Contents

Trainer, Consultant, Software engineer
Ákos Nagy

https://hireme.dotnetfalcon.com
https://dotnetfalcon.com

.NET Memory Management

Length: 5 days

Prerequisites: Strong and confident .NET knowledge, C# programming knowledge and Visual Studio knowledge, basic knowledge of general programming concepts

Software requirements:

- Visual Studio 2019
- WinDbg Preview
- Windows SDK
- PerfView
- CLRProfiler
- VMMap

Description: The course discusses the automated memory management of the .NET framework using both a theoretical and practical approach with demonstrating possibilities to configure and finetune memory management and programming best practices to improve the memory profile of applications.

Topics:

- Overview of memory management in .NET
    - Different memory partitions and their roles
    - Purpose and handling of the new object pointer
    - Garbage collector algorithm
    - Garbage collector algorithm optimization
    - Lifecycle management
    - GC flavors
    - Segments
- Overview of GC API
- Diagnostic tools
    - Performance counters
    - Perfview, CLRProfiler, Visual Studio Profiler
    - WinDbg, SOS, SOSEx
    - Microbenchmarking
    - ClrMD
- Programming best-practices
    - Using value types, boxing
    - Span
    - ValueTask
    - Pooling
    - Event handling
    - Collection
    - Delegates
    - Using strings
    - Lifecycle management

-

Developing web portals using ASP.NET Core

Length: 5 days

Prerequisites: Solid knowledge of C# and the .NET framework, confidence using Visual Studio

Software requirements: Visual Studio 2019 or newer and ASP.NET Core SDK 3.1 or newer, with local SQL Express (or other SQL database server) installed, SQL Server Management Studio recommended

Description: During the course attendants get to know the modern, new-generation ASP.NET Core web development framework and associated design and architectural patterns. so that they can develop well-structured, testable web applications.

Topics:

- Overview of the MVC architecture
  - Purpose of the View and Controller
  - Purpose of the Model
- Creating the Model, View and Controller
  - Data access
  - Connecting components
  - Loose-coupling, reusability, creating unit testable components
  - Using dependency injection
  - Dependency injection in ASP.NET Core
- Implementing CRUD operations
- Validation on the server- and client-side
  - Implementing custom validation
- Reusing view components
  - TagHelpers, Partial view
  - Implementing custom taghelpers
- Routing rules
- Filters
  - Types of filters, interfaces
  - Order of filters
  - Creating ActionFilters, ExceptionFilters
- Configuring
  - Using appSettings
  - Options pattern
  - Logging
- Middlewares
  - Overview of the ASP.NET Core Middleware architecture
  - Creating custom middlewares
  - Overview of the template middlewares
  - Error handling using middlewares
  - Other built-in middleware-s (session, cookiepolicy)

- Authentication and authorization
    - Expanding the data model with user management
    - Configuring middlewares
    - Registration and login
    - Role-based authorization
    - Claims-based authorization

Developing REST applications using ASP.NET Core

Length: 5 days

Prerequisites: Solid knowledge of C# and the .NET framework, confidence using Visual Studio

Software requirements: Visual Studio 2019 or newer and ASP.NET Core SDK 3.1 or newer, with local SQL Express (or other SQL database server) installed, SQL Server Management Studio recommended

Description: During the course attendants get to know the modern, new-generation ASP.NET Core web development framework and associated design and architectural patterns. so that they can develop well-structured, testable web applications.

Topics:

- Creating endpoints
    - Data access
    - Connecting components
    - Loose-coupling, reusability, creating unit testable components
    - Using dependency injection
- Implementing CRUD operations
- Validation
- Headers, http methods
- Routing rules
- Filters
    - Types of filters, interfaces
    - Order of filters
    - Creating ActionFilters, ExceptionFilters
- Configuring
    - Using appSettings
    - Options pattern
    - Logging
- Middlewares
    - Overview of the ASP.NET Core Middleware architecture
    - Creating custom middlewares
    - Overview of the template middlewares
    - Error handling using middlewares
    - Other built-in middleware-s (session, cookiepolicy)
- Authentication and authorization
    - Expanding the data model with user management
    - Configuring middlewares
    - Registration and login
    - Role-based authorization
    - Claims-based authorization
- Testing with Fiddler and PostMan
- Creating client-side applications using .NET Core
    - Configuring Swagger documentation

