Index

Contents

.NET Core with Azure Curriculum

.NET Core 6.0 and C# 10.0

Program Duration: 9 Days.

Table of contents

    - Parse() vs TryParse() vs Convert Class methods
- Debugging in C#
    - Various Types of .NET Projects
    - Tracing, Debugging, Build
    - Compile Options
    - Using break points
    - Using break conditions
    - Using watch and output window
    - Creating multiple projects within one solution
    - Customizing Visual Studio Settings – Extensions, NUGet Package, Environmental Settings
- OOP with C#
    - Structures and enums
    - The architecture of a class in C#
    - Instance, Class & Reference variables
    - Access Modifier
    - Abstract Classes
    - Constructors, Destructors, The GC
    - .NET Base class library
    - Inheritance in C#
    - Method Overloading
    - Method Overriding
    - Operator Overloading
    - Method Hiding
    - Access modifiers : private, pubic, protected, internal, protected internal, new
    - Anonymous types
    - Abstract classes
    - Sealed classes
    - Creating Interfaces
    - Implementing Interface inheritance
    - Declaring properties within Interfaces
    - Namespaces
    - Creating and using Generic classes
    - Indexers & Properties
    - Auto Implemented properties
    - Static Classes
    - Property Accessors
    - Partial types
    - Extension methods
    - Object Initializer
- Evaluating Regular Expressions in C#
    - RegEx Class
    - Forming Regular Expression
    - Methods for Regular Expression
- Exception Handling
    - Exceptions in C#
    - Exception class hierarchy
    - Try block
    - Multiple catch blocks
    - Finally block

    - Purpose of throw keyword
    - Purpose of inner exception
    - Creating Custom Exception
- Garbage Collection in C#
    - Role of a Garbage Collector
    - Garbage Collection Algorithm
    - Finalize vs Dispose
- Collections & Generics
    - System.Collections Namespace
    - Collection Interfaces
    - Collection Classes
    - The collection API
    - Working with Generics
    - Creating Generic class, Generic Methods, Interfaces, Delegates
    - Collection Initializers
    - Iterators
    - Constraints
- Anonymous Types, Delegates, Events & Lambda
    - Extension Methods
    - Anonymous Type
    - Var and Dynamic
    - Introduction to Delegates
    - Events in C#
    - Anonymous Methods
    - Lambda Expression
    - Expression Tree
- File I/O and Serialization
    - Using StreamReader, StreamWritter
    - Using BinaryReader, BinaryWriter
    - Using File, FileInfo, Directory, DirectoryInfo
    - Serialization modes: Binary, SOAP, XML
    - JSON serialization
- Threading, Parallel and Async programming with C#
    - Task Parallel Library
    - Threads Vs. Tasks
    - Task Based Asynchronous Model
    - Async and Await
    - Using Locks
- New Features in C# 8.0
    - Readonly members
    - Default interface methods
    - Pattern matching enhancements:
    - Using declarations
    - Static local functions
    - Disposable ref structs
    - Nullable reference types
    - Asynchronous streams
    - Asynchronous disposable
    - Indices and ranges
    - Null-coalescing assignment

    - Unmanaged constructed types
    - Stackalloc in nested expressions
    - Enhancement of interpolated verbatim strings
- New Features in C# 10.0
    - Record structs
    - Improvements of structure types
    - Interpolated string handlers
    - global using directives
    - File-scoped namespace declaration
    - Extended property patterns
    - Improvements on lambda expressions
    - Allow const interpolated strings
    - Record types can seal ToString()
    - Improved definite assignment
    - Allow both assignment and declaration in the same deconstruction
    - Allow AsyncMethodBuilder attribute on methods
    - CallerArgumentExpression attribute
    - Enhanced #line pragma
    - Warning wave 6

SOLID Principles and Best Practices

Program Duration: 2 Days.

Table of contents

- SOLID Principles
- Coding Conventions
- Best Practices

Git

Table of contents

Program Duration: 1 Day.

- Getting Started with Git
    - Install the Git Tools
    - Clone an Existing Repository
    - Add Files to a Repository
    - Edit Files in a Git Repository
    - Create and Merge Branches
    - Rewrite History in a Git Repository
    - Resolve Merge Conflicts

RDBMS & SQL Server

Program Duration: 4 Days.

