* **Day 1: C# - Basic OO programming, Collections and Generics**
* **Module 1: •NET Framework**
  + **.NET Framework and features**
    - **CLR,CLS,BCL**
  + **Assembly**
    - **Types**
  + **JIT**
* **Module 2: Objects and Types**
  + **Topics**
  + Creating and Using Classes
  + Classes and Structs
  + Classes
  + Anonymous Types
  + Structs
  + Partial Classes
  + Static Classes
  + The Object Class
  + Extension Methods
* **Module 3: Arrays & Collection**
  + Overview of Arrays
  + Creating and using Arrays
  + Understanding NET collections
  + Ordered vs Unordered Collections
  + Collection classes – CollectionBase, ArrayList, HashTable, Dictionary, Queue, Stack, LinkedList, SortedList, Generic Collection
  + Collection Interfaces: IList, IEnumerable, IEnumerator, IDictionary
  + Specialized Collections like CollectionsUtil, NameValueCollection,
* **Module 4: Generics**
  + **Topics**
  + Generics Overview
  + Creating Generic Classes
  + Generics Features
  + Generic Interfaces
  + Generic Method
* **Day 2**
* **Module 5: Delegates, Lambdas and Events**
  + - **Topics**
    - Referencing Methods
    - Delegates
    - Lambda Expressions
    - Events
* **Module 5: Language Integrated Query (LINQ)**
  + **Topics**
  + LINQ Overview
  + Standard Query Operators
  + Parallel LINQ
  + LINQ Providers
* **Day 3: ADO.NET**
  + **Data-Centric Applications and ADO.NET**
    - Design of Data-Centric Applications
    - ADO.NET Architecture
    - ADO.NET and XML
    - Hands-On Practice Exercise: Data-Centric Applications and ADO.NET
  + **Connecting to Data Sources**
    - Choosing a .NET Data Provider
    - Defining and Managing a Connection
    - Handling Connection Exceptions
    - Hands-On Practice Exercise: Connecting to Data Sources
  + **Performing Connected Database Operations**
    - Working in a Connected Environment
    - Building Command Objects
    - Executing commands that return a Single Value, Rows or no return
    - Using Transactions
    - Hands-On Practice Exercise: Performing Connected Database Operations
  + **Building Datasets from Existing Data Sources**
    - Configuring a DataAdapter to Retrieve Information
    - Populating a DataSet Using a DataAdapter
    - Configuring a DataAdapter to Update the Underlying Data Source
    - Persisting Changes to a Data Source
    - How to Handle Conflicts
    - Hands-On Practice Exercise: Retrieving Data into a Disconnected Application
* **Day 4: Web API**
  + **Introduction to ASP.NET Web API**
    - **ASP.NET Web API Architecture**
    - **Architectural Features**
    - **ASP.NET Web API Elements**
    - **HTTP Processing Pipeline**
    - **Code request and response will be verified through fiddler**
  + **Routing**
    - **Understanding Routing Mechanism**
    - **Defining Web API Routes**
    - **HTTP Methods and Routing**
    - **Having Multiple Web API Routes**
    - **Route Defaults**
    - **Optional Parameters**
    - **Route Constraints**
  + **Controllers and Actions**
    - **Overview**
    - **Overview of HttpRoutingDispatcher**
    - **Controller Type Selection**
    - **Creating a Controller**
    - **IHttpController Interface and Your Own Controller**
    - **ApiControllers: Out of the Box IHttpController Implementation**
    - **Controller Actions**
    - **Action Selection**
    - **Return Values**
    - **Per-Controller-Type Configuration**
  + **Media Type Formatters and Model Binding**
    - **Overview**
    - **Formatter Processing Model**
    - **Content Negotiation Algorithm**
    - **Media Type Formatters**
    - **Default Formatters**
    - **"JsonMediaTypeFormatter XMLMediaTypeFormatter**
    - **FormUrlEncodedMediaTypeFormatter"**
    - **Model Binding**
    - **Model Binder Mechanism**
    - **Default Model Binder**
    - **Customizing Model Binding**
  + **Input Validation**
    - **Overview**
    - **How Validation Works Through Parameter Binding**
* **Assessment for CSharp & ADO.NET**
* **Day 5: Minimal JS to render the results of the Web API**

## Introduction to JavaScript

* + What JavaScript Is?
  + What JavaScript Is Not?
  + What JavaScript Is Used for?
  + JavaScript and Its Place in a Web Page.