Secure application development in .NET

Length: 4 days

Prerequisites: Strong C# and .NET knowledge, confidence using Visual Studio, high-level ASP.NET Core knowledge, knowledge of database design and the HTTP protocol

Software requirements: Visual Studio 2017 or newer

Description: The goal of the course is to prepare students for secure web application development. During the course we look at the basic concepts of software security, how the most common attacks are carried out and how can they be prevented. The possible defensive mechanisms are discussed not only in theory but are also implemented in practice so it can be discussed in detail how they work related to software security.

Topics:

- Introduction
    - Application security in software development, common attacks, threats
    - Secure application development
    - Software security models, frameworks, standards
    - Microsoft Security Development Lifecycle
- Input validation
    - General practices
    - ASP.NET Core validation
    - SQL Injection attacks
    - XSS, content security policy
- Authentication and authorization
    - ASP.NET Core Identity framework
        - Structure of the framework
        - Custom user database
        - External authentication
    - IdentityServer
    - Role-based authorization
    - Claims-based authorization
- Cookie handling, HSTS
- Basics of cryptography
    - Basics methods
    - Hashing
    - Digital signatures, certificates
- Secure session handling
- Error handling best practices
- Introduction to static and dynamic application security testing

Language elements of the modern C# language

Length: 4 nap

Prerequisites: Solid C# knowledge, confidence using Visual Studio, basic knowledge of the .NET framework

Software requirements: Since the purpose of the course is to demonstrate the newest language elements of the C# language, it is recommended to set up the newest version of both Visual Studio and the .NET Core frameworks.

Description: The purpose of the course is to give an overview of the new versions of the C# programming language with their respective new features.

Topics:

New features of the C# language versions

- New features of C# 6
  - New features for properties
  - Null-conditional operator
  - Expression-bodied members
  - Nameof, using static
  - Exception filter
  - Other useful small features
- New features of C# 7 and 7.1
  - Pattern matching
  - tuple
  - local functions
  - ref returns
  - async main
- New features of C# 7.2
  - Reference semantics using value types
- New features of C# 7.3
  - new generic constraints
- New features of C# 8
  - nullable reference types
  - async enumerable
  - range and index
  - default interface implementation
  - ?=, using declaration and other useful small features
- New features of C# 9
  - using record types
  - init properties
  - covariant return types
  - target-typed new
  - source generators

    - other small features
- New features of C# 10
    - record struct
    - global using directives
    - file-scoped namespace
    - interpolated string constant
    - pattern matching improvements

Developing applications using Microsoft Azure

Length: 4 days

Prerequisites: Solid C# programming knowledge, confidence using Visual Studio, basic knowledge of .NET development, databases and web development; Azure account for every participant

Software requirements:

- Visual Studio 2015 or newer
- Azure Storage emulator
- Azure Storage Explorer
- CosmosDb emulator
- AzCopy V10
- Visual Studio Code
- Azure CLI
- Git command line client
- FTP client

Description: During the course attendants get an introduction to the possibilities of the Microsoft Azure cloud platform and their usage in modern application development: data storage using relational and non-relational databases, developing, deploying, operating and managing web and other PaaS services.

Topics:

- Brief overview of the Azure platform
- Introduction to Azure Storage Services
    - Overview of storage accounts
    - Application development using Azure blob storage, direct access, shared access tokens
    - Using Azure Table storage in modern applications
    - Using Azure Queue Storage in distributed applications
    - Using Azure File Storage
- Azure App Services
    - Overview of Azure App Service plans
    - Deploying web apps to app service plans
    - Scaling, backups
    - Continuous delivery, integrating with version control systems, deployment slots
    - Creating and running web jobs
    - Serverless programming using Azure functions
    - Deploying applications from the marketplace
- Modern data storage solutions
    - Azure SQL
    - Application development using Azure Redis Cache
    - Data storage in Azure Cosmos Db
    - Search-as-a-Service with Azure Search
- Azure Cognitive services

Data management using Entity Framework Core

Length: 5 days

Prerequisites: Solid C# knowledge, confidence using Visual Studio, basic .NET Core knowledge, basic knowledge of relational databases

Software requirements: Visual Studio 2019 or newer, Sql Server Management Studio, Sql Server Profiler

Description: The course gives an overview of the Entity Framework Core ORM: creating models, tracking changes of models, handling entities (CRUD operations), N-layer applications, advanced techniques.