Table of contents

- Introduction to RDBMS
    - Introduction to databases
    - Data Models in Database
    - Properties of RDBMS
    - Normalization
    - CODD's Relational Database Rules
    - Data Integrity
    - T-SQL Language
- Working with Data Types, Tables & Data Integrity covering DDL, DML, DCL statements
    - Working with Data Types (Only Basics of Data Types)
    - Working with Schema
    - Working with Tables
    - Implementing Data Integrity
- Beginning with Transact-SQL
    - Transact-SQL
    - System Functions
    - Advanced T-SQL Queries`
    - Advanced T-SQL Statements
    - Other T-SQL Statements
    - Set Operators
    - Transact-SQL
    - System Functions
    - Advanced T-SQL Queries
    - Advanced T-SQL Statements
    - Other T-SQL Statements
- Working with Joins and Subqueries
    - What are Joins?
    - Types of joins
    - Subqueries
- Database Objects: Indexes and Views
    - Introduction to Index in SQL Server
    - Introduction to Views in SQL Server
- Stored Procedures
    - Stored Procedure
    - Implementing Stored Procedure
    - Exception handling using TRY-CATCH
- NoSQL Database
    - Brief History of NoSQL Databases
    - NoSQL Database Features
    - Types of NoSQL Database
    - Difference between RDBMS and NoSQL
    - Why NoSQL?
    - When should NoSQL be Used?
    - Demo
- Azure SQL Database:
    - Introduction to Azure SQL Database
    - Demo

LINQ and Entity Framework Core

Program Duration: 2.5 Days.

Table of contents

- Raw SQL And Stored Procedures

ASP.NET Core Web API

Program Duration: 3 Days.

Table of contents

- Introduction to .Net Core WebAPI
  - Introduction to Web Service
  - Introduction to REST API
  - Introduction to Web API
  - Difference between Web Service, WCF Service and Web API
  - HTTPS Verbs
  - Web API Routing
  - Configuring WebApi
  - Testing the Web API Project with Postman and Swagger
  - Building first ASP.NET Core Web API
  - Fluent Validation
- Working with Relational Data using Entity Framework Core
  - Relationships in EF Core
  - HTTP Response Status Codes
  - Try-Catch-Finally block
  - Throwing custom exceptions
  - Global error handling
  - Custom global error handling
  - DML Manipulation using Repository Pattern
- Controller Action Return Types
  - Introduction to Controller Action Return Types
  - Specific Type
  - IActionResult
  - ActionResult<Type>
  - Custom Return Type
- Web API Versioning
- Web API Logging
- Unit Testing in Web API
- Building Microservices

Angular 10

Program Duration: 6 Days.

Table of contents

- Introducing TypeScript
  - TypeScript Syntax
  - Programming Editors
  - The Type System – Defining Variables
  - The Type System – Defining Arrays

    - Type in Functions
    - Type Inference
    - Defining Classes
    - Class Methods
    - Visibility Control
    - Class Constructors
    - Class Constructors – Alternate Form
    - Interfaces
    - Working with ES6 Modules
    - var vs let
    - Arrow Functions
    - Arrow Function Compact Syntax
    - Template Strings
    - Generics in Class
    - Generics in Function

- Introducing Angular
    - What is Angular?
    - Central Features of the Angular Framework
    - Appropriate Use Cases
    - Building Blocks of an Angular Application
    - Basic Architecture of an Angular Application
    - Installing and Using Angular
    - Anatomy of an Angular Application
    - Running the Application
    - Building and Deploying the Application
- Components & Templates
    - Creating a Component Using Angular CLI
    - The Component Class
    - The @Component Decorator
    - Registering a Component to Its Module
    - Component Template
    - Using a Component
    - Component Hierarchy
    - Component Lifecycle Hooks
    - Template Location
    - The Mustache {{ }} Syntax
    - Setting DOM Element Properties
    - Setting Element Body Text
    - Event Binding
    - Expression Event Handler
    - Attribute Directives
    - Structural Directives
    - Looping Using ngFor
    - Grouping Elements
    - Template Reference Variable
    - @Output() - Child Component
    - @Output() - Parent Component
    - Full Two Way Binding
    - Setting up Two Way Data Binding in Parent