## JavaScript Building Blocks: Data Types, Literals, and Variables

* + Data Types
  + Primitive Data Types
  + Composite Data Types
  + Variables
  + Valid Names
  + Declaring and Initializing Variables
  + Dynamically or Loosely Typed Language
  + Scope of Variables
  + Concatenation and Variables

## Functions

* + What Is a Function?
  + Function Declaration and Invocation
  + Return Values
  + Anonymous Functions as Variables
  + Closures
  + Recursion
  + Functions Are Objects

## Objects

* + What Are Objects?
  + Objects and the Dot Syntax
  + Creating an Object with a Constructor
  + Properties of the Object
  + Methods of the Object

## Classes and User-Defined Functions

* + What Is a Class?
  + What Is this?
  + Inline Functions as Methods
  + Object Literals

## jQuery Fundamentals

* + Introduction - Why use jQuery?
  + Referencing a jQuery Script
  + Using Content Delivery Networks
  + Using the jQuery Ready Function
  + Getting to Know the jQuery Documentation

## Using jQuery Selectors

* + What are Selectors?
  + Selecting Nodes by Tag Name
  + Selecting Nodes by ID
  + Selecting Nodes by Class Name
  + Selecting Nodes by Attribute Value
  + Selecting Input Nodes
  + Additional Selector Features

## Interacting with the DOM

* + Introduction
  + Iterating Through Nodes
  + Modifying Properties and Attributes
  + Adding and Removing Nodes
  + Modifying Styles
  + Modifying Classes

## Handling Events

* + Introduction
  + jQuery Event Model Benefits
  + Handling Events
  + Click Event
  + Change Event
  + Mouse Events
  + Binding to Events
  + Handling Hover Events

## Working with Ajax Features

* + Introduction
  + jQuery Ajax Functions
  + Loading HTML Content from the Server
  + load() Function Demo
  + Making GET Requests
  + Making POST Requests
  + Introduction to the ajax() Function
* **Project Introduction**
* **Day 6: HTML 5**

## Introduction to HTML 5

* + - Introduction to HTML, CSS and JavaScript
    - Semantics
    - Demo
    - New Semantic elements

## HTML5 Canvas

* + - Introduction to Canvas tag
    - Basic Canvas Drawing
    - Complex drawing elements with Canvas
    - Transforming objects in Canvas using Translate tag
    - Canvas Text API
    - Images with Canvas