Topics:

- Overview and configuration of Entity Framework Core
- Creating Entity Framework Core models
- Scaffolding from existing databases
- Conventions
- Configurating relationships
- Shadow state
- Lazy-loading, eager loading, explicit loading
- Entity tracking, using Entity Framework in multitier applications
- Advanced mapping (inheritance, splitting, owned entities)
- Interceptors, query filters
- EF Core migrations
- Exercise-day

C# programming basics

Length: 5 days

Prerequisites: Basic programming and IT knowledge

Software requirements: Visual Studio 2013 or newer

Topics: The purpose of the course is to introduce the attendants to the basics of the C# programming language and the .NET framework so that by the end of the course they are able to create simple solutions using Visual. During the course an overview of the basic programming constructs, object-oriented programming, user and file input handling is given.

Topics:

- Introduction
  - Introduction to the Visual Studio IDE
  - Creating console applications
  - Source files, project, solution
  - Assemblies; relationship of .NET and C#
  - Overview of debugging possibilities
- Basics of the C# language
  - Overview of the language syntax
  - Using local variables
  - Introduction of C# primitive types: operations, precedence
  - Using arrays
  - Using conditional expressions
  - Using loops
  - Creating „functions”
- Object-oriented programming
  - Basics of the object-oriented paradigm
  - Creating classes and objects
  - Encapsulation, data hiding, accessibility levels
  - Inheritance
  - Polymorphism; virtual and abstract methods, abstract classes and interfaces
  - Static members
- Other elements of the .NET type system
  - Delegates
  - Events
  - Enums
  - Generics
- Structured exception handling
  - Role of exceptions
  - Catching and handling exceptions
  - Throwing exceptions
  - Creating custom exception types
- File-handling

- Overview of streams
- Opening files, reading contents
- Writing to files
- Handling unmanages resources
- StreamReader, StreamWriter, File API

Advanced C# programming

Length: 5 days

Prerequisites: see C# programming basics topics

Software requirements: Visual Studio 2013 or newer

Description: The purpose of the course is to help attendants build on their previous, introductory knowledge of C# and .NET and show them advanced solutions to frequent problems.

Topics:

- Language Integrated Query
  - Evolution of LInQ
  - Language elements that support LInQ
  - Lambda expressions
  - LInQ operators
  - Solving problems related to collections using LInQ in a declarative way
  - Datasource-independent structure
- Asynchronous and parallel programming
  - Introduction to the Task parallel library
  - Composing tasks
  - Using the Parallel class
  - C# async-await keywords
  - Basics of mutual exclusion
- Exercise-day

Expert C# programming

Length: 5 days

Prerequisites: see Advanced C# programming topics

Software requirements: Visual Studio 2015 or newer and WinDbg

Description: The purpose of the course is to provide a deeper description of the .NET framework and C# programming language so that attendance can solve not so frequent, but hard and very influential problems with elegant and performant solutions.

Topics:

- Co- and contravariance in .NET
- Advanced threading
    - Advanced TPL, TaskCompletionSource, using the Parallel class
    - Using concurrent collections
    - Mutual exclusion and synchronization methods
        - Interlocked
        - Distinction of user and kernel mode methods
        - Implementing custom methods
- Handling IL-code
    - Reading IL-code, basics of the language and the framework
    - Lightweight IL-generation
    - Generating full assemblies
    - Using Castle.DynamicProxy
    - Solving frequent, performance-critical problems
- Advanced debugging
    - Using S.O.S. and SOSEx
    - Dump files
    - Diagnosing memory and threading issues

Dependency injection in .NET

Length: 5 days

Prerequisites: Solid C# knowledge, confidence using Visual Studio, basic .NET framework knowledge strong object-oriented knowledge.

