

- Still looking for a job?
- Malaysian Diploma / Degree holder?
- Interested in a career in ICT industry?

**FREE
TRAINING!**

JOIN MSC MALAYSIA JOB CAMP

Course Name: .NET DEVELOPER

Duration: 15 days

Training Location: Nota Asia (M) Sdn Bhd, Subang
Jaya / KL Plaza

.NET DEVELOPER

Modules Details / Curriculum

- Core Web Application Technologies with Microsoft Visual Studio 2005
- Advanced Web Application Technologies with Microsoft Visual Studio
- Building a 3-tier Architecture with ASP.NET 2.0
- Mock Exam

(Course outline as enclosed)

The Microsoft Certified Technology Specialist credential enables professionals to target specific technologies and distinguish themselves by demonstrating in-depth knowledge and expertise in their subject area technologies. Developers holding the Microsoft Certified Technology Specialist: .NET Framework 2.0 Web Applications (MCTS: .NET Framework 2.0 Web Applications) certification have demonstrated breadth and depth of skills and knowledge of Web application technology and data access in Web applications.



MSC Malaysia via its K-Workers Development Initiatives (KDI) is driving the provision of last mile skills training to potential knowledge workers (k-workers) for the ICT industry. Trainings are currently done through partnership with training providers appointed by Multimedia Development Corporation (MDeC).

MSC Malaysia Job Camp, a KDI programme, provides fresh graduates and available k-workers the necessary training to fill immediate vacancies in MSC, Malaysia Status companies.

For further inquiries, please contact:

Nota Asia (M) Sdn Bhd

Sharifah Zawanah

sufyana@notaasia.com

03.5636.2080

CT-08-04, Level 8, Subang Square Corporate Tower,
Jalan SS15/4G, 47500 Subang Jaya, Selangor.

In Collaboration



MSC MALAYSIA - Giving You the Edge Through ICT

.NET Web Developer

Duration: 15 days

Preparation Guide

Preparation towards MCTS: .NET Framework 2.0 Web Applications Certification

The Microsoft Certified Technology Specialist credential enables professionals to target specific technologies and distinguish themselves by demonstrating in-depth knowledge and expertise in their subject area technologies. Developers holding the Microsoft Certified Technology Specialist: .NET Framework 2.0 Web Applications (MCTS: .NET Framework 2.0 Web Applications) certification have demonstrated breadth and depth of skills and knowledge of Web application technology and data access in Web applications.

To achieve this certification, the candidate must pass two exams, one focusing on .NET Framework 2.0 foundational skills and one focusing on Web-based client development.

Required Exams for MCTS: .NET Framework 2.0 Web Applications (Two Exams Required):

- Exam 70-536: TS: Microsoft .NET Framework 2.0 - Application Development Foundation
- Exam 70-528: TS: Microsoft .NET Framework 2.0 - Web-Based Client Development

Stage 1: Core Web Application Technologies with Microsoft Visual Studio 2005 (3 days)

At Stage 1 Workshop Completion

After completing this workshop, students will be able to:

- Create a Web application.
- Program a Web application.
- Add and configure server controls for a Web application.
- Use master pages to establish a common layout for a Web application.
- Manage state data for a Web application.
- Access and display data in a Web application.
- Control access to a Web application.
- Deploy a Web application.
- Create a mobile Web application.

Workshop Outline

Unit 1: Creating a Web Application

This unit describes the different types of Web sites that you can create with Visual Studio 2005. It introduces the concept of event handling, and shows how to work with default event handlers for an object. It also explains how to control a Web application through the hierarchy of configuration files.

Lessons

- Visual Studio Web Site Types
- Default Event Handling in Web Applications
- Web Configuration Files

Lab 1: Creating a Web Application

- Exercise 1. Creating a New Web Application
- Exercise 2. Configuring and Building a Web Application

After completing this unit, students will be able to:

- Describe the types of Web sites that they can create with Visual Studio 2005.
- Describe the concept of a default event handler for an object.
- Explain how the Machine.Config and Web.Config files control the settings for a Web application.
- Create a new Web application.
- Configure and build a Web application.

Unit 2: Programming a Web Application

This unit introduces the advanced event-handling capabilities of ASP.NET 2.0 and describes how to work with events in Visual Studio 2005. It shows how to work with non-default event handlers and centralized event handlers. It also addresses other common Web programming concepts, including:

- Detecting the type, version, and capability of the browser being used to view a Web site.
- Accessing information in an ASP.NET Web Page header.
- Using the `HttpResponse.Write` method to provide feedback to users.
- Handling page-level errors.

