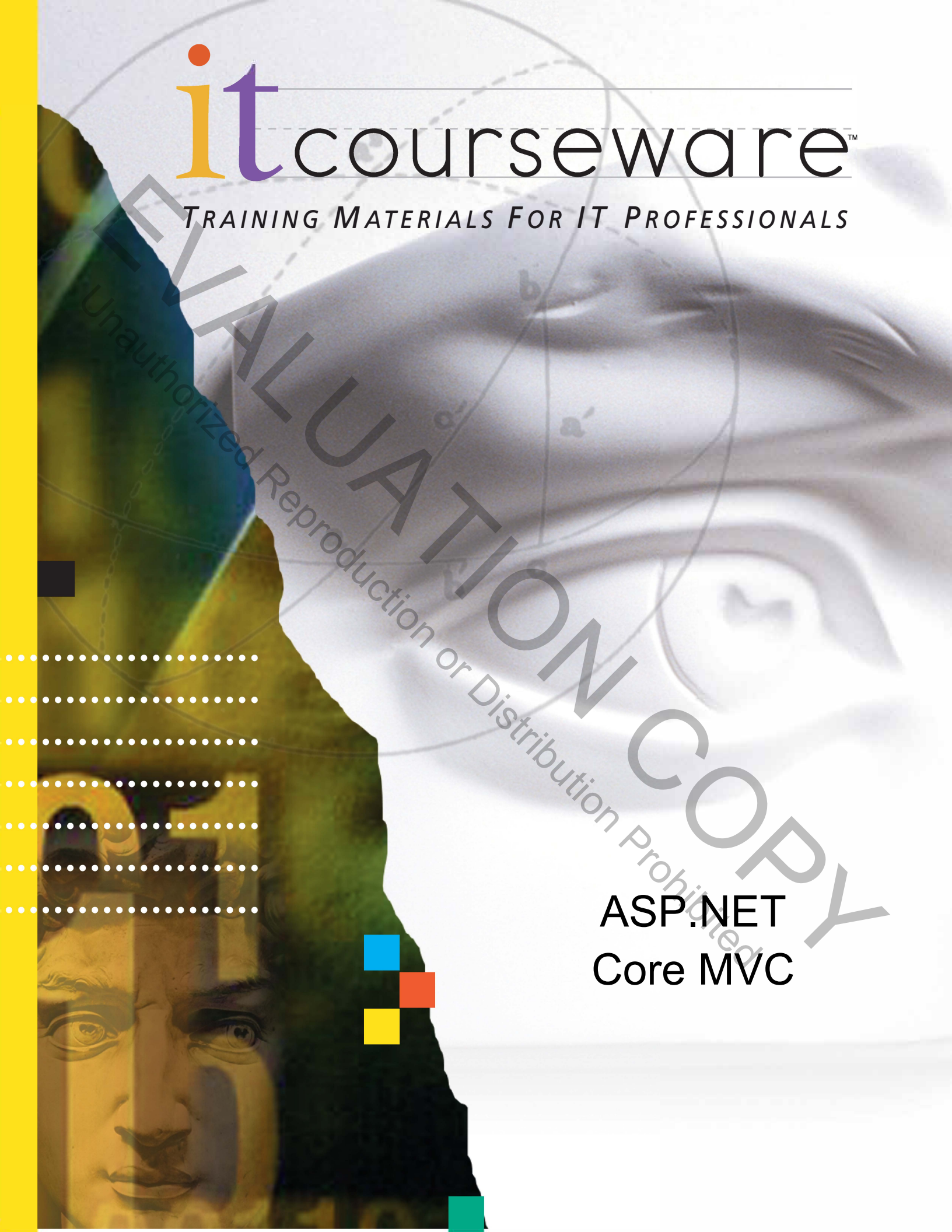

itcourseware™
TRAINING MATERIALS FOR IT PROFESSIONALS
EVALUATION COPY
Unauthorized Reproduction or Distribution Prohibited
ASP.NET
Core MVC

ASP.NET Core MVC
Rev. 5.0

Student Guide

™ is a trademark of Object Innovations.