Software requirements: Visual Studio 2013 or newer

Description: During the course attendants get an overview of dependency injection, its purpose, tools, architectural aspects, application and usage. An introduction is given into the patterns and anti-patterns and based on these we discuss how it should be used in applications with the help of widely used dependency injection containers.

Topics:

- Object-oriented programming: revision
    - Encapsulation, inheritance, polymorphism
    - S.O.L.I.D
- Dependency injection: architectural overview
    - What is dependency injection? Facts and misconceptions
    - Purpose, advantages, applications
    - Patterns, anti-patterns, refactoring
    - DI 3D: Object composition, lifetime management, interception
    - Dependency injection containers
- Dependency injection in application development
    - Implementing dependency injection containers
    - Using dependency injection containers in different applications
        - ASP.NET MVC
        - ASP.NET Core
        - Console applications
        - WPF
        - Powershell
    - Interception
        - Decorator pattern
        - Circuit breaker pattern
        - Dynamic proxies
- Overview of Microsoft.Extensions.Dependency container
    - registration
    - resolving
    - lifecycle management
    - decorators, composites, sequences
    - parameterized registration
- Overview of the Autofac dependency injection container
    - registration
        - Reflection, Lambda, Instance components
        - Generic types

        - Passing parameters
        - Assembly scanning
        - Advanced registrations
    - Resolving
        - Passing parameters
        - Implicit relationships
        - Lifecycle events
    - Lifecycle management
    - Advance application methods
        - Aggregate services
        - Interception using dynamic proxies
        - Adapters and decorators
        - Circular references

Design patterns in C#-.NET

Length: 4 days

Prerequisites: Solid C# programming knowledge, confidence using Visual Studio, basic .NET knowledge, confident knowledge of object-orientation basics

Software requirements: Visual Studio 2013 or newer

Description: During the course we discuss the classic object-oriented design patterns, their implementation, functionality, and application in complex systems. We start by diving into high-level object-oriented principles, then implement the classic „Gang-of-four" patterns, finally look at how to combine these patterns in more complex systems.

Topics

- Programming basics: creating reusable components
- Object-oriented principles
- Defining design patterns
- History of design patterns: Gang of Four
- Design pattern types (creational, structural, behavioral)
- Object-oriented design patterns
    - Adapter
    - Bridge
    - Builder
    - Chain of Responsibility
    - Command
    - Composite
    - Decorator
    - Facade
    - Factory Method
    - Flyweight
    - Interpreter
    - Iterator
    - Mediator
    - Memento
    - Abstract Factory
    - Observer
    - Prototype
    - Proxy
    - Visitor
    - Singleton
    - State
    - Strategy
    - Template Method
- System of patterns, creating complex systems

Developing unit testable applications using .NET framework

Length: 5 days

Prerequisites: Solid C# programming knowledge, confidence using Visual Studio, basic .NET knowledge, confident knowledge of object-orientation basics

Software requirements: Visual Studio 2013 or newer

Description: The purpose of the course is to discuss how to create well-structured applications that can be unit tested in a performant, automated way. In order to achieve this, the course discusses many related topics, such as object-oriented design principles and patterns, dependency injection, mocking and many others. The course also discusses the methods of creating and implementing automated unit test.

Topics

- Fundaments of unit testing
    - What is a unit test?
    - What is not a unit test?
    - Connection to other test types
    - When to unit test?
    - Overview of test design techniques (black box, white box techniques)
- Developing unit testable applications
    - Requirement of loose-coupling
    - Using interfaces
    - Related design patterns (Repository, unit of work)
    - Principle of inversion of control
    - Dependency inversion as the fundamental principle of loose coupling
    - Inversion of control and dependency inversion with dependency injection
        - Dependency injection containers in .NET
        - Managing object lifetime
    - Handling dependencies: mock, stub, fake, dummy
        - Mock-framework
    - Creating unit tests
        - Using knowledge from previous section (dependency injection, mocks)
        - Unit testing framework
- Refactoring
    - What is refactoring? Why refactor?
    - OO basics (encapsulation, inheritance, polymorphism, S.O.L.I.D.)
    - Basic techniques
        - Renaming
        - Extract (interface, class, method or others)
        - Push-down, pull-up (method, field and others)
        - Replace (interchanging coding elements)
        - Using polymorphism instead of branching
        - Using factory method or abstract factory
        - Using strategy

Data management using Entity Framework

Length: 4 days

Prerequisites:: Solid C# knowledge, confidence using Visual Studio, basic knowledge of relational databases

Software requirements: Visual Studio 2013 or newer, Sql Server Management Studio, Sql Server Profiler

Description: The course gives an overview of the Entity Framework ORM: creating models, tracking changes of models, handling entities (CRUD operations), N-layer applications, advanced techniques.

