# bet365 .NET Training

1. C# Programming
2. Installing, Configuring and Managing Microsoft SQL Server Microsoft SQL Server 2014
3. Program ASP.NET MVC

The training is basedon the following official Microsoft courses.

• Programming in C# and .Net

<https://www.microsoft.com/en-ca/learning/course.aspx?cid=20483>

• Developing Data Access Solutions with Microsoft® Visual Studio® 2010

<https://www.microsoft.com/en-ca/learning/course.aspx?cid=10265>

• Windows® Communication Foundation Solutions

<https://www.microsoft.com/en-ca/learning/course.aspx?cid=10263>

• Programming in HTML5 with JavaScript and CSS3

<https://www.microsoft.com/en-ca/learning/course.aspx?cid=20480>

• Developing ASP.NET MVC 4 Web Applications

<https://www.microsoft.com/en-ca/learning/course.aspx?cid=20486>

• Querying Microsoft SQL Server 2012/2014

<https://www.microsoft.com/en-us/learning/course.aspx?cid=20461>

• Developing Microsoft SQL Server 2014 Databases

<https://www.microsoft.com/en-us/learning/course.aspx?cid=20464>

# 1. C# Programming

1. Introduction to C#

2. Primitive data types and variables

3. Operators in C#

4. Conditional statements

5. Loops

6. Arrays

7. Multidimensional arrays

8. Methods (Functions)

9. Classes object and structures

10. Constructors, getters, setters and fields

11. Access modifiers

12. Generics, enumerations and attributes

13. Interfaces and polymorphism

14. SOLID principles

15. Exceptions in C#

16. Files input and output

17. Lambdas, extension methods, delegates, LINQ

**C# + .Net**

Would it be possible to also include coverage of the following topic areas:

* Review of C# Syntax, Working with Visual Studio
* Creating Methods, Handling Exceptions, and Monitoring Apps
* Creating Classes and Implementing Type-safe Collections
* Creating a Class Hierarchy by Using Inheritance
* Reading and Writing Local Data
* Improving Application Performance and Responsiveness
* Creating Reusable Types and Assemblies
* Encrypting and Decrypting Data
* ReSharper and Unit Testing,
* Dependency Injection, IoC and Moq

### Review of C# Syntax, Working with Visual Studio

This reviews the core syntax and features of the C# programming language. It also provides an introduction to the Visual Studio 2012 debugger.

**Lessons**

• Overview of Writing Applications using C#

• Visual Studio Overview

• Datatypes, Operators, and Expressions

• C# Programming Language Constructs

**Lab: Developing the Class Enrolment Application**

After completing this, students will be able to:

• Describe the architecture of .NET Framework applications and use the features that Visual Studio 2012 and C# provide to support .NET Framework development.

• Use the basic data types, operators, and expressions provided by C#.

• Use standard C# programming constructs.

### Creating Methods, Handling Exceptions, and Monitoring Apps

This explains how to create and call methods, catch and handle exceptions. This also describes the monitoring requirements of large-scale applications.

**Lessons**

• Creating and Invoking Methods

• Creating Overloaded Methods and Using Optional and Output Parameters

• Handling Exceptions

• Monitoring Applications

**Lab: Extending the Class Enrolment Application Functionality**

After completing this, students will be able to:

• Create and invoke methods, pass parameters to methods, and return values from methods.

• Create overloaded methods, and use optional parameters and output parameters.

• Catch and handle exceptions and write information to the event log.

• Explain the requirement for implementing logging, tracing, and profiling when building large-scale applications.

### Creating Classes and Implementing Type-safe Collections

This explains how to create classes, define and implement interfaces, and create and use generic collections. This also describes the differences between value types and reference types in C#.

**Lessons**

• Creating Classes

• Defining and Implementing Interfaces

• Implementing Type-safe Collections

**Lab: Adding Data Validation and Type-safety to the Grades Application**

After completing this, students will be able to:

• Create and use custom classes.

• Define and implement custom interfaces.

• Use generics to implement type-safe collections.

### Creating a Class Hierarchy by Using Inheritance

This explains how to use inheritance to create a class hierarchy and extend a .NET Framework class. This also describes how to create generic classes and define extension methods.

