

	Subject	Date	Time
1	Basics of C# Programming		
1.1	C# language and .NET platform	3.01.24	18:00
1.2	Variables		
1.3	Data types		
1.4	Static variables and constants		
1.5	Working with the console application		
1.6	Arrays		
1.7	C# arithmetic/comparison operations		
1.8	If else and Switch case constructions		
1.9	Loops (For, Foreach, While, Do-while)	4.01.24	18:00
1.10	Methods		
1.11	Method with params parameter		
1.12	Ref and Out keywords in methods		
1.13	Local and recursive function		
1.14	Tuple and Enum		
2	Classes, structures and namespaces		
2.1	Classes and objects	9.01.24	18:00
2.2	Constructors, initializer, and destructors		
2.3	Fields and properties		
2.4	Method and static method in class		
2.5	Structures		
2.6	Record type		
2.7	Namespace and global namespace		
2.8	Partial and extended classes	11.01.24	18:00
2.9	Value types and reference types		
2.10	Nullability in value types and reference types		
2.11	Accessibility of the class and class members		
3	Improve searching and designing knowledges		
3.1	How to search/find what you need?		
3.2	How to use AI chats correctly?		
3.3	Select a project to atomize your organization		

4	Delegates, Events, and Lambdas	16.01.24	18:00
4.1	Delegates and using of that		
4.2	Action and Func Delegates		
4.3	Anonymous Methods		
4.4	Lambdas		
4.5	Events		
5	Object-Oriented Programming (OOP)		
5.1	What is OOP and its concepts in C#?	18.01.24	18:00
5.2	Inheritance		
5.3	Abstract Classes		
5.4	Read-only Properties in a Class		
5.5	Virtual Methods and Properties		
5.6	Hiding, Overriding, and Abstract Methods		
5.7	Interfaces		
5.8	Interface Inheritance		
5.9	Generic Classes		
5.10	Generic Methods		
5.11	Generic Properties		
6	Collections and LINQ		
6.1	List<T>		
6.2	Dictionary<Tkey, Tvalue>		
6.3	ConcurrentDictionary<Tkey, Tvalue>		
6.4	Span<T>		
6.5	Queue<T>		
6.6	Stack<T>		
6.7	HashSet<T>		
6.8	LINQ-запросы		
6.9	IEnumerable<T> и IQueryable<T>		
11	Multithreading		
11.1	Introduction		
11.2	Running Code Simultaneously		
11.3	Processes		

11.4	Threads		
8	Parallelism		
8.1	Introduction		
8.2	Difference Between Concurrency and Parallelism		
8.3	Launching a New Thread		
9	Asynchronous and Synchronous		
9.1	Introduction		
9.2	Difference Between Concurrency, Parallelism, and Asynchrony		
9.3	Async/await methods		
10	Concurrency		
10.1	Introduction		
10.2	Difference Between Concurrency and Multithreading		
10.3	Avoid of concurrency issue		
11	Memory Management in .NET		
11.1	Value and reference types		
11.2	Stack and Heap		
11.3	Mutable and Immutable classes		
11.4	Boxing and Unboxing		
11.5	Garbage collector		
11.6	Small/Large Object Heap		
11.7	Managed and unmanaged code		
11.8	Dispose Pattern		
11.9	Finalizer		
12	SOLID		
12.1	Single Responsibility Principle		
12.2	Open-closed Principle		
12.3	Liskov substitution Principle		
12.4	Interface Segregation Principle		
12.5	Dependency Inversion Principle		

13	Design patterns		
13.1	Introduction		
13.2	Pattern Types		
13.3	Creational patterns		
13.4	Structural patterns		
13.5	Behavioral patterns		
13.6	Repository, Strategy, Dependency Injection pattern		
14	Exam 1		
15	ASP.NET Core		
15.1	Introduction to ASP.NET Core		
15.2	Rules for creating routes		
15.3	Logging in ASP.NET Core		
16	REST and API		
16.1	Introduction to REST		
16.2	Basic principles		
16.3	Http methods and responses		
17	Entity Framework Core (ORM)		
17.1	Entity Framework Core		
17.2	Using Entity Framework Core in ASP.NET Core		
17.3	Modeling and creating tables		
17.4	Creating relationships between tables		
17.5	CRUD with Entity Framework Core		
17.6	Repository pattern for CRUD operation		
17.7	Approaches for obtaining data: Eager, Lazy loading		
17.8	Migration management		
18	Dependency injection (DI)		
18.1	Introduction		
18.2	Dependency Lifecycle		

18.3	Service provider		
18.4	Creating your own services		
19	Middlewares		
19.1	Introduction		
19.2	Creating a simple Middleware		
20	Authorization and authentication		
20.1	Introduction		
20.2	Authorization and authentication methods		
20.3	Authorization and Authentication in REST		
20.4	Adding Authentication to ASP.NET Core (Bearer)		
20.5	Adding Authorization in ASP.NET Core		
20.6	JWT tokens		
20.7	Role of JWT in REST		
20.8	Header, Payload, Signature in JWT		
20.9	Creating a server for generating JWT tokens		
21	Data Validation in ASP.NET Core		
21.1	Introduction		
21.2	Creating services for validation		
21.3	Using the FluentValidation framework		
22	Request and Response		
22.1	Data transfer object (DTO)		
22.2	Using record		
22.3	Using the AutoMapper framework		
23	MediatR and CQRS		
23.1	Introduction		
23.2	ASP.NET Core: Request Handling		
24	Test Driven Development Methodology		
24.1	Creating a Project Using TDD		

24.2	TDD principles		
24.3	Practice: Creating a simple calculator using TDD		
25	Unit testing		
25.1	Using the NUnit framework		
25.2	Using the xUnit framework		
25.3	Mock testing		
25.4	Mocking using Moq		
25.5	Mocking using NSubstitute		
26	Integration tests		
26.1	Working with WebApplicationBuilderFactory		
26.2	Creating an HttpClient from a WebApplicationBuilderFactory		
26.3	EF Core configuration under different test environments		
26.4	Using services from WebApplicationBuilderFactory		
27	Blazor		
28.1	Introduction		
28.2	Blazor Web Assembly and Server		
28.3	Razor pages		
28.4	Create your first Blazor Web Assembly project		
29	Blazor pages and templates		
29.1	Creating pages and specifying the address		
29.2	Creating and applying a template		
29.3	Writing C# code inside a page		
30	Blazor Authentication and Authorization		
30.1	Introduction		
30.2	Creating a CustomAuthstateProvider		
30.3	Saving the token in local storage		
30.4	Authorization in pages		
31	Blazor CRUD		

31.1	Creating Pages and Project Templates		
31.2	Using TabBlazor		
31.3	Execution of CRUD warehouses		
32	DevOps – project publication		
32.1	Introducing Azure		
32.2	Creating resources on Azure		
32.3	Create Azure Key Vault		
32.4	Creating a Resource API		
32.5	Creating SQL Server and Database		
32.6	Creating a Blazor Static Web App		
32.7	Creating a CI-CD for publishing		
33	Preparing for the Demo		