Topics:

- Overview of ORM frameworks
- Introduction to entity framework
- Entity Framework Code First
    - Creating and configuring models
    - Conventions, creating custom conventions
    - Handling model changes
- Brief overview of other modelling possibilities
- Entity handling, state management, lazy loading and proxy generation
- Error handling, performance optimization
- Developing multitier applications using Entity Framework
- Exercise-day

Developing REST services using ASP.NET Web API 2

Length: 5 days

Prerequisites: Solid knowledge of C# and the .NET framework, confidence using Visual Studio

Software requirements: Visual Studio 2013 or newer, with local SQL Express (or other SQL database server) installed, SQL Server Management Studio recommended

Description: During the course attendants get to know the ASP.NET WEB API 2 web development framework and associated design and architectural patterns. so that they can develop well-structured, testable web services and client applications.

Topics:

- Introduction
  - MVC architectural pattern
  - Web development using Visual Studio
- Defining the data model and data access layer
  - Short introduction to ORM frameworks
  - Entity Framework model
  - Configuring the model
  - Repository - Unit of work design patterns
- Creating the business logic
  - IoC container, Dependency injection
  - Using the EF model
  - Unit testing in ASP.NET Web API 2
- Controllers and actions
  - Creating controllers
  - Using the business logic and the data model
  - Defining actions, HTTP GET, POST, PUT, DELETE
  - Parameters, model binding
  - Routing, route constraints, controller factory
  - Error handling, logging
  - Using filters (ActionFilter, Resultfilter)
  - HttpContext, RouteData, TempData
  -
- Validation
  - Using ModelState, server side validation using DataAnnotations and ValidatableObject
- Authentication and authorization
  - ASP.NET Identity
- Developing client applications
  - Client-side testing using Fiddler
  - Creating client applications, connecting to the server
  - Authenticatiion
  - SPA development using Knockout.js

Developing web portals using ASP.NET MVC 5

Length: 5 days

Prerequisites: Solid knowledge of C# and the .NET framework, confidence using Visual Studio

Software requirements: Visual Studio 2013 or newer, with local SQL Express (or other SQL database server) installed, SQL Server Management Studio recommended

Description: During the course attendants get to know the ASP.NET MVC web development framework and associated design and architectural patterns. so that they can develop well-structured, testable web applications.

Topics:

- Introduction
  - MVC architectural pattern
  - Web development using Visual Studio
- Defining the data model and data access layer
  - Short introduction to ORM frameworks
  - Entity Framework model
  - Configuring the model
  - Repository - Unit of work design patterns
- Creating the business logic
  - IoC container, Dependency injection
  - Using the EF model
  - Unit testing in ASP.NET MVC
- Controllers and actions
  - Creating controllers, AsyncController
  - Using the business logic and the data model
  - Defining actions, HTTP GET, POST
  - Implementing CRUD operations
  - Parameters, model binding
  - Routing, route constraints, controller factory
  - Using filters (ActionFilter, Resultfilter, Exception filter)
- View
  - Razor View Engine, syntax, functionality
  - Layout pages
  - Templates, scaffolding, using HthmlHelpers
  - Partial view
  - ViewData, ViewBag
  - Creating custom HtmlHelpers
- Client-side support in ASP.NET MVC
  - CSS and Javascript bundle
  - AJAX Controller Action
  - AJAX Helpers
- Validation

    - Server- and client-side Validation using DataAnnotations
    - IValidatableObject-based validation
    - Remote validation, client-side validation
- Authentication and authorization
    - Windows authentication
    - Implementing user management with ASP.NET Identity

Operating web portals using Internet Information Services

Length: 3 days

Prerequisites: Basic knowledge of static and dynamic websites, HTTP-knowledge

Software requirements: Windows 10 operating system

Description: The course gives an introduction into the basic services of Internet Information Services so that attendants will be able to deploy, manage and operate web based services and also into the common best-practices of hosting web based and FTP-based services.

