPDATE & DELETE Operations, Index Fragmentation, Fill Factor, Pad Index
Indexes – Advanced Types	Covering Indexes, Filtered Indexes, Full-Text Indexes, Advanced Indexing Strategies

Phase IV (Updating Your Skills to SQL Server 2022) - Premium Phase

Topic	Key Concepts Covered
T-SQL Enhancements in SQL Server 2022	Enhanced STRING_SPLIT with Ordinal Positions, LEAST() & GREATEST() Functions, Date & Time Enhancements (DATE_BUCKET, DATETRUNC, GENERATE_SERIES), Approximate Query Processing (APPROX_COUNT_DISTINCT), JSON Enhancements (JSON_PATH_EXISTS, Improved ISJSON, JSON_QUERY, JSON_MODIFY)
Enhancing Data Security and Flexibility in SQL Server 2022	Dynamic SQL, EXEC vs sp_executesql, Preventing SQL Injection, Performance Considerations, Static vs Dynamic SQL, Dynamic Data Masking (DDM), Granular Masking Rules, Role-Based Masking, SQL Server Auditing, Server & Database Audits, Audit Enhancements, Compliance (GDPR, HIPAA)
Intelligent Query Processing (IQP) for Performance in SQL Server 2022	Batch Mode on Rowstore, Approximate Query Processing, Scalar UDF Inlining, Table Variable Deferred Compilation, Adaptive Joins
Backup and Recovery Enhancements in SQL Server 2022	Backup & Restore Architecture, Snapshot Backups, Accelerated Database Recovery (ADR), Backup Compression, Backup Encryption, Restore Performance Improvements, System Database Backup Strategies
CLR Integration & Azure SQL Database in SQL Server 2022	SQL CLR Integration, CLR Stored Procedures & Functions, CLR Security, SQL Server vs Azure SQL Database, Deployment Models (Single Database, Elastic Pool, Managed Instance), Hybrid Connectivity, Cloud Migration Basics

ASP.NET Core Web API 10.0 Programming

Start learning how to build modern RESTful Web APIs using ASP.NET Core and become a professional Web API Developer from a beginner/novice level.

About the Course & Importance of ASP.NET Core Web API

ASP.NET Core Web API is Microsoft's modern framework for building high-performance, secure, scalable, and cross-platform RESTful services. It is widely used by organizations to develop enterprise applications, cloud-native solutions, mobile backends, microservices, e-commerce platforms, banking systems, healthcare applications, and Software-as-a-Service (SaaS) products. Whether you are a beginner or an experienced software professional, learning ASP.NET Core Web API is an essential skill for developing modern distributed applications that communicate efficiently across different platforms and devices.

This course is designed for students, fresh graduates, working professionals, software developers, and architects who want to build a strong foundation in ASP.NET Core Web API 10.0. It starts with Web API fundamentals and gradually progresses to advanced concepts such as REST Architecture, HTTP Methods, Routing, Model Binding, Model Validation, Dependency Injection, Entity Framework Core, Repository Pattern, CRUD Operations, Authentication & Authorization using JWT, Logging, Exception Handling, API Versioning, Caching, Rate Limiting, File Upload & Download, Background Services, Minimal APIs, GraphQL, OData, gRPC, Swagger/OpenAPI Documentation, API Security, Performance Optimization, and Deployment through practical real-world examples.

By the end of this course, you will have the confidence and practical knowledge needed to design, develop, secure, test, document, and deploy enterprise-grade Web APIs. You will also gain the skills required to build scalable backend services that integrate seamlessly with web applications, mobile applications, cloud platforms, and microservices-based architectures.

Learning ASP.NET Core Web API:

There are 4 phases to learn ASP.NET Core Web API.

Phase I –Introduction to APIs – Foundation of REST & Web API

Phase II –Building Real-World APIs (CRUD + EF + Services)

Phase III –Advanced API Security & Infrastructure

Phase IV –Advanced API Security & Infrastructure

Pre-Requisites:

NA

Duration : 45 days

Course Materials & Textbooks : 1. Running Notes in class, 2. Pdfs, 3. HandOuts, 4. Certification Material in Pdfs will be provided for this course.