**Lessons**

• Creating Class Hierarchies

• Extending .NET Framework Classes

• Creating Generic Types

**Lab: Refactoring Common Functionality into the User Class**

After completing this, students will be able to:

• Define abstract classes and inherit from base classes to create a class hierarchy.

• Inherit from .NET Framework classes and use extension methods to add custom functionality to the inherited class.

• Create generic classes and methods.

### Reading and Writing Local Data

This explains how to read and write data by using file input/output (I/O) and streams, and how to serialize and deserialize data in different formats.

**Lessons**

• Reading and Writing Files

• Serializing and Deserializing Data

• Performing I/O Using Streams

**Lab: Generating the Grades Report**

After completing this, students will be able to:

• Read and write data to and from the file system by using file I/O.

• Convert data into a format that can be written to or read from a file or other data source.

• Use streams to send and receive data to or from a file or other data source.

### Improving Application Performance and Responsiveness

This explains how to improve the throughput and response time of applications by using tasks and asynchronous operations.

**Lessons**

• Implementing Multitasking by using Tasks and Lambda Expressions

• Performing Operations Asynchronously

• Synchronizing Concurrent Access to Data

**Lab: Improving the Responsiveness and Performance of the Application**

After completing this, students will be able to:

• Create tasks and lambda expressions to implement multitasking.

• Define and use asynchronous methods to improve application responsiveness.

• Coordinate concurrent access to data shared across multiple tasks by using synchronous primitives and concurrent collections.

### Creating Reusable Types and Assemblies

This explains how to examine the metadata of types by using reflection, create and use custom attributes, generate managed code at runtime, and manage different versions of assemblies.

**Lessons**

• Examining Object Metadata

• Creating and Using Custom Attributes

• Generating Managed Code

• Versioning, Signing and Deploying Assemblies

**Lab: Specifying the Data to Include in the Grades Report**

After completing this, students will be able to:

• Examine the metadata of objects at runtime by using reflection.

• Create and use custom attribute class.

• Generate managed code at runtime by using CodeDOM.

• Manage different versions of an assembly and deploy an assembly to the Global Assembly Cache.

### Encrypting and Decrypting Data

This explains how to encrypt and decrypt data by using symmetric and asymmetric encryption.

**Lessons**

• Implementing Symmetric Encryption

• Implementing Asymmetric Encryption

**Lab: Encrypting and Decrypting Grades Reports**

After completing this, students will be able to:

• Perform symmetric encryption by using the classes in the System.Security namespace.

• Perform asymmetric encryption by using the classes in the System.Security namespace.

### ReSharper and Unit Testing

This lesson explains the benefits of using tools to assist in software development.

**Lessons**

• Using ReSharper to refactor code, when to accept it’s suggestions and when not too

• The benefits of Unit Testing code and assessing code coverage

**Lab: ReSharper and Unit Testing**

After completing this, students will be able to:

• Refactor some code, understanding the nature of the suggestions and the impact of refactoring on a legacy code base.

• Write unit tests for their code and understand how code should be structured to enable ease of testing.

### Dependency Injection, IoC and Moq

This lesson explains the principal of Dependency Injection, Inversion of Control and Moqing frameworks.

**Lessons**

• Understand the principal of Dependency Injection

• Understand how to use Inversion of Control and containers.

• Understand Moquing frameworks and how to use Moq classes in testing.

**Lab: Dependency Injection, IoC and Moq**

After completing this, students will be able to:

• Test their code using Moq classes and an IoC container.

## **2. Installing, Configuring and Managing Microsoft SQL Server Microsoft SQL Server** 2014

**(8 days)**

1: Introduction to SQL Server 2014

Lessons

* Overview of SQL Server 2014
* SQL Server 2014 Editions and Components
* Installing SQL Server 2014
* SQL Server Management Studio Enhancements

Lab: Installing SQL Server 2014

2: What’s New in SQL Server PerformanceLessons:

* Columnstore Indexes for Operational Analytics
* In-memory OLTP Enhancements
* The Query Data Store
* Native JSON
* Temporal Tables

Lab: Implementing SQL Server 2014 Performance Improvements

3: What’s New in SQL Server Security?

* Using Always Encrypted
* Row-level Security
* Dynamic Data Masking

Lab: Implementing SQL Server 2014 Security Improvements

4: What’s New in SQL Server Availability and Scalability?