Lessons

- Event Handling in Web Applications
- Browser Capability Detection
- Page Header Retrieval
- Page-Level and Application-Level Error Handling

Lab 2: Programming a Web Application

- Exercise 1. Implementing Non-Default Event Handlers
- Exercise 2. Detecting Browser Capabilities and Setting Page Header Properties
- Exercise 3. Handling Page-Level Exceptions

After completing this unit, students will be able to:

- Describe various event-handling techniques.
- Explain how to detect browser types and capabilities.
- Explain how to access page headers.
- Describe how to handle page-level errors and application-level errors.
- Implement advanced techniques for handling events.
- Implement browser-capability detection.
- Implement page-header manipulation.
- Implement page-level and application-level error handling.

Unit 3: Adding and Configuring Server Controls

This unit explains how to use the HTML controls and Web server controls provided by Visual Studio 2005 and ASP.NET 2.0. It shows how to design and build Web-based user interfaces, and it teaches how to program Web server controls. This unit also describes how the ASP.NET 2.0 postback model works and how it can be used.

Lessons

1. HTML Controls and Web Server Controls
2. Types of Web Server Controls
3. Working with Web Server Controls
4. The ASP.NET 2.0 Page Postback Model

Lab 3: Adding and Configuring Server Controls

- Exercise 1. Building Graphical User Interfaces with HTML Controls
- Exercise 2. Building Graphical User Interfaces with Web Server Controls
- Exercise 3. Programming Web Server Controls and Working with Postbacks

After completing this unit, students will be able to:

- Explain the differences between HTML controls and Web server controls.
- Describe the different types of Web server controls.
- Explain how to use HTML controls and Web server controls.
- Explain how the postback model of ASP.NET 2.0 works.
- Create Web-based user interfaces with HTML controls and Web server controls.
- Write code that interacts with Web server controls.
- Write code that interacts with the postback model of ASP.NET 2.0.

Unit 4: Creating a Common Layout by Using Master Pages

This unit explains how to use master pages to define common layouts for Web pages. Master pages provide developers with a new set of features for ensuring consistent page layout. Students will work with master pages and nested master pages in the lab to build a Web application that has a consistent layout and functionality across Web pages.

Lessons

- What Are Master Pages?
- What Are Content Pages?
- Nested Master Pages

Lab 4: Creating a Common Layout by Using Master Pages

- Exercise 1. Designing a Master Page
- Exercise 2. Adding and Configuring Content Pages
- Exercise 3. Designing Nested Master Pages

After completing this unit, students will be able to:

- Describe the concept of a master page.
- Describe the concept of a content page.
- Describe nested master pages.
- Design master pages.
- Configure content pages.
- Design nested master pages.

Unit 5: Managing State for a Web Application

This unit describes the different state management technologies that students can use in ASP.NET 2.0 Web applications. It discusses how controls can retain state data over multiple requests, and then explains how developers can work with this state data. This unit then shows how to store state data in the Application and Session objects provided by ASP.NET 2.0. It also discusses the different session-data storage mechanisms. Finally, this unit explains how to use the Cache object to cache and retrieve state data.

Lessons

- ViewState Properties and ControlState Data
- Application and Session Objects
- Strategies for Managing Session State Data
- The Cache Object

Lab 5: Managing State for a Web Application

- Exercise 1. Configuring ViewState Properties for Web Server Controls
- Exercise 2. Storing and Retrieving Application and Session State
- Exercise 3. Implementing Out-of-Process Session State
- Exercise 4. Storing and Managing State Data in the Cache Object

After completing this unit, students will be able to:

- Describe the ViewState and ControlState data models for Web pages.
- Describe the Application and Session objects and explain how state data is stored and retrieved in these objects.
- Describe various session-state data-storage strategies.
- Describe the Cache object and explain how you can use it to store and manage state data.
- Configure ViewState properties and ControlState properties for Web server controls.
- Store and retrieve Application and Session state.
- Implement out-of-process session state.
- Store and manage state data in the Cache object.