Topics:

- Overview of the world wide web
    - Network protocols
    - DNS
    - HTTP
    - Firewalls
- IIS basics
    - Creating websites
    - Website binding
    - Hosting static content
    - Hosting dynamic content
    - App pools and their configuration
- Security
    - IIS authentication: Anonymous, Basic, Digest, Windows
    - Installing SSL certificates
    - SSL binding
    - Client-side certification
- Monitoring, logging, operations-management
    - Logging
    - Parsing W3C logs
    - Failed requests, long running requests
    - Performance counters
    - Remote management
    - Backups
    - Performance optimization
- Operating FTP-sites
- Load-balancing, high-availability

Build automation using MSBuild

Length: 4 days

Prerequisites: For the „Custom tasks and loggers" section basic knowledge of the C# programming language and Visual Studio; for the other sections basic IT-knowledge (using command line; XML files, basic file management, environment variables)

Software requirements: Visual Studio 2015 or newer

Description: The purpose of the course is to give an in-depth introduction into the MSBuild build engine. After the course, attendants, can create MSBuild projects and extend it with custom steps, elements or components and implement error handling and diagnostic to create better performing solutions.

Topics:

- MSBuild basics
  - MSBuild command line
  - Properties, Items, Metadata, Tasks and Targets
  - Conditions
  - Item functions, property functions, MSBuild functions
  - Import
  - Error handling
- Extending the MSBuild process
  - PreBuildEvent & PostBuildEvent
  - Target overriding
  - Target hooks
  - Target injection
- Custom tasks and loggers
  - Creating ciustom tasks: ITask, Task, ToolTask
  - Inline tasks, implementing TaskFactory
  - Custom loggers
  - Binary logger
  - Publishing tasks suing NuGet packages
- Batching
  - Target batching
  - Task baching
  - Multibatch
  - Shared metadata
  - Incremental and partial build
- MSBuild in practice
  - Solution build
  - Performance diagnostics
  - Built-in tasks: copy, message, exec, msbuild
  - Extensions: MSBuild Community Tasks, MSBuild Extensions; NUnit task
  - Creating reusable target files
  - Setting assembly version

- Zipping output
- Uploading to an FTP site or Azure
- Encrypting web.config files

Windows Presentation Foundation

Length: 5 days

Prerequisites: Solid C# knowledge, confidence using Visual Studio, strong .NET knowledge

Software requirements: Visual Studio 2013 or newer

Description: The purpose of the course is to give an in-depth introduction into the WPF framework and show how to implement loosely-coupled, testable desktop applications with reusable components focusing on the MVVM architectural pattern.

Topics:

- XAML basics: syntax, structure
- Some basic Frameworkelements
    - element composition
    - layout controls
    - item controls
- XAML structuring
    - styles
    - datatemplate, datatemplate selector
    - resources, resource dictionary
    - triggers
    - control template
    - using and creating dependency és attached property
    - creating user controls
    - event handling
    - globalization and localization
- Developing application using the MVVM architecture
    - MVVM basics, defining components
    - Basics of data binding: updating, mode, error handling
    - Validation
    - Command design pattern, using commands
    - Attached command behavior
    - Event to command
    - Messaging
- Advanced MVVM
    - Dependency injection in MVV, using the proper abstraction, testability
    - creating custom markup extensions
    - custom XAML parsing
- Asynchronous MVVM
    - Asynchronous data binding
    - Asynchronous commands

Windows Communication Foundation

Length: 3 days

Prerequisites: Solid C# knowledge, confidence using Visual Studio, strong .NET knowledge

Software requirements: Visual Studio 2013 or newer

Description: The purpose of the course is to give and introduction into the fundamentals of the service-oriented architecture and based on this show how to implement WS-* web services using the Windows Communication Foundation Framework. During the course both the client and server side tools are discussed.