* Enhanced AlwaysOn Availability Groups
* What’s New in Database Caching?
* Using Windows Server 2014

Lab: Availability and Scalability with SQL Server 2014

5: Disaster Recovery

* Mirroring
* Transactional Replication
* Log Shipping
* AlwaysOn

Lab: Disaster Recovery

~~6: What’s New for SQL Server in the Cloud?~~

~~Lessons:~~

* ~~Stretch Database~~
* ~~Enhanced Backups to Azure~~
* ~~Introduction to SQL Server 2016 Hybrid Cloud~~
* ~~What’s New in SQL Server Azure Database~~

~~Lab: Using SQL Server in a Hybrid Cloud~~

**SQL Server**

Would it be possible to also include coverage of the following topic areas:

* Introduction to T-SQL Querying
* Writing SELECT Queries
* Querying Multiple Tables
* Sorting and Filtering Data
* Using DML to Modify Data
* Using Built-In Functions
* Grouping and Aggregating Data
* Using Subqueries
* Using Table Expressions
* Using Set Operators
* Querying data with Stored Procedures
* Programming with T-SQL
* Implementing Error Handling
* Implementing Transactions
* Improving Query Performance
* Designing and Implementing Tables and Indexes
* Designing Optimized Index Strategies
* Designing Views, Stored Procedures, UDF
* Responding to Data Manipulation via Triggers

### Introduction to T-SQL Querying

This introduces Transact SQL as the primary querying language of SQL Server. It discusses the basic structure of T-SQL queries, the logical flow of a SELECT statement, and introduces concepts such as predicates and set-based operations.

**Lessons**

• Introducing T-SQL

• Understanding Sets

• Understanding Predicate Logic

• Understanding the Logical Order of Operations in SELECT statements

**Lab: Introduction to Transact-SQL Querying**

After completing this, you will be able to:

• Describe the elements of T-SQL and their role in writing queries

• Describe the use of sets in SQL Server

• Describe the use of predicate logic in SQL Server

• Describe the logical order of operations in SELECT statements

### Writing SELECT Queries

This introduces the fundamentals of the SELECT statement, focusing on queries against a single table.

**Lessons**

• Writing Simple SELECT Statements

• Eliminate Duplicates with DISTINCT

• Using Column and Table Aliases

• Write Simple CASE Expressions

**Lab: Writing Basic SELECT Statements**

After completing this, you will be able to:

• Write simple SELECT statements.

• Eliminate duplicates using the DISTINCT clause.

• Use column and table aliases.

• Write simple CASE expressions.

### Querying Multiple Tables

This explains how to write queries which combine data from multiple sources in SQL Server. This introduces the use of JOINs in T-SQL queries as a mechanism for retrieving data from multiple tables.

**Lessons**

• Understanding Joins

• Querying with Inner Joins

• Querying with Outer Joins

• Querying with Cross Joins and Self Joins

**Lab: Querying Multiple Tables**

After completing this, you will be able to:

• Describe how multiple tables may be queried in a SELECT statement using joins.

• Write queries that use inner joins.

• Write queries that use outer joins.

• Write queries that use self-joins and cross joins.

### Sorting and Filtering Data

This explains how to enhance queries to limit the rows they return, and to control the order in which the rows are displayed. The also discusses how to resolve missing and unknown results.

**Lessons**

• Sorting Data

• Filtering Data with Predicates

• Filtering with the TOP and OFFSET-FETCH

• Working with Unknown Values

**Lab: Sorting and Filtering Data**

After completing this, you will be able to:

• Filter data with predicates in the WHERE clause.

• Sort data using ORDER BY.

• Filter data in the SELECT clause with TOP.

• Filter data with OFFSET and FETCH.

### Using DML to Modify Data

This describes the use of Transact-SQL Data Manipulation Language to perform inserts, updates, and deletes to your data.

**Lessons**

• Inserting Data

• Modifying and Deleting Data

**Lab: Using DML to Modify Data**

After completing this, you will be able to:

• Insert new data into your tables.

• Update and delete existing records in your tables.

### Using Built-In Functions

This introduces the use of functions that are built in to SQL Server Denali, and will discuss some common usages including data type conversion, testing for logical results and nullability.

