| | Subject | Date |
|---|---|---|
| | **Basics of C# Programming** | |
| 1. | Introduced agenda of C# course | 24.02.25 |
| 2. | C# language and .NET platform | |
| 3. | Variables | 27.02.25 |
| 4. | Data types | |
| 5. | Static variables and constants | |
| 6. | Working with the console application | |
| 7. | Arrays | |
| 8. | C# arithmetic/comparison operations | |
| 9. | If else and Switch case constructions | 03.03.25 |
| 10. | Loops (For, Foreach, While, Do-while) | |
| 11. | Methods | |
| 12. | Method with params parameter | |
| 13. | Ref and Out keywords in methods | |
| 14. | Local and recursive function | |
| 15. | Tuple and Enum | 06.03.25 |
| | | |
| | **Classes, Structs, Records and Namespaces** | |
| 1. | Classes and objects | |
| 2. | Constructors, initializers, and destructors | |
| 3. | Fields and properties | |
| 4. | Method and static method in class | 10.03.25 |
| 5. | Structs | |
| 6. | Record type | |
| 7. | Namespace and global namespace | |
| 8. | Partial and extended classes | |
| 9. | Value types and reference types | 13.03.25 |
| 10. | Mutable And Immutable | |
| 11. | Boxing and Unboxing | |
| 12. | Nullability in value types and reference types | |
| 13. | Accessibility of the class and class members | |
| | | |
| | Improve searching and designing knowledges | 17.03.25 |
| 1. | How to search/find what you need? | |

| | | |
|---|---|---|
| 2. | How to use AI chats correctly? | |
| 3. | Select a project to atomize your organization | |
| | | |
| | **Delegates, Events, and Lambdas** | |
| 1. | Delegates and using of that | |
| 2. | Action and Func Delegates | |
| 3. | Anonymous Methods | |
| 4. | Lambdas | |
| 5. | Events | 20.03.25 |
| | | |
| | **Collections and LINQ queries** | |
| 1. | List<T> | |
| 2. | Dictionary<Tkey, Tvalue> | |
| 3. | ConcurrencyDictionary<Tkey, Tvalue> | |
| 4. | Span<T> | |
| 5. | Queue<T> | |
| 6. | Stack<T> | 24.03.25 |
| 7. | HashSet<T> | |
| 8. | IEnumerable<T> and IQueryable<T> | |
| 9. | LINQ-queries | |
| | | |
| | **Multithreading** | 01.04.25 |
| 1. | Introduction | |
| 2. | Running Code Simultaneously | |
| 3. | Processes | |
| 4. | Threads | |
| 5. | Parallelism | 03.04.25 |
| | | |
| | **Asynchronous and Synchronous** | |
| 1. | Introduction | |
| 2. | Difference Between Concurrency, Parallelism, and Asynchrony | |
| 3. | Async/await methods | 07.04.25 |
| | | |
| | **Concurrency** | |

| | | |
|---|---|---|
| 1. | Introduction | |
| 2. | Difference Between Concurrency and Multithreading | |
| 3. | Avoid of concurrency issue | |
| | | |
| | **Exam** 1 | |
| 1. | Review of all passed items | 14.04.25 |
| 2. | Tests | 17.04.25 |
| 3. | Review projects | |
| | | |
| | **Object-Oriented Programming (OOP)** | |
| | What is OOP and its concepts in C#? | |
| | Inheritance | |
| | Abstract Classes | |
| | Read-only Properties in a Class | |
| | Virtual Methods and Properties | |
| | Hiding, Overriding, and Abstract Methods | |
| | Interfaces | |
| | Interface Inheritance | |
| | Generic Classes | |
| | Generic Methods | |
| | Generic Properties | |
| | | |
| | **Entity Framework Core (ORM)** | |
| | Entity Framework Core | |
| | Using Entity Framework Core in ASP.NET Core | |
| | Modeling and creating tables | |
| | Creating relationships between tables | |
| | CRUD with Entity Framework Core | |
| | Repository pattern for CRUD operation | |
| | Approaches for obtaining data: Eager, Lazy loading | |
| | Migration management | |
| | | |
| | **ASP.NET Core** | |
| | Introduction to ASP.NET Core | |
| | Rules for creating routes | |

| | | |
|---|---|---|
| | Logging in ASP.NET Core | |
| | | |
| | **REST and API** | |
| | Introduction to REST | |
| | Basic principles | |
| | Http methods and responses | |
| | | |
| | **Memory Management in .NET** | |
| | Value and reference types | |
| | Stack and Heap | |
| | Mutable and Immutable classes | |
| | Boxing and Unboxing | |
| | Garbage collector | |
| | Small/Large Object Heap | |
| | Managed and unmanaged code | |
| | Dispose Pattern | |
| | Finalizer | |
| | | |
| | **SOLID** | |
| | Single Responsibility Principle | |
| | Open-closed Principle | |
| | Liskov substitution Principle | |
| | Interface Segregation Principle | |
| | Dependency Inversion Principle | |
| | Use SOLID in real use cases | |
| | | |
| | **Architectural/Design Patterns** | |
| | Introduction | |
| | Architectural patterns | |
| | Creational design patterns | |
| | Structural design patterns | |
| | Behavioral design patterns | |
| | Repository, Strategy, Dependency Injection pattern | |
| | | |

| | | |
|---|---|---|
| | **Dependency injection (DI)** | |
| | Introduction | |
| | Dependency Lifecycle | |
| | Service provider | |
| | Creating your own services | |
| | | |
| | **Middlewares** | |
| | Introduction | |
| | Creating a simple Middleware | |
| | | |
| | **Authorization and authentication** | |
| | Introduction | |
| | Authorization and authentication methods | |
| | Authorization and Authentication in REST | |
| | Adding Authentication to ASP.NET Core (Bearer) | |
| | Adding Authorization in ASP.NET Core | |
| | JWT tokens | |
| | Role of JWT in REST | |
| | Header, Payload, Signature in JWT | |
| | Creating a server for generating JWT tokens | |
| | | |
| | **Request and Response** | |
| | Data transfer object (DTO) | |
| | Using record | |
| | Using the AutoMapper framework | |
| | | |
| | **Data Validation in ASP.NET Core** | |
| | Introduction | |
| | Creating services for validation | |
| | Using the FluentValidation framework | |
| | | |
| | **MediatR and CQRS** | |
| | Introduction | |
| | ASP.NET Core: Request Handling | |

| | | |
|---|---|---|
| | | |
| | **Exam** 2 | |
| | Tests | |
| | Review projects | |
| | | |
| | **Test Driven Development Methodology** | |
| | Creating a Project Using TDD | |
| | TDD principles | |
| | Practice: Creating a simple calculator using TDD | |
| | | |
| | **Unit testing** | |
| | Using the NUnit framework | |
| | Using the xUnit framework | |
| | Mock testing | |
| | Mocking using Moq | |
| | Mocking using NSubstitute | |
| | | |
| | **Integration tests** | |
| | Working with WebApplicationBuilderFactory | |
| | Creating an HttpClient from a WebApplicationBuilderFactory | |
| | EF Core configuration under different test environments | |
| | Using services from WebApplicationBuilderFactory | |
| | | |
| | **DevOps – project publication** | |
| | Introducing Azure | |
| | Creating resources on Azure | |
| | Create Azure Key Vault | |
| | Creating a Resource API | |
| | Creating SQL Server and Database | |
| | Creating a Blazor Static Web App | |
| | Creating a CI-CD for publishing | |