- Template Driven & Reactive Forms
    - Template Driven Forms
    - Importing Forms Module
    - Two Way Data Binding
    - Form Validation
    - Angular Validators
    - Displaying Validation State Using Classes
    - Additional Input Types
    - Reactive Forms Overview
    - Import ReactiveFormsModule
    - Getting Input Values
    - Setting Form Values
    - Validation
    - Using a Custom Validator
    - Sub FormGroups - Component Class
    - Sub FormGroups - HTML Template

- Services & Dependency Injection
    - The Service Class
    - What is Dependency Injection?
    - Injecting a Service Instance
    - Injectors
    - Dependency Injection in Other Artifacts
    - Providing an Alternate Implementation
- Pipes & Data Formatting
    - Built-In Pipes
    - Using Pipes in HTML Template
    - Chaining Pipes
    - Using a Pipe with ngFor
    - A Filter Pipe
- Angular Routing & Angular Modules
    - The Router Component
    - The Angular Router API
    - Creating a Router Enabled Application
    - Passing Route Parameters
    - Anatomy of a Module Class
    - @NgModule Properties
    - Using One Module from Another
- HTTP Client
    - The Angular HTTP Client
    - Importing HttpClientModule
    - Service Using HttpClient
    - Making a GET Request
    - Observable Object
    - Error Handling & Customizing the Error Object
    - Returning an HttpResponse Object
    - Creating New Observables
    - Observable Operators

    - The map and filter Operators
- Observables & RxJS Library
    - Observables Overview
    - Observables in Angular
    - Introduction to RxJS library
    - Angular Authentication With JSON Web Tokens (JWT)

Azure DevOps

Table of contents

Program Duration: 1 Day

- Azure DevOps
    - Introduction to Azure DevOps
    - Why Azure DevOps?
    - Components of Azure DevOps
        - Pipelines
        - Boards
        - Artifacts
        - Repos
        - Test Plans

Cloud Computing & Azure Fundamentals

Program Duration: 2 Days.

Table of contents

- Cloud Computing & Microsoft Azure Fundamentals
- Microsoft Azure Portal Overview (Utilization, Cost)
- Overview of Cloud Computing (Benefit, Capex/Opex)
- Overview of Public and Private Cloud & Hybrid
- Core Azure Service
- Availability Zones, Availability Set, Resource Groups, Azure Resource Manager
- Security & Governance Overview
- Identity, Azure Active Directory, Users & Groups
- Subscriptions and Accounts, Azure Policy, Role-based Access Control (RBAC)
- Azure Portal and Cloud Shell & Azure PowerShell and CLI
- Azure Key Vault Services

Cloud Networking & Storage

Program Duration: 4 Days.

Table of contents

- Azure Virtual Machine

    - Create Virtual Machine
    - Manage Virtual Machine
    - Create/Manage Disks
    - Create VM images
    - Create Scale Set
    - Load Balance VMs
- Virtual Network
    - Create Virtual Network (Portal, PowerShell & CLI)
    - Add, Change or Delete a Subnet
    - Connect Virtual Networks
        - Virtual Network Peering
    - Filter Network Traffic
    - Route Network Traffic
        - Route Table
    - Restrict Virtual Network
    - Secure Network Traffic
        - Network Security Groups
        - Application Security Groups
        - Virtual Network Service End Points
    - Monitor Virtual Networks
        - Virtual Network Terminal Access Point (TAP)
- Azure Load Balancer
    - Load Balancer Algorithm
    - Load Balancer Components
    - Create Public & Internal Load Balancer
    - Standard Load Balancer & Availability Zones
- VPN Gateway, Azure Application Gateway
    - Create & Manage VPN Gateway
    - Create Site to Site VPN Connection
    - Gateway Configuration Settings
    - Create Application Gateway
    - Support High Traffic Volumes
    - Autoscaling and Zone-redundant Application Gateways
- Azure CDN
    - Create an Azure CDN Profile and endpoint
    - Monitor health of Azure CDN Resources
    - Azure Diagnostic Logs
    - Azure CDN Usage Patterns
- Azure Traffic Manager
    - Create Traffic Manager Profile
    - Traffic Manager Routing Methods
    - Traffic Manager EndPoints
    - Enable Resource Logging

- Azure Storage Services
    - Core Storage Services
        - Azure Blobs
        - Azure Files
        - Azure Queues
        - Azure Tables