Unit 6: Accessing and Displaying Data

This unit describes how to add database connections to the Web.Config file and the benefits that this approach adds when building manageable Web applications. This unit then describes the new data controls for accessing

data in a variety of formats. It includes details about using the `SqlDataSource` control, the `XmlDataSource` control, and the `ObjectDataSource` control. This unit also describes how user interface data controls are bound to the data source controls, and it includes a discussion about binding data-aware standard controls to data.

Lessons

- Database Connections and the `Web.Config` File
- Relational Data and Data Source Controls
- XML Data and Data Source Controls
- Object Data and Data Source Controls

Lab 6: Accessing and Displaying Data

- Exercise 1: Creating and Retrieving Database Connections
- Exercise 2: Accessing Data by Using `SqlDataSource` Controls and Data Controls
- Exercise 3: Accessing Objects as Data with `ObjectDataSource` Controls
- Exercise 4: Accessing XML Data by Using `XmlDataSource` Controls

After completing this unit, students will be able to:

- Explain how to store and retrieve database connections by using the `Web.Config` file.
- Explain how to use data source controls to access relational data.
- Explain how to use data source controls to access XML data.
- Explain how to use data source controls to access object data.
- Create and retrieve database connections by using the `Web.Config` file.
- Access relational data by using the `SqlDataSource` control and data controls.
- Access XML data by using the `XmlDataSource` control and data controls.
- Access objects as data by using the `ObjectDataSource` control and data controls.

Unit 7: Controlling Access to a Web Application

This unit describes authentication and authorization for Web applications. It also shows how to develop login, sign-up, and other membership pages for Web applications based on the ASP.NET 2.0 Membership system.

Lessons

- Authentication for Web Applications
- Authorization for Web Applications
- Site Membership Systems Using the Membership Class
- Web Site Security Administration Using the Roles Class

Lab 7: Controlling Access to a Web Application

- Exercise 1: Configuring Authentication and Authorization for a Web Application
- Exercise 2: Implementing a Membership Registration Page
- Exercise 3: Implementing a Login Page and Adding Login Controls
- Exercise 4: Creating a Membership Management Administrative User Interface

After completing this unit, students will be able to:

- Describe the authentication methods for Web applications.
- Describe the authorization methods for Web applications.
- Describe the main components of a membership system.
- Describe how to build a security administration interface.
- Configure authentication and authorization for a Web application.
- Implement a membership registration page.
- Implement a login page.
- Create a membership management administrative user interface.

Unit 8: Deploying a Web Application

This unit describes three different ways to deploy Web applications:

- Using the Copy Web Site utility to deploy a Web application in a non-compiled state
- Using the Publish Web Site utility to deploy a precompiled version of the Web application

- Building Microsoft Windows(Installer packages to create a redistributable application with full setup logic

Lessons

- The Copy Web Site Utility
- The Publish Web Site Utility
- Windows Installer Setup Packages

Lab 8: Deploying a Web Application

- Exercise 1. Deploying a Web Application by Using the Copy Web Site Utility
- Exercise 2. Precompiling and Deploying a Web Application by Using the Publish Web Site Utility
- Exercise 3. Building a Windows Installer Package for Deploying a Web Application

After completing this unit, students will be able to:

- Describe how to use the Copy Web Site utility to deploy a Web application.
- Describe how to use the Publish Web Site utility to precompile and deploy a Web application.
- Describe how to build Windows Installer packages for deploying a Web application.
- Deploy a Web application by using the Copy Web Site utility.
- Precompile and deploy a Web application by using the Publish Web Site utility.
- Build and run a Windows Installer setup application for deploying a Web application.

Unit 9: Making Web Applications Available to Mobile Devices

This unit explains how to enable browsers running on mobile devices, such as Pocket PCs and mobile phones, to access pages within your application.

Lessons

- Device Emulators for Mobile Web Forms
- Mobile Device Detection and Redirection
- Mobile Web Forms
- Device-Specific Features in Mobile Web Forms

Lab 9: Making Web Applications Available to Mobile Devices

- Exercise 1. Managing Redirection for Mobile Devices
- Exercise 2. Designing and Implementing a Mobile Web Form
- Exercise 3. Designing Device-Specific Features for a Mobile Web Application
- Exercise 4. Browsing a Mobile Web Application with Specific Device Emulators

After completing this unit, students will be able to:

- Explain how to detect mobile devices and redirect them to an appropriate page in a Web application.
- Describe mobile Web pages, forms, and mobile controls.
- Explain how to use device-specific features in mobile Web pages to respond to the different capabilities of mobile devices.
- Explain how to use device emulators in Visual Studio 2005 to test mobile Web pages.
- Design and implement mobile Web forms.
- Design device-specific features for mobile Web pages.