Reference Websites:www.dotnetgurukul.com

Instructor: Praveen Kumar M

Learning from Praveen Sir, is always different and you will experience it within few sessions – you attend and work regularly as per given guidance then you will be best in any team that you work in any company – assured.

ASP.NETCore Web API Course Syllabus by Praveen

Phase I - Introduction to APIs – Foundation of REST & Web API -  Available Free on YouTube

Topic	Key Concepts Covered
Introduction to APIs – Foundation of REST & Web API	Client–Server Architecture, Types of APIs (Public, Private, Partner), API Communication Styles (REST, SOAP, GraphQL, gRPC, OData), Requests & Responses (Path, Query, Body), JSON, HTTP Status Codes
REST Architecture, API Evolution & Core Constraints	What is REST, Why REST, REST Constraints (Client–Server, Resources, Representations, Statelessness, Uniform Interface, Cacheability, Layered System)
ASP.NET Core Web API – Project Structure & Pipeline	Web API Project Structure, Enterprise Folder Organization, Middleware Pipeline, Controllers, Controller vs ControllerBase, Benefits of [ApiController]
Web API Controllers & Routing Behavior	[ApiController] Features, Automatic Model Validation, Attribute Routing, Action Selection, Routing Conflicts
Routing Types & Advanced Routing	Attribute Routing vs Conventional Routing, Route Parameters, Optional & Default Values, Route Constraints (int, alpha, length, regex, guid, decimal)
HTTP Methods & Web API Operations	HTTP Methods (GET, POST, PUT, DELETE, PATCH, HEAD, OPTIONS), Safe vs Unsafe Methods, Idempotent vs Non-Idempotent Operations
Model Binding, DTOs & Validation	Model Binding Sources ([FromRoute], [FromBody], [FromQuery], [FromHeader]), DTOs vs Entities, Model Validation Pipeline
HTTP Status Codes & API Responses	HTTP Status Code Categories (2xx, 4xx, 5xx), Common Response Codes (200, 201, 204, 400, 404, 409, 500), Real-World API Response Design

Phase II – Building Real-World APIs (CRUD + EF + Services)-  Premium Phase

Topic	Key Concepts Covered
Dependency Injection in Web API (Services & Lifetimes)	Introduction to Dependency Injection (DI), Benefits of Decoupling & Testability, Service Lifetimes (Singleton, Scoped, Transient), Request-Based Lifetime Behavior
Repository Pattern for Web API	Repository Pattern, Clean Controller Design, IRepository & Repository Implementation, Optional Service Layer, Scoped DbContext, Dependency Injection Request Lifecycle
Applying DI Lifetimes in Real-World Applications	Choosing the Appropriate Service Lifetime, Transient Services, Singleton Services, Lifetime Rules, Best Practices
Entity Framework Core Integration & CRUD Operations	Entity Framework Core Overview, DbContext, DbSet, CRUD Operations (POST, PUT, DELETE), Clean Architecture Flow, Basic Exception Handling

Paging, Sorting & Filtering in Web APIs	Query Parameter Filtering, LINQ-Based Filtering, Sorting (Ascending & Descending), Pagination using Skip() & Take(), Edge Cases
AutoMapper for Web APIs (Entity $\rightleftharpoons$ DTO)	Importance of DTOs, Request & Response DTOs, AutoMapper Concepts, Manual vs Automatic Mapping, Performance Considerations
N-Tier Architecture for Web APIs	Layered Architecture, API Layer, Application Layer, Domain Layer, Infrastructure Layer, Transition from Repository Pattern to Enterprise Architecture
Asynchronous APIs & Performance Optimization	Importance of Asynchronous Programming, Thread Blocking, async/await, Task-Based Asynchronous Pattern (TAP), Scalability & Performance
File Upload & Download in Web APIs	IWebHostEnvironment, IFormFile, multipart/form-data, File Validation, File Upload & Download Strategies