**Lessons**

• Writing Queries with Built-In Functions

• Using Conversion Functions

• Using Logical Functions

• Using Functions to Work with NULL

**Lab: Using Built-In Functions**

After completing this, you will be able to:

• Write queries with built-in scalar functions.

• Use conversion functions.

• Use logical functions.

• Use functions that work with NULL.

### Grouping and Aggregating Data

This introduces methods for grouping data within a query, aggregating the grouped data and filtering groups with HAVING. The is designed to help the student grasp why a SELECT clause has restrictions placed upon column naming in the GROUP BY clause as well as which columns may be listed in the SELECT clause.

**Lessons**

• Using Aggregate Functions

• Using the GROUP BY Clause

• Filtering Groups with HAVING

**Lab: Grouping and Aggregating Data**

After completing this , you will be able to:

• Write queries which summarize data using built-in aggregate functions.

• Use the GROUP BY clause to arrange rows into groups.

• Use the HAVING clause to filter out groups based on a search condition.

### Using Subqueries

This will introduce the use of subqueries in various parts of a SELECT statement. It will include the use of scalar and multi-result subqueries and the use of the IN and EXISTS operators.

**Lessons**

• Writing Self-Contained Subqueries

• Writing Correlated Subqueries

• Using the EXISTS Predicate with Subqueries

**Lab: Using Subqueries**

After completing this, you will be able to:

• Describe the uses of queries which are nested within other queries.

• Write self-contained subqueries which return scalar or multi-valued results.

• Write correlated subqueries which return scalar or multi-valued results.

• Use the EXISTS predicate to efficiently check for the existence of rows in a subquery.

### Using Table Expressions

This introduces T-SQL expressions which return a valid relational table, typically for further use in the query. This discusses views, derived tables, common table expressions and inline table-valued functions.

**Lessons**

• Using Views

• Using Inline Table-Valued Functions

• Using Derived Tables

• Using Common Table Expressions

**Lab: Using Table Expressions**

After completing this, you will be able to:

• Write queries which use derived tables.

• Write queries which use common table expressions.

• Create simple views and write queries against them.

• Create simple inline table-valued functions and write queries against them.

### Using Set Operators

This introduces the set operators UNION, INTERSECT, and EXCEPT to compare rows between two input sets

**Lessons**

• Writing Queries with the UNION Operator

• Using EXCEPT and INTERSECT

• Using APPLY

**Lab: Using Set Operators**

After completing this, you will be able to:

• Write queries which combine data using the UNION operator

• Write queries which compare sets using the INTERSECT and EXCEPT operators

• Write queries which manipulate rows in a table by using APPLY with the results of a derived table or function

### Querying data with Stored Procedures

This introduces the use of existing stored procedures in a T-SQL querying environment. It discusses the use of EXECUTE, how to pass input and output parameters to a procedure, and how to invoke system stored procedures.

**Lessons**

• Writing Queries with PIVOT and UNPIVOT

• Passing Parameters to Stored Procedures

• Creating Simple Stored Procedures

• Working with Dynamic SQL

**Lab: Executing Stored Procedures**

After completing this, you will be able to:

• Return results by executing stored procedures.

• Pass parameters to procedures.

• Create simple stored procedures which encapsulate a SELECT statement.

• Construct and execute dynamic SQL with EXEC and sp\_executesql.

### Programming with T-SQL

This provides a basic introduction to T-SQL programming concepts and objects. It discusses batches, variables, control of flow elements such as loops and conditionals, how to create and execute dynamic SQL statements, and how to use synonyms.

**Lessons**

• T-SQL Programming Elements

• Controlling Program Flow

**Lab: Programming with T-SQL**

After completing this, you will be able to:

• Describe the language elements of T-SQL used for simple programming tasks.

• Describe batches and how they are handled by SQL Server.

• Declare and assign variables and synonyms.

• Use IF and WHILE blocks to control program flow.

### Implementing Error Handling

This introduces the use of error handlers in T-SQL code. It will introduce the difference between compile errors and run-time errors, and will cover how errors affect batches. The will also cover how to control error handling using TRY/CATCH blocks, the use of the ERROR class of functions, and the use of the new THROW statement.

**Lessons**

• Using TRY / CATCH Blocks

• Working with Error Information

**Lab: Implementing Error Handling**

After completing this, you will be able to:

• Describe SQL Server's behavior when errors occur in T-SQL code.