Authors: Robert Hurlbut and Robert J. Oberg

Object Innovations
877-558-7246
www.objectinnovations.net

Published in the United States of America.

Table of Contents (Overview)

Directory Structure

- **The course software installs to the root directory *C:\OIC\MvcCore*.**
  - Example programs for each chapter are in named subdirectories of chapter directories **Chap02**, **Chap03**, and so on.
  - The **Labs** directory contains one subdirectory for each lab, named after the lab number. Starter code is frequently supplied, and answers are provided in the chapter directories.
  - The **Demos** directory is provided for doing in-class demonstrations led by the instructor.
  - The file **Links.htm** contains link to web sites and pages mentioned in the text along with other useful links.
- **Data files install to the directory *C:\OIC\Data*.**

Table of Contents (Detailed)

Chapter 1

Introduction to ASP.NET Core MVC

Introduction to ASP.NET Core MVC

Objectives

After completing this unit you will be able to:

- **Describe advantages and disadvantages of ASP.NET Web Forms.**
- **Understand the Model-View-Controller (MVC) pattern**
- **Outline the parts of an ASP.NET MVC application.**
- **Describe advantages of ASP.NET MVC and issues to consider in using this technology.**
- **Describe ASP.NET Core 5.0.**
- **Understand the use of unit testing in creating ASP.NET MVC applications.**

Review of ASP.NET Web Forms

- **ASP.NET Web Forms provide a way to build web applications.**
- **You can use compiled, object-oriented languages with ASP.NET, including C# and Visual Basic.**
  - All the power of the .NET Framework is available to you, including the extensive class library.
- **Code and presentation elements can be cleanly separated.**
  - Code can be provided in a separate section of a Web page from user interface elements.
  - The separation can be carried a step further by use of separate "code behind" files.
- **ASP.NET Web Forms comes with an extensive set of server controls that provide significant functionality out of the box.**
- **Server controls transparently handle browser compatibility issues.**
- **Configuration is handled by XML files without need of any registry settings, and deployment can be done simply by copying files.**

Advantages of ASP.NET Web Forms

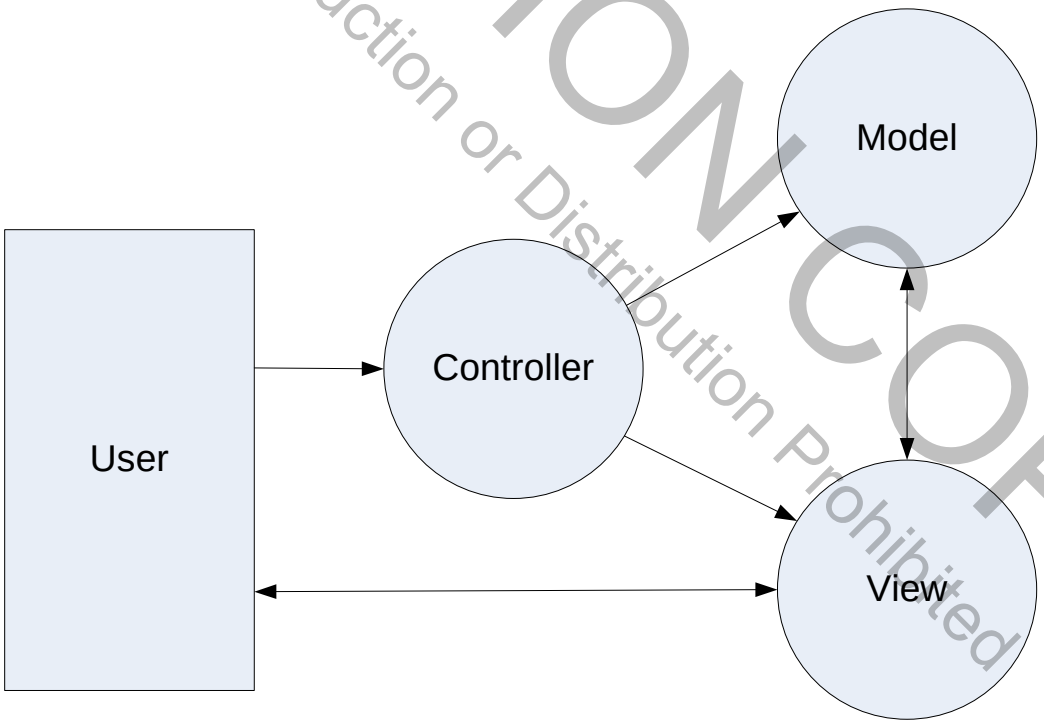
- **ASP.NET Web Forms continue to be supported and have their own advantages:**
  - A rich event model supported in hundreds of server controls facilitates easy development of Web server applications, following a familiar GUI development paradigm.
  - View state makes it easy to manage state information.
  - The model works well for individuals and small teams doing rapid application development.
  - The large number of built-in and third-party components also facilitates rapid application development.
- **In general, Web Forms are quite easy to work with and generally require less code.**
- **Important Note: Web Forms are *not* available in ASP.NET Core.**
  - The brief discussion of Web Forms in this introduction is intended to place in context the newer technology, ASP.NET MVC.