        - Azure Disks
    - Creating Storage Account
    - Azure Blob Storage
        - Work with blobs
        - Upload, download & list blobs
        - Encrypt & Decrypt blobs using Azure Key Vault
    - Introduction to Azure Data Lake Storage
    - Authorize access to Azure Storage
    - Choosing Data Storage Technology in Azure
    - Structured Data, Semi Structure Data, Unstructured Data
    - Accessing Azure Storage using Azure SDK & C#.NET
- Azure API Management
    - API Management Features
    - Create an Instance
    - Manage API Management
        - Import and Publish First API
        - Mock API response
        - Monitor Published APIs
    - Self-Hosted Gateway Overview
    - Overview of Products
    - Creating and Publishing Products
        - Add an API to an existing Product

Azure PaaS Services

Program Duration: 5 Days.

Table of contents

- Azure Web App
    - What is App Service
    - App Service Plan and Comparison between them
    - App Service Environments
    - Creating ASP.NET Web APP
    - Deploying App Using Visual Studio & Kudu
    - Run App in Staged Environments using deployment slots
    - Working with Configurations
    - Accessing Configuration using ASP.NET Application
    - Host API with CORS
    - Creating and Using Web Jobs
- Azure Function App
    - Overview of Serverless Computing & Benefits
    - Create Function App using C# in Azure Portal
    - Deploying Function App using Visual Studio
    - Bindings & Triggers and their Types
    - Implementing Various Triggers
    - Accessing Functions using Keys
    - Durable Functions
    - Serverless Comparison
    - Hosting Plan

- Azure Service Bus
    - Service Bus Messaging
    - Queues, Topics, and Subscriptions
    - Create Service Bus Queue
    - Create Service Bus Topics & Subscriptions
    - Send and Receive Messages – Queues
    - Accessing Service Bus using C#.NET Azure SDK
    - Handle Service Bus Events via Event Grid
    - Publish and Subscribe for Messages
    - Monitor Service Bus using Azure Monitor
- Azure Logic App
    - Introduction to Logic App
    - Single-tenant versus Multi-Tenant
    - Creating Logic App
    - Schedule Based Workflows
    - Approval Based Workflows
    - Creating Azure Storage and Azure Function Work Flow
    - Deploy Logic Apps
        - Automated Logic App Deployment
- Azure Data Factory
    - Pipelines and activities in Azure Data Factory
    - Azure Data Factory UI
    - Create Azure Data Factory UI
    - Create Data Flow
    - Copy and Ingest Data
    - Transform Data with Mapping Data Flows
    - ETL Mapping
- Overview of Azure Analysis Service
    - Create Server-Portal
    - Adding Model, Configure Server Administrator and User Roles
    - Authentication and User Permissions
    - Service Principles
- Azure Database Services & Azure SQL
    - Database Workloads in Azure
    - OLAP in Azure Introduction
    - Non-Relational Databases in Azure
    - Azure SQL Introduction
    - Migrate to Azure SQL
    - SQL Managed Instances
    - Elastic Pools
    - Instance Pools
    - Create SQL Database
    - Configure Firewall
    - Configuring Security
    - Logins, user accounts, roles, and permissions
- Cosmos DB
    - Introduction to Azure Cosmos DB
    - NoSQL Vs Relational Databases
    - Cosmos DB Resource Model
    - Global Distribution

    - Partitioning and horizontal scaling
    - Create an Azure Cosmos Account
    - Build a .NET Web APP to Manage Data
    - Query Data with SQL Queries
    - Introduction to the Azure Cosmos DB Cassandra API
    - Types of Consistencies in Cosmos DB
- Containers, ACR & ACI
    - Introduction to Containerization of Web Apps
    - Comparing On-Premises, Virtual Machines & Containerized deployments
    - Introduction to ACI & Features
    - Docker Hub Vs ACR
    - Comparing Various Options to Deploy Containerized Apps in Azure (Highlight benefits & Limitation)
        - Azure App Service
        - Azure Container Instances
        - Azure Kubernates Service
        - Azure Functions
    - Create ASP.NET Core Web Application with Docker Support (MVC / Web API)
    - Creating Tags for Versioning App
    - Pushing Containerized App on Azure Container Registry
    - Create ACI Service Instance
    - Deploy App on ACI by pulling image from ACR
    - Environments, Containers & Revisions
    - Application LifeCycle Management
- Ways to authenticate App Services in Azure