• Implement structured exception handling in T-SQL.

• Return information about errors from system objects.

• Raise user-defined errors and pass system errors in T-SQL code.

### Implementing Transactions

This introduces the concepts of transaction management in SQL Server. It will provide a high-level overview of transaction properties, cover the basics of marking transactions with BEGIN, COMMIT and ROLLBACK.

**Lessons**

• Transactions and the Database Engine

• Controlling Transactions

**Lab: Implementing Transactions**

After completing this, you will be able to:

• Describe transactions and the differences between batches and transactions.

• Describe batches and how they are handled by SQL Server.

• Create and manage transactions with transaction control language statements.

• Use SET XACT\_ABORT to define SQL Server's handling of transactions outside TRY / CATCH blocks.

• Describe the effects of isolation levels on transactions.

### Improving Query Performance

This presents several key guidelines for writing well-performing queries, as well as ways to monitor the execution of your queries and their impact on Microsoft SQL Server

**Lessons**

• Factors in Query Performance

• Displaying Query Performance Data

• Execution plans

• Indexes – Clustered and Non-Clustered

**Lab: Improving Query Performance**

After completing this, you will be able to:

• Describe components of well-performing queries.

• Display and interpret basic query performance data

### Designing and Implementing Tables and Indexes

This explains creating tables, indexes, constraints

**Lessons**

• Designing Tables

• Data Types

• Working with Schemas

• Creating and Altering Tables

• Partitioning Data

• Compressing Data

• Enforcing Data Integrity

• Implementing Domain Integrity

• Implementing Entity and Referential Integrity

• Core Indexing Concepts

• Data Types and Indexes

• Single Column and Composite Indexes

**Lab:**

• Design Tables.

• Work with Schemas.

• Create and Alter Tables.

• Implement entity and referential integrity.

• Describe core indexing concepts.

• Choose appropriate data types for indexes.

• Design and implement clustered and nonclustered indexes.

### Designing Optimized Index Strategies

This explains covering indexes and the INCLUDE clause as well as the use of padding, hints and statistics. This also covers the use of the Database Engine Tuning Advisor and index-related dynamic management views to assess indexing strategies.

**Lessons**

• Covering Indexes

• Managing Indexes

• Working with Execution Plans

• Using the DTE

**Lab:**

After completing this, you will be able to:

• Describe the elements of an execution plan.

• Design effective indexing strategies.

• Monitor your system to assess the performance of your indexing strategy.

### Designing Views, Stored Procedures, UDF

**Lessons**

• Introduction to Views

• Creating and Managing Views

• Performance Considerations for Views

• Introduction to Stored Procedures

• Working With Stored Procedures

• Implementing Parameterized Stored Procedures

• Controlling Execution Context

• Overview of Functions

• Designing and Implementing Scalar Functions

• Designing and Implementing Table-Valued Functions

• Implementation Considerations for Functions

• Alternatives to Functions

**Labs:**

• Creating

O Views

O Stored Procedures

O UDF

### Responding to Data Manipulation via Triggers

This explains what DML triggers are and how they enforce data integrity. Also it focusses on the different types of triggers available, and how to define triggers in a database.

**Lessons**

• Designing DML Triggers

• Implementing DML Triggers

• Advanced Trigger Concepts

**Lab:**

After completing this, you will be able to:

• Design DML triggers.

• Implement DML triggers.

• Explain advanced DML trigger concepts.

### Architecture and Data Access Technologies

This describes the commonly used data access technologies and scenarios in which you are likely to use them.

**Lessons**

• Data Access Technologies

• Data Access Scenarios

**Lab: Analyzing Data Access Scenarios**

After completing this, students will be able to:

• Describe the key data access technologies available to .NET Framework developers.

• Assign appropriate data access technologies to common data access scenarios.

### Using ADO.NET

ADO.NET is a highly flexible framework for building applications that require access to data held in a data source. This introduces ADO.NET and explains how you can use it to develop scalable, high-performance, data-driven applications.

**Lessons**

• Retrieving and Modifying Data by Using ADO.NET Commands

• Retrieving and Modifying Data by Using DataSets

• Managing Transactions and Concurrency in Multiuser Scenarios

**Lab: Using ADO.NET**

After completing this, students will be able to:

• Retrieve and update data by using ADO.NET commands and stored procedures.