Topics:

- Introduction, theoretical definitions
- Defining endpoints, Address, Binding, Contract
- Adding behaviors
- Discussion of WSDL
- Discussion of hosting options (self-hosted, IIS)
- Diagnostics, logging, tracing
- Error handling, FaultContract, IErrorHandler
- Managing lifecycle of service host object
- Extending the WCF pipeline, custom behaviors
- WCF Security, discussion of message and transport layer solutions
- Duplex communication
- Dependency injection in WCF using Autofac

Developing web applications using Blazor

Length: 5 days

Prerequisites: Solid C# knowledge, confidence using Visual Studio, basic .NET framework knowledge

Software requirements: Visual Studio 2022 or newer, ASP.NET Core SDK 6.0 or newer

Description: The purpose of the course is to give an introduction to the new and emerging Blazor technology.

Topics:

- Introduction to Blazor, frequent tasks in client-side developemnt
    - Blazor Webassembly, Blazor server and other Blazor hosting models
    - Blazor components, lifecycle
    - Using HttpClient properly, HttpClientFactory
    - OpenAPI
    - CORS
- Creating Blazor webassembly components
    - Creating components
    - Displaying data, one-way binding
    - RenderFragment
    - Event handling
    - Parameters, two-way binding
    - Cascading parametes
    - Error handling
- Developing Blazor webassembly applications
    - Dependency injection int Blazor
    - Creating pages, routing
    - Handling layouts
    - Creating forms, validation
    - Authentication and authorization
    - Localization
- Testing Blazor components
    - Using bUnit
- Other tools of Blazor webassembly
    - AOT compilation
    - Lazy-loading assemblies

Application Development Using MAUI

Length: 5 days

Prerequisites: Solid C# knowledge, confidence using Visual Studio, basic .NET framework knowledge

Szoftveres igény: Visual Studio 2022 or newer, ASP.NET Core SDK 7.0 or newer

Leírás: A tanfolyam célja, hogy a hallgatók megismerkedjenek a MAUI keretrendszerrel, modern, cross-platform alkalmazások fejlesztésének lehetőségeivel és az alkalmazások megfelelő, hosszú távú karbantarthatóságot elősegítő felépítésével.

Tematika:

- Intorduction to MAUI
    - Application development possibilities
    - Cross-platform phylosophy
    - Creating platform-specific code
    - Creating shared code
- Introduction to XAML
    - XAML basics, syntax
    - Styles, resoruces, data binding
    - MAUI controllers
    - MAUI layouts
    - MAUI pages
- MVVM application developmnet
    - MVVM architecture
    - Advanced data binding
    - Using dependency injection
    - Testing
- Application development using WinUI

Application development using Blazor

Modern communication solutions

Length: 5 days

Prerequisites: Solid C# knowledge, confidence using Visual Studio, basic .NET framework knowledge , ASP.NET Core basic knowledge, basic grasp of HTTPS protocol and HTTP -based client-server applications.

Software requirements: Visual Studio 2022, .NET 7

Description: A tanfolyam célja, hogy a hallgatók megismerjék a hagyományos HTTP alapú alkalmazásfejlesztés alternatíváit, egyéb elosztott módon működő rendszerek fejlesztésébe betekintést kapjanak és a legmodernebb, alkalmazások közötti kommunikációs protokollokat, eszközöket megismerjék .NET-ben.

Topics:

- Introduction to Websockets
- Websockets in .NET: SignalR
    - Creating the server-side
    - Connecting to SignalR application from the client -side
    - SignalR backplane
    - Authentication methods
- Brief description of HTTP/2
- Creating HTTP/2 applications in .NET
- HTTP/2 in practice: gRPC
    - creating gRPC applications in .NET
    - connecting to gRPC applications
    - Authentication methods
- Overview of QUIC and HTTP/3
    - Creating HTTP/3 applications in .NET

Microsoft Certified Professional exam study courses

Microsoft Certified Professional exam preparation courses based on the official Microsoft syllabi.

Topics:

- Programming C# 70-483
- Azure exam preparation
- Mobile and web development

Comprehensive list of courses: https://docs.microsoft.com/en-us/learn/certifications/browse/