Disadvantages of ASP.NET Web Forms

- **Key disadvantages of ASP.NET Web Forms include**
  - ViewState tends to be large depending on the number of server controls contained on the page, thus increasing the size of the page and the length of the response time from server to browser
  - ASP.NET Web Forms provide tight coupling with the code-behind classes which make automated testing of the back-end code apart from the web pages more difficult
  - Because the code-behind classes are so tightly coupled to the web forms, developers are encouraged to mix presentation code with application logic in the same code-behind classes which can lead to fragile and unintelligible code
  - Limited control of HTML rendered through use of server controls

Model-View-Controller Pattern

- **The Model-View-Controller (MVC) design pattern divides an application into three conceptual components:**
  - A **model** represents the data and operations that are meaningful to the domain of the application. It implements the application logic for the domain.
  - **Views** display a user interface for portions of the model. Typically the UI is created from model data.
  - **Controllers** handle incoming requests, work with the model, and select a view to render a UI back to the user.



ASP.NET MVC

- **ASP.NET MVC is a framework based on ASP.NET for creating Web applications.**
  - It is an alternative to Web Forms in classical .NET.
- **ASP.NET MVC 1.0 was installed on top of .NET 3.5 SP1 and Visual Studio 2008 SP.**
- **ASP.NET MVC 2.0 is integrated into .NET 4.0 and Visual Studio 2010.**
- **ASP.NET MVC 3.0 is a separate download and adds important new features, such as the Razor view engine.**
- **ASP.NET MVC 4.0 is integrated into .NET 4.5 and Visual Studio 2012.**
- **ASP.NET MVC 5.0 is integrated into .NET beginning with .NET 4.5.2 and Visual Studio 2013.**
  - ASP.NET MVC 5 is covered in OI course 4143 .
- **ASP.NET MVC 6.0, part of ASP.NET Core, unifies MVC with Web API.**
- **The latest version is ASP.NET Core 5.0 MVC.**

ASP.NET Core

- **ASP.NET Core is a completely new version of ASP.NET that is open-source and cross-platform.**
  - It runs on Windows, Mac and Linux.
- **It runs on.NET Core.**
- **ASP.NET Core is very modular providing flexibility in using what is needed for an application with minimal overhead.**
- **ASP.NET Core supports both MVC and Web API in an integrated manner, but it does not support Web Forms.**
- **This course focuses on MVC running on .NET Core 5.0, with an introduction to Web API.**

What is .NET Core?