• Retrieve and update data by using DataSet objects.

• Implement transactions and handle concurrency exceptions.

# 3. Program ASP.NET MVC

**(8 days)**

1. Introduction to MVC architecture

2. Introduction to ASP.NET

3. Routing

4. Controllers and Actions

5. Razor views

6. Bundling and minification

7. Areas

8. AJAX in ASP.NET

9. Scaffolding, editor templates, session, viewbag

10. Web security in ASP.NET

11. Caching data

12. Localization (optional)

**Web Development – ASP MVC – Server-side Development**

Would it be possible to also include coverage of the following topic areas:

* Exploring ASP.NET MVC4
* Designing ASP.NET MVC 4 Web Applications
* Developing ASP.NET MVC 4 Models
* Developing ASP.NET MVC 4 Controllers
* Developing ASP.NET MVC 4 Views
* Testing and Debugging ASP.NET MVC 4 Web Applications
* Structuring ASP.NET MVC 4 Web Applications
* Building Responsive Pages in ASP.NET MVC 4 Web Applications
* Using JavaScript and jQuery for Responsive MVC 4 Web Apps
* Implementing Web APIs in ASP.NET MVC 4 Web Applications
* Overview of new technologies used with ASP.NET
* Design and Architecture of ASP.NET MVC Web Applications

## Web Development – ASP MVC – Server-side Development

### Exploring ASP.NET MVC4

The goal of this is to outline to the students the components of the Microsoft Web Technologies stack, which can be used to host a completed web application. Students will also learn about ASP.NET 4.5 and be introduced to the web forms, web pages, and MVC programming models. Finally they will see an overview of ASP.NET MVC 4, including new features and configuration.

**Lessons**

• Overview of Microsoft Web Technologies

• Overview of ASP.NET 4.5

• Introduction to ASP.NET MVC 4

**Lab: Exploring ASP.NET MVC4**

After completing this, students will be able to:

• Describe the Microsoft Web Technologies stack and select an appropriate technology to use to develop any given application.

### Designing ASP.NET MVC 4 Web Applications

The goal of this is to introduce students to the typical design process that architects must complete when they plan an MVC 4 application. At this stage in the design process, MVC 4 has been selected as the most appropriate programming model, but the details of the application, such as the overall architecture, Controllers, Views, Models, and routes to create, have not been fixed. How to plan such details is shown during this.

**Lessons**

• Planning in the Project Design Phase

• Designing Models, Controllers, and Views

**Lab: Designing ASP.NET MVC 4 Web Applications**

After completing this, students will be able to:

• Design the architecture and implementation of a web application that will meet a set of functional requirements, user interface requirements, and address business models.

### Developing ASP.NET MVC 4 Models

The goal of this is to enable the students to create Models within an MVC application that implement the business logic necessary to satisfy business requirements. The also describes how to implement a connection to a database, or alternative data store, using the Entity Framework and LINQ.

**Lessons**

• Creating MVC Models

• Working with Data

**Lab: Developing ASP.NET MVC 4 Models**

After completing this , students will be able to:

• Create MVC Models and write code that implements business logic within Model methods, properties, and events.

### Developing ASP.NET MVC 4 Controllers

The goal of this is to enable students to add Controllers to MVC applications and to implement actions that respond to user input and other events. The students will learn how Controllers relate to Models and how to implement Controller actions that define the View used to display or edit data. This also covers how to write action filters that run code before or after multiple actions in the Controller. The students will learn about situations when action filters are useful.

**Lessons**

• Writing Controllers and Actions

• Writing Action Filters

**Lab: Developing ASP.NET MVC 4 Controllers**

After completing this, students will be able to:

• Add Controllers to an MVC Application to manage user interaction, update models, and select and return Views.

### Developing ASP.NET MVC 4 Views

The goal of this is to describe the role of Views in an MVC web application and enable users to create and code them. The syntax of a Razor View is of critical importance for students to understand because it defines both the layout and the functionality of the data display. HTML Helpers will also be discussed in detail and common Helpers, such as Html.ActionLink() and Html.EditorFor(), will be described. Reusing code by defining Partial Views and Razor Helpers will be discussed as well.