Stage 2: Advanced Web Application Technologies with Microsoft Visual Studio 2005 (2 days)

At Stage 2 Workshop Completion

After completing this workshop, students will be able to:

1. Build dynamic Web applications.
2. Create controls for Web applications.
3. Optimize Web applications.
4. Build customizable Web applications.
5. Build Web Part pages and Web Parts.

Workshop Outline

Unit 1: Building Dynamic Web Applications

This unit introduces many different aspects of dynamic Web applications. It includes discussions on creating and configuring controls at run time. It then explains how to build dynamic globalization features into a Web application to ensure that it is localizable, including using localized resources and applying different master page layouts in response to culture and language settings. It concludes with explanations about how to enable dynamic configuration for site administrators.

Lessons

- Dynamic Control Creation
- Localization and Globalization
- Dynamic Master Pages
- Dynamic Web Configuration

Lab 1: Building Dynamic Web Applications

- Exercise 1. Dynamically Adding and Configuring Controls
- Exercise 2. Dynamically Applying Master Pages
- Exercise 3. Adding Localization Features
- Exercise 4. Dynamically Configuring Web Applications

After completing this unit, students will be able to:

- Explain dynamic control creation in ASP.NET 2.0.
- Add and configure controls dynamically.
- Explain how to incorporate globalization and localization features into Web applications.
- Add localization features to a Web application.
- Describe when and how to implement dynamic master pages.
- Apply master pages dynamically.
- Describe dynamic Web configuration scenarios.
- Dynamically configure Web applications.

Unit 2: Creating Controls for Web Applications

This unit explains how developers create different types of controls for different scenarios. The different types of controls include user controls, custom Web server controls, composite Web server controls, and templated controls.

Lessons

- User Controls
- Custom Web Server Controls
- Composite Web Server Controls
- Templated Controls

Lab 2: Creating Controls for Web Applications

- Exercise 1. Creating User Controls
- Exercise 2. Creating Custom Web Server Controls
- Exercise 3. Creating Composite Web Server Controls
- Exercise 4. Creating Templated Controls

After completing this unit, students will be able to:

- Describe user controls and the underlying enabling technologies.
- Create user controls.
- Describe custom Web server controls and the underlying enabling technologies.
- Create Web server controls.
- Describe composite controls and how composite controls are created.
- Create composite Web server controls.
- Describe templated controls and the interfaces that enable their implementation.
- Create templated controls.

Unit 3: Optimizing Web Application Performance

This unit introduces topics that will help you improve the performance of Web applications. It describes how the Page Scripting Object Model can help reduce the number of round trips for communication between the server and the browser, and then explains how tracing and instrumentation can be used to monitor and, therefore, improve the performance of a Web application. The unit discusses how caching and asynchronous processing can help increase Web application performance; it then highlights some considerations that developers must address if the Web application is to be deployed in a Web farm environment.

Lessons

- The Page Scripting Object Model
- Tracing and Instrumentation in Web Applications
- ASP.NET 2.0 Caching Techniques
- Asynchronous Processing in Web Applications
- Web Farm Development Considerations

Lab 3: Optimizing Web Application Performance

- Exercise 1. Accessing the Page Scripting Object Model
- Exercise 2. Implementing ASP.NET Caching Techniques
- Exercise 3. Implementing Tracing and Instrumentation Techniques in Web Applications
- Exercise 4. Implementing Asynchronous Processing in Web Applications

After completing this unit, students will be able to:

- Describe the Page Scripting Object Model.
- Access Page Scripting Object Model functionality.
- Explain how to use tracing and instrumentation to monitor and improve the performance of a Web application.
- Implement tracing and instrumentation in Web applications.
- Describe ASP.NET 2.0 caching techniques.
- Implement ASP.NET 2.0 caching techniques.
- Explain how asynchronous processing can lead to improved performance for Web applications.
- Implement asynchronous processing in Web applications.
- Describe strategies for dealing with session state management issues when deploying Web applications in a Web farm environment.
- Develop Web applications for Web farm environments.

Unit 4: Implementing Personalization and Themes in Web Applications

This unit introduces building customizable functionality into a Web application by adding personalization support. It discusses using the personalization features of ASP.NET 2.0 to provide this functionality. In addition, it discusses applying themes to Web applications and allowing users to choose color schemes to personalize their

experience in using the Web application. It concludes by explaining how to include features that enable users to personalize themes.