Phase III –Advanced API Security & Infrastructure - Premium Phase

Topic	Key Concepts Covered
Rate Limiting in ASP.NET Core Web API	Introduction to Rate Limiting, Why APIs Need Rate Limits, Real-World Use Cases (Login, OTP, Banking APIs), IP-Based vs Client-Based Limiting, Time Windows & Thresholds, HTTP 429 (Too Many Requests), Best Practices for Secure APIs
JWT Authentication in ASP.NET Core Web APIs	Authentication vs Authorization, JWT Structure (Header, Payload, Signature), Real-World JWT Usage, JWT Configuration in appsettings.json, Issuer, Audience, Secret Key, Token Expiration, Claims
OAuth 2.0 Authentication in ASP.NET Core Web API	OAuth 2.0 Fundamentals, OpenID Connect (OIDC), Authorization Flow, Access Token vs ID Token, Google OAuth Endpoints, Scopes, Token Decoding
Role-Based & Policy-Based Authorization (JWT)	JWT Recap, Claims vs Roles vs Policies, Adding Role Claims to JWT, Creating Authorization Policies, Role-Based vs Policy-Based Authorization
CORS in ASP.NET Core Web APIs (Cross-Origin Resource Sharing)	Same-Origin Policy, Understanding CORS Errors, Configuring CORS Policies, AllowAnyOrigin, AllowAnyMethod, AllowCredentials, Security Considerations
Global Exception Middleware in ASP.NET Core	Exceptions vs Errors, Common API Error Scenarios (400, 409, 500), Limitations of try...catch in Controllers, Middleware-Based Global Exception Handling, RequestDelegate, Request Pipeline
Input Validation using FluentValidation	Why Validation Should Not Be Performed in Controllers, FluentValidation Fundamentals, FluentValidationAutoValidation, Request Pipeline Validation, Rule-Based Validation, Custom Validation Messages

Phase IV –Advanced API Security & Infrastructure - Premium Phase

Topic	Key Concepts Covered
Introduction to Minimal APIs	Motivation & Evolution of Minimal APIs, Controllers vs Minimal APIs, Program.cs-Only API Model, Characteristics of Minimal APIs, Recommended Use Cases, Scenarios Where Minimal APIs Should Be Avoided
Parameter Binding & Request Handling in Minimal APIs	Lambda Handlers (MapGet, MapPost, MapPut, MapDelete), Request Lifecycle, Convention-Based Parameter Binding, Route, Query, Header & Body Parameters
Dependency Injection in Minimal APIs	Dependency Injection Fundamentals, Controllers vs Minimal APIs DI Model, Registering Services (Singleton, Scoped, Transient), Injecting Services into Lambda Handlers, Using Built-In Services (ILogger, IConfiguration)
OData in ASP.NET Core	Introduction to OData, Challenges with Traditional APIs, OData Query Options, Entity Data Model (EDM), OData Request Pipeline, Metadata, OData Convention Model Builder, IQueryable, Filtering, Sorting, Paging, Field Selection
Introduction to GraphQL	GraphQL Fundamentals, Over-Fetching & Under-Fetching Problems, Schema, Types, Queries, Mutations, GraphQL Setup in ASP.NET Core, Product & Order Schema Design, Single Endpoint for Multiple Entities, Error Handling, Performance Basics
gRPC in ASP.NET Core	Introduction to gRPC, Real-World Use Cases, Protocol Buffers (Proto Files), Unary & Streaming Calls, Creating gRPC Services, Message & Service Definitions, Client-Server Communication, Performance Comparison with REST
REST vs Minimal APIs vs OData vs GraphQL vs gRPC	Architectural Differences, Performance Comparison, Payload Size & Network Efficiency, Strengths & Limitations, Real-World Use Cases, API Style Selection, Hybrid API Architecture Design