- **.NET Core is a general purpose development platform that is cross-platform, supporting:**
  - Windows
  - Mac
  - Linux
- **It is open-source (MIT license) maintained by Microsoft and the .NET community on GitHub.**
- **.NET Core is composed of:**
  - A .NET runtime
  - A set of framework libraries
  - A set of SDK tools and language compilers
  - The app host **dotnet**, which is used to launch .NET Core apps.
- **.NET Core is distributed through NuGet packages.**
  - Packages are relatively fine grained, allowing smaller app size.
  - Metapackages are more coarse-grained, describing a set of packages that are meaningful together.
- **NuGet is the package manager for the Microsoft development platform. It provides a gallery as a central repository.**

Advantages of ASP.NET MVC

- **Key advantages of ASP.NET MVC include:**
  - The MVC pattern promotes **separation of concerns** into input logic (controller), business logic (model) and UI (view). This aids in managing complexity.
  - These components are loosely coupled, promoting parallel development.
  - This loose coupling also facilitates automated testing.
  - Views are created using standard HTML and cascading style sheets, giving the developer a high degree of control over the user interface.
  - There is no view state, reducing the load on the browser in rendering a page.
- **Separation of Concerns:**
  - Each component has one responsibility
  - SRP – Single Responsibility Principle
  - DRY – Don't Repeat Yourself
  - More easily testable
  - Helps with concurrent development

ASP.NET MVC Considerations

- **Key considerations in using ASP.NET MVC include:**
  - Writing View contents the old ASP-like way (though this is now easier with the newer Razor view syntax).
  - Unit testing and Test Driven Development (TDD) are encouraged and used more but also bring a steep learning curve.
  - Need to understand HTML controls and style sheets, but at the same time this allows a designer to work independently of the coders.

Goals of ASP.NET MVC

- **The ASP.NET MVC Framework has the following goals:**
  - Frictionless Testability
  - Tight control over markup
  - User/Search Engine friendly URLs
  - Leverage the benefits of ASP.NET
  - Conventions and Guidance
- **Extensibility**
  - Replace any component of the system
  - Interface-based architecture
  - Very few sealed methods / classes

ASP.NET Core 5.0

- **.NET 5.0 is the next major release of .NET Core following 3.1.**
- **Microsoft dropped "Core" from the name to emphasize that this is the main implementation of .NET going forward.**
  - .NET 5.0 supports more types of apps and more platforms than .NET Core or .NET Framework.
- **ASP.NET Core 5.0 is based on .NET 5.0 but retains the name "Core" to avoid confusing it with ASP.NET MVC 5.**
- **Read more in “What’s new in .NET 5.0”.**

https://docs.microsoft.com/en-us/dotnet/core/dotnet-five

Unit Testing

- **Unit testing lets you specify the expected behavior of individual classes or other small code units in isolation.**
- **ASP.NET MVC encourages unit testing of the Models and the Controllers of the application to verify expected behaviors.**
- **Separation of concerns makes unit testing of individual components feasible.**
- **There are several choices for unit testing in .NET Core.**
  - The new **dotnet test** supports cross-platform testing.
  - Microsoft's MSTest is integrated into Visual Studio and can be used for testing .NET Core on Windows.
  - The open-source xUnit can be used for testing .NET Core applications.
- **This link provides information on unit testing in .NET Core:**

https://docs.microsoft.com/en-us/dotnet/core/testing/

Summary

- **ASP.NET Web Forms is still used and has both advantages and disadvantages compared to MVC.**
  - But Web Forms is not available in ASP.NET Core.
- **The Model-View-Controller (MVC) pattern is useful in creating applications that have separation of concerns.**
- **ASP.NET Core is open-source and cross platform and unifies MVC and Web API.**
- **Unit testing is helpful and encouraged in developing ASP.NET MVC applications.**

7400 E. Orchard Road, Suite 1450 N
Greenwood Village, Colorado 80111
Ph: 303-302-5280
www.ITCourseware.com