	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	23.01.24	18:00
5.7	Interfaces		
5.8	Interface Inheritance		
5.9	Generic Classes	25.01.24	18:00
5.10	Generic Methods		
5.11	Generic Properties		
6	**Collections and LINQ queries**		
6.1	List<T>	30.01.24	18:00
6.2	Dictionary<Tkey, Tvalue>		
6.3	ConcurrencyDictionary<Tkey, Tvalue>		
6.4	Span<T>		
6.5	Queue<T>		
6.6	Stack<T>		
6.7	HashSet<T>		
6.8	IEnumerable<T> and IQueryable<T>		
6.9	LINQ-queries	1.02.24	18:00
	Exam 1	6.02.24	18:00
	Tests		
	Review projects		

7	**Entity Framework Core (ORM)**	8.02.24	18:00
7.1	Entity Framework Core		
7.2	Using Entity Framework Core in ASP.NET Core		
7.3	Modeling and creating tables		
7.4	Creating relationships between tables	13.02.24	18:00
7.5	CRUD with Entity Framework Core		
7.6	Repository pattern for CRUD operation		
7.7	Approaches for obtaining data: Eager, Lazy loading	15.02.24	18:00
7.8	Migration management		
8	**ASP.NET Core**	20.02.24	18:00
8.1	Introduction to ASP.NET Core		
8.2	Rules for creating routes		
8.3	Logging in ASP.NET Core		
9	**REST and API**	22.02.24	18:00
9.1	Introduction to REST		
9.2	Basic principles		
9.3	Http methods and responses		
10	**Multithreading**	27.02.24	18:00
10.1	Introduction		
10.2	Running Code Simultaneously		
10.3	Processes		
10.4	Threads		
11	**Parallelism**	29.02.24	18:00
11.1	Introduction		
11.2	Difference Between Concurrency and Parallelism		
11.3	Launching a New Thread		
12	**Asynchronous and Synchronous**	05.03.24	18:00
12.1	Introduction		
12.2	Difference Between Concurrency, Parallelism, and Asynchrony		

12.3	Async/await methods		
13	**Concurrency**		
13.1	Introduction		
13.2	Difference Between Concurrency and Multithreading		
13.3	Avoid of concurrency issue		
14	**Memory Management in .NET**	07.03.24	18:00
14.1	Value and reference types		
14.2	Stack and Heap		
14.3	Mutable and Immutable classes		
14.4	Boxing and Unboxing		
14.5	Garbage collector		
14.6	Small/Large Object Heap		
14.7	Managed and unmanaged code	12.03.24	18:00
14.8	Dispose Pattern		
14.9	Finalizer		
15	**SOLID**	12.03.24	18:00
15.1	Single Responsibility Principle		
15.2	Open-closed Principle		
15.3	Liskov substitution Principle		
15.4	Interface Segregation Principle		
15.5	Dependency Inversion Principle		
15.6	Use SOLID in real use cases	14.03.24	18:00
16	**Design/Architectural Patterns**	19.03.24	18:00
16.1	Introduction		
16.2	Architectural patterns		
16.3	Creational design patterns		
16.4	Structural design patterns		
16.5	Behavioral design patterns		
16.6	Repository, Strategy, Dependency Injection pattern		

17	**Dependency injection (DI)**		
17.1	Introduction		
17.2	Dependency Lifecycle		
17.3	Service provider		
17.4	Creating your own services		
18	**Middlewares**		
18.1	Introduction		
18.2	Creating a simple Middleware		
19	**Authorization and authentication**		
19.1	Introduction		
19.2	Authorization and authentication methods		
19.3	Authorization and Authentication in REST		
19.4	Adding Authentication to ASP.NET Core (Bearer)		
19.5	Adding Authorization in ASP.NET Core		
19.6	JWT tokens		
19.7	Role of JWT in REST		
19.8	Header, Payload, Signature in JWT		
19.9	Creating a server for generating JWT tokens		
20	**Data Validation in ASP.NET Core**		
20.1	Introduction		
20.2	Creating services for validation		
20.3	Using the FluentValidation framework		
21	**Request and Response**		
21.1	Data transfer object (DTO)		
21.2	Using record		
21.3	Using the AutoMapper framework		
22	**MediatR and CQRS**		
22.1	Introduction		
22.2	ASP.NET Core: Request Handling		

	Exam 2		
	Tests		
	Review projects		
23	**Test Driven Development Methodology**		
23.1	Creating a Project Using TDD		
23.2	TDD principles		
23.3	Practice: Creating a simple calculator using TDD		
24	**Unit testing**		
24.1	Using the NUnit framework		
24.2	Using the xUnit framework		
24.3	Mock testing		
24.4	Mocking using Moq		
24.5	Mocking using NSubstitute		
25	**Integration tests**		
25.1	Working with WebApplicationBuilderFactory		
25.2	Creating an HttpClient from a WebApplicationBuilderFactory		
25.3	EF Core configuration under different test environments		
25.4	Using services from WebApplicationBuilderFactory		
26	**Blazor**		
26.1	Introduction		
26.2	Blazor Web Assembly and Server		
26.3	Razor pages		
26.4	Create your first Blazor Web Assembly project		
27	**Blazor pages and templates**		
27.1	Creating pages and specifying the address		
27.2	Creating and applying a template		
27.3	Writing C# code inside a page		

28	**Blazor Authentication and Authorization**		
28.1	Introduction		
28.2	Creating a CustomAuthStateProvider		
28.3	Saving the token in local storage		
28.4	Authorization in pages		
29	**Blazor CRUD**		
29.1	Creating Pages and Project Templates		
29.2	Using TabBlazor		
29.3	Execution of CRUD warehouses		
30	DevOps – project publication		
30.1	Introducing Azure		
30.2	Creating resources on Azure		
30.3	Create Azure Key Vault		
30.4	Creating a Resource API		
30.5	Creating SQL Server and Database		
30.6	Creating a Blazor Static Web App		
30.7	Creating a CI-CD for publishing		
	Preparing for the Demo		