## Local Storage

* + - Types of Storage
    - Checking for HTML5 storage support
    - Declaring manifest for your page
    - ApplicationCache object
    - Deleting local cache
* **Project work**
* **Day 7: Bootstrap**
  + **Introduction**
    - **What is a front-end framework**
    - **Bootstrap intro**
    - **Why Bootstrap**
  + **Designing for multiple devices**
    - **Responsive Web Design**
    - **Mobile First Responsive Design**
  + **Getting started with Bootstrap**
  + **Downloading Bootstrap**
    - **Different ways to add Bootstrap to your page**
  + **Container**
    - **Bootstrap Grid System**
    - **Multiple grids**
    - **Rows and offsets**
    - **Responsive images, videos, maps**
    - **Bootstrap style vs your style**
    - **Bootstrap themes**
  + **Bootstrap elements and components**
    - **Icons**
    - **Buttons**
    - **Typography**
    - **Navigations**
    - **Tables**
    - **Lists**
    - **Forms**
    - **Headers**
    - **Breadcrumbs**
    - **Pagination**
    - **Dropdowns**
    - **Panels**
  + **Bootstrap Plug-ins**
    - **Carousel**
    - **Accordion**
    - **Modal window**
    - **Tab**
    - **Tooltip**
    - **Collapse**
* **Project work**
* **Day 8: Angular 8**
* Module 1. Introducing Angular
  + What is Angular?
  + Central Features of the Angular Framework
  + Why Angular?
  + Scope and Goal of Angular
  + Angular vs. AngularJS
  + But Isn't It "Angular 2"?
  + Installing and Using Angular
  + Adding Angular and Dependencies to Your App
  + Building Blocks of an Angular Application
  + A Basic Angular Application
  + Basic App - index.html
  + Basic-App: Application Module File
  + Basic-App: Main Bootstrap File
  + Basic-App: The Component File
  + Summary
* Module 2. Development Setup of Angular
  + Angular is Modular
  + Managing Angular Files and Dependencies
  + What is Node.js?
  + Application of Node.js
  + Installing Node.js and NPM
  + "Hello World!" Node app
  + Node Libraries
  + Node Package Manager (NPM)
  + Package.json
  + Semantic Version Numbering
  + Package Version Numbering Syntax
  + Updating Packages
  + Uninstalling Packages
  + Installing Angular Packages
  + Angular CLI
  + Angular Development Overview
  + Angular
  + Development
  + Dependencies
  + TypeScript Definitions
  + Testing Tools
  + Development Servers
  + Angular
  + Application
  + Dependencies
  + Module Loaders
  + SystemJS Module Loader
  + WebPack Module Bundler
  + Additional Application Dependencies
  + Summary
* Module 3. Introduction to TypeScript and ES6
  + Programming Languages for Use with Angular
  + TypeScript Syntax
  + Programming Editors
  + The Type System – Defining Variables
  + The Type System – Defining Arrays
  + The Type System – Classes & Objects
  + Class Constructors
  + Class Constructors – Alternate Form
  + Interfaces
  + Parameter and Return Value Types
  + Working with Modules
  + TypeScript Transpilation
  + var, let
  + and const - Defined
  + var, let and const - Usage
  + Arrow Functions
  + Template Strings
  + Template Strings – Variables and Expressions
  + Template Strings – Multiline
  + Generics - Class
  + Generics - Methods
  + Generics - Restricting Types
  + Generics - Restricting Types: Example
  + Summary
* **Project work**
* **Day 9: Angular5…**
* Module 4. Components in Angular
  + What is a Component?
  + An Example Component
  + Component Starter
  + Developing a Simple Login Component
  + Login Component: Add HTML
  + The HTML Component Template
  + The templateUrl property
  + Login Component: Add CSS Styling
  + Login Component: Hook Up Input Fields and Button
  + Login Component: Fields & Button in the Component Class
  + Component Decorator Properties
  + Component Lifecycle Hooks
  + Using a Lifecycle Hook:
  + OnInit
  + Summary
* Module 5. Data and Event Binding
  + Binding Syntax
  + One-Way Output Binding
  + Binding Displayed Output Values
  + Two-Way Binding of Input Fields
  + Input Binding Examples
  + Binding Events
  + Binding Events Examples
  + Setting Element Properties
  + Setting Properties: Examples
  + Passing Data into Components using @Input()
  + Passing Data from Child to Parent using @Output()
  + @Output() Example - Child Component
  + @Output() Example - Parent Component
  + Summary
* Module 6. Attribute Directives and Property Bindings
  + What are Directives
  + Directive Types
  + Apply Styles by Changing Classes
  + Changing Classes – Example
  + Applying Styles Directly
  + Applying Styles Directly - Example
  + Obsolete Directives and Property Binding
  + Controlling Element Visibility
  + Setting Image Source Dynamically
  + Setting Hyperlink Source Dynamically
  + Summary
* Module 7. Structural Directives
  + Structural Directives
  + Adding and Removing Elements Dynamically
  + Looping Using ngFor
  + ngFor - Basic Syntax
  + ngFor - Full Template Syntax
  + Creating Tables with ngFor
  + ngFor Local Variables