Lessons

- ASP.NET 2.0 Personalization Features
- Theme Support in ASP.NET 2.0

Lab 4: Implementing Personalization and Themes in Web Applications

- Exercise 1. Configuring Personalization
- Exercise 2. Implementing Personalization Functionality
- Exercise 3. Adding Themes to the Web Application
- Exercise 4. Implementing Personalized Themes

After completing this unit, students will be able to:

- Describe the personalization features provided by ASP.NET 2.0.
- Describe ASP.NET 2.0 theme support.
- Configure personalization for a Web application.
- Implement personalization features.
- Add themes to a Web application.
- Implement customizable themes.
-

Unit 5: Building Web Part Pages and Web Parts

This unit introduces the concept of a Web part, and describes how it is used in portal pages and other scenarios. It introduces the concept of a Web part page, and discusses how a Web part page contains some Web parts that provide the user interface, along with other controls that manage the Web part infrastructure. Additionally, it introduces the advanced features of connected Web parts and discusses scenarios where they are typically used.

Lessons

- What Is a Web Part?
- What Is a Web Part Page?
- Connected Web Parts

Lab 5: Building Web Part Pages and Web Parts

- Exercise 1. Creating a Web Part Page
- Exercise 2. Creating a Web Part
- Exercise 3. Creating Connected Web Parts

After completing this unit, students will be able to:

- Describe what a Web Part is and the purpose of Web Parts.
- Describe the components of a Web Part page and identify scenarios when Web Part pages are useful features of Web applications.
- Describe the more advanced features of Web Parts, including connections between Web Parts.
- Create Web Part pages.
- Create Web Parts.
- Create connected Web Parts.

Stage 3: Building a 3-tier Architecture with ASP.NET 2.0 (Project) (9 days)

Project Outline

Task 1: Building Basic Components

- Task 1.1: Creating a Data Access Layer
- Task 1.2: Creating a Business Logic Layer
- Task 1.3: Master Pages and Site Navigation

Task 2: Building Basic Reporting Pages

- Task 2.1: Displaying Data With the ObjectDataSource
- Task 2.2: Declarative Parameters
- Task 2.3: Programmatically Setting the ObjectDataSource's Parameter Values

Task 3: Building Master/Detail Pages

- Task 3.1: Master/Detail Filtering With a DropDownList
- Task 3.2 : Master/Detail Filtering With Two DropDownLists
- Task 3.3: Master/Detail Filtering Across Two Pages
- Task 3.4: Master/Detail Using a Selectable Master GridView with a Details

Task 4: Implementing Custom Formatting

- Task 4.1: Custom Formatting Based Upon Data
- Task 4.2: Using TemplateFields in the GridView Control
- Task 4.3: Using TemplateFields in DetailsView
- Task 4.4: Using the FormView's Templates

Task 5: Editing, Inserting, and Deleting Data

- Task 4.1: An Overview of Inserting, Updating, and Deleting Data
- Task 4.2: Examining the Events Associated with Inserting, Updating, and Deleting
- Task 4.3: Handling BLL-Level and DAL-Level Exceptions in an ASP.NET Page
- Task 4.4: Adding Validation Controls to the Editing and Inserting Interfaces
- Task 4.5: Customizing the Data-Modification Interface
- Task 4.6: Implementing Optimistic Concurrency
- Task 4.7: Adding Client-Side Confirmation when Deleting
- Task 4.8: Limiting Data-Modification Functionality Based on the User

Task 6: Implementing Paging and Sorting

- Task 6.1: Paging and Sorting Report Data

Stage 4: Mock Exams (1 day)

Required Exams for MCTS: .NET Framework 2.0 Web Applications (Two Exams Required):

- Exam 70-536: TS: Microsoft .NET Framework 2.0 - Application Development Foundation
- Exam 70-528: TS: Microsoft .NET Framework 2.0 - Web-Based Client Development

Course Structure

Week	Duration	Module
1	3 days	Stage 1: Core Web Application Technologies with Microsoft Visual Studio 2005
	2 days	Stage 2: Advanced Web Application Technologies with Microsoft Visual Studio 2005
2 & 3	9 days	Stage 3: Building a 3-tier Architecture with ASP.NET 2.0 (Project)
	1 day	Stage 4: Mock Exam