**Lessons**

• Creating Views with Razor Syntax

• Using HTML Helpers

• Reusing Code in Views

**Lab: Developing ASP.NET MVC 4 Views**

After completing this, students will be able to:

• Create Views in an MVC application that display and edit data and interact with Models and Controllers.

### Testing and Debugging ASP.NET MVC 4 Web Applications

The goal of this is to enable students to increase the resilience and quality of an application by locating and correcting code errors, bugs, and other unexpected results. MVC applications are well suited to unit testing techniques and these techniques ensure a high quality of code by systematically testing the functionality of each small component. In addition the debugging tools and exception handling available in Visual Studio will be explained.

**Lessons**

• Unit Testing MVC Components

• Implementing an Exception Handling Strategy

**Lab: Testing and Debugging ASP.NET MVC 4 Web Applications**

After completing this, students will be able to:

• Run unit tests and debugging tools against a web application in Visual Studio 2012 and configure an application for troubleshooting.

### Structuring ASP.NET MVC 4 Web Applications

The goal of this is to enable students to structure a web application in such a way that users can rapidly locate the information they need. Two aspects of the design are emphasized: the URLs presented in the browser address bar should be understandable and can be controlled by adding routes to the ASP.NET Routing Engine, and the navigation controls, such as menus and breadcrumb trails, should present the most relevant links to frequently read pages. Search Engine Optimization is important throughout this.

**Lessons**

• Analysing Information Architecture

• Configuring Routes

• Creating a Navigation Structure

**Lab: Structuring ASP.NET MVC 4 Web Applications**

After completing this, students will be able to:

• Develop a web application that uses the ASP.NET routing engine to present friendly URLs and a logical navigation hierarchy to users.

### Building Responsive Pages in ASP.NET MVC 4 Web Applications

The goal of this is to describe to the students how partial page updates and caching can optimize the responsiveness of a web application. Students will see how to make use of AJAX helpers and partial views to update small portions of a page instead of refreshing the entire page. The also covers the different caches developers can use to store rendered pages and discusses how to configure caching for maximum performance.

**Lessons**

• Using AJAX and Partial Page Updates

• Implementing a Caching Strategy

**Lab: Building Responsive Pages in ASP.NET MVC 4 Web Applications**

After completing this, students will be able to:

• Use partial page updates and caching to reduce the network bandwidth used by an application and accelerate responses to user requests.

### Using JavaScript and jQuery for Responsive MVC 4 Web Apps

The goal of this is to teach the students techniques that run code on the browser. This approach can increase the responsiveness of the application because a rendered page can respond to a user action without reloading the entire page from the server. Students will learn about the jQuery script library and how to use it to call web services and update user interface components.

**Lessons**

• Rendering and Executing JavaScript Code

• Using jQuery and jQueryUI

**Lab: Using JavaScript and jQuery for Responsive MVC 4 Web Applications**

After completing this, students will be able to:

• Write JavaScript code that runs on the client-side and utilizes the jQuery script library to optimize the responsiveness of an MVC web application.

### Implementing Web APIs in ASP.NET MVC 4 Web Applications

The goal of this to introduce the concept of a Web API to students and to describe how to make an application’s core functionality more broadly available for integration into other web and mobile applications. Students will learn about the new Web API feature of MVC 4 and see how to build a RESTful Web API and call it from other applications.

**Lessons**

• Developing a Web API

• Calling a Web API from Mobile and Web Applications

**Lab: Implementing Web APIs in ASP.NET MVC 4 Web Applications**

After completing this, students will be able to:

• Describe what a Web API is and why developers might add a Web API to an application.

### Overview of new technologies used with ASP.NET

The goal of the is to introduce the new technologies used together with ASP.NET MVC Application, like knockout, angular, typescript and so on.

**Lessons**

• Introduction to AngularJS, and Typescript

• Overview of knockout

**Lab: Implementing Web APIs in ASP.NET MVC 4 Web Applications**

After completing this, students will be able to:

• Add and use AngularJS and Typescript in ASP.NET Web applications

### Design and Architecture of ASP.NET MVC Web Applications

The goal of this is to introduce the best practices and good architectural designs used when developing ASP.MVC web applications.

**Lessons**

• ASP.NET Application Architectures

• Best practices when developing ASP.Net Web Apps