  + ngFor Changes in the backing data source
  + Swapping Elements with ngSwitch
  + ngSwitch - Basic Syntax
  + ngSwitch - Full Template Syntax
  + Summary
* **Project work**
* **Day 10: Angular5…**
* Module 8. Template Driven Forms
  + Template Driven Forms
  + Importing Forms Module
  + A Basic Angular Form
  + Binding Input Fields
  + Accessing the Form Object
  + Binding the Form Submit Event
  + The Submit Function
  + Basic HTML5 Validation - "required" Attribute
  + HTML5 vs. Angular Validation
  + Angular Validators
  + Angular Validation State
  + Displaying
  + Form
  + Validation State
  + Displaying
  + Field
  + Validation State
  + Displaying Validation State Using Classes
  + Disabling Submit when Form is Invalid
  + Submitting the Form
  + Binding to Object Variables
  + Additional Input Types
  + Checkboxes
  + Select (Drop Down) Fields
  + Rendering Options for Select (Drop Down)
  + Date fields
  + Radio Buttons
  + Summary
* Module 9. Model Driven Forms
  + Model Driven Forms Overview
  + Setup for Model Driven Forms
  + Form Component Setup
  + Setup Main FormGroup
  + formControlName
  + FormControl Object
  + Getting Form Values
  + FormBuilder Form Initialization
  + Validation
  + Built-In Validators
  + Custom Validator
  + Using a Custom Validator
  + Useful FormGroup and FormControl Properties/Functions
  + Sub FormGroups - Component Class
  + Sub FormGroups - HTML Template
  + Why Use Sub FormGroups
  + Summary
* **Project work**
* **Day 11: Angular5…**
* Module 10. Angular Modules
  + Why Angular Modules?
  + Angular Built-in Modules
  + The Root Module
  + How to Create a Module
  + Feature Modules
  + @NgModule Properties
  + Using One Module From Another
  + Importing BrowserModule or CommonModule
  + Lazy-Loaded Modules
  + How to Organize Modules?
  + Component moduleID Property
  + Summary
* Module 11. Services and Dependency Injection
  + What is a Service?
  + Creating a Basic Service
  + What is Dependency Injection?
  + What Dependency Injection Looks Like
  + Injecting Services
  + Dependency Injection Hierarchy Diagram
  + Using a Service in a Component: Dedicated Instance
  + Injection Hierarchy - Dedicated Instance
  + Using a Service in a Component: Dedicated Instance - Example Code
  + Using
  + onInit
  + to Initialize Component Data
  + Using a Shared Service Instance
  + Injection Hierarchy - Shared Instance
  + Dependency Injection and @Host
  + Dependency Injection and @Optional
  + Summary
* Module 12. HTTP Client
  + The Angular HTTP Client
  + Using The HTTP Client - Overview
  + Importing HttpModule
  + Importing Individual Providers into Services
  + Service Using Http Client
  + Service Imports
  + The Observable object type
  + What does an Observable Object do?
  + Making a Basic HTTP GET Call
  + Using the Service in a Component
  + The PeopleService Client Component
  + Client Component Code Review
  + Importing Observable Methods
  + Enhancing the Service with .map() and .catch()
  + Using .map()
  + Using .catch()
  + Using toPromise()
  + GET Request
  + GET Request with Options
  + POST Request
  + POST Request Example
  + Reading HTTP Response Headers
* Summary
* **Project work**
* **Day 12: Application Development Assessment**
* Module 13. Pipes and Data Formatting
  + What are Pipes?
  + More on Pipes
  + Formatting Changes in Angular
  + Using a Built-in Pipe
  + Built-In Pipes
  + Using Pipes in HTML
  + Chaining Pipes
  + Using Pipes in JavaScript
  + Some Pipe Examples
  + Decimal Pipe
  + CurrencyPipe
  + Custom Pipes
  + Custom Pipe Example
  + Using Custom Pipes
  + A Filter Pipe
  + A Sort Pipe
  + Pipe Category: Pure and Impure
  + Pure Pipe Example
  + Impure Pipe Example
  + Summary
* Module 15. The Angular Component Router
  + Routing and Navigation
  + The Component Router
  + Traditional Browser Navigation
  + Component Router Terminology
  + Setting up the Component Router
  + Local URL Links
  + Browser pushState and <base href>
  + Routes
    - The app.routes.ts File
    - The app.routes.ts File - Example
    - Importing Routing in app.module.ts
    - A Basic App With Routing
    - App Routes
    - AppComponent - Code
    - AppComponent - Router Related Features
    - AppComponent - <router-outlet>
    - AppComponent - routerLinks
    - Programmatic Navigation
    - Basic Navigation
    - Passing Data During Navigation
    - Creating Routes with Route Parameters
    - Navigating with Route Parameters
    - Using Route Parameter Values
    - Retrieving the Route Parameter Synchronously
    - Retrieving a Route Parameter Asynchronously
    - Query Parameters
    - Query Parameters - Example Component
    - Query Parameters - queryParams
    - Query Parameters - Navigation
    - Retrieving Query Parameters Asynchronously
    - Problems with Manual URL entry and Bookmarking
    - Fixing Manual URL entry and Bookmarking
    - Summary
* **Project Work completion and Evaluation**