

## **Exam 70-480/Course 20480B**

# **Programming in HTML5 with JavaScript and CSS3**

### **About this course**

This course provides an introduction to HTML5, CSS3, and JavaScript. This course helps students gain basic HTML5/CSS3/JavaScript programming skills. This course is an entry point into both the Web application and Windows Store apps training paths. The course focuses on using HTML5/CSS3/JavaScript to implement programming logic, define and use variables, perform looping and branching, develop user interfaces, capture and validate user input, store data, and create well-structured application.

The lab scenarios in this course are selected to support and demonstrate the structure of various application scenarios. They are intended to focus on the principles and coding components/structures that are used to establish an HTML5 software application.

This course uses Visual Studio 2012, running on Windows 8.

### **Audience profile**

This course is intended for professional developers who have 6-12 months of programming experience and who are interested in developing applications using HTML5 with JavaScript and CSS3 (either Windows Store apps for Windows 8 or web applications).

This course is intended for students who have the following experience:

- 1 – 3 months experience creating Web applications, including writing simple JavaScript code
- 1 month experience creating Windows client applications
- 1 month of experience using Visual Studio 2010 or 2012

This course is not intended for developers with three or more months of HTML5 coding experience.

Students choosing to attend this course without a developer background should pay special attention to the training prerequisites. Developers who have more than 5 years programming experience may find that portions of this training are fundamental in nature when presenting the syntax associated with certain programming tasks.

Individuals who are interested in taking exam [70-480](#): Programming in HTML5 with JavaScript and CSS3, can also attend this course.

### **At course completion**

After completing this course, students will be able to:

- Explain how to use Visual Studio 2012 to create and run a Web application.
- Describe the new features of HTML5, and create and style HTML5 pages.
- Add interactivity to an HTML5 page by using JavaScript.
- Create HTML5 forms by using different input types, and validate user input by using HTML5 attributes and JavaScript code.
- Send and receive data to and from a remote data source by using XMLHttpRequest objects and jQuery AJAX operations.
- Style HTML5 pages by using CSS3.

- Create well-structured and easily-maintainable JavaScript code.
  - Use common HTML5 APIs in interactive Web applications.
  - Create Web applications that support offline operations.
  - Create HTML5 Web pages that can adapt to different devices and form factors.
  - Add advanced graphics to an HTML5 page by using Canvas elements, and by using and Scalable Vector Graphics.
  - Enhance the user experience by adding animations to an HTML5 page.
  - Use Web Sockets to send and receive data between a Web application and a server.
  - Improve the responsiveness of a Web application that performs long-running operations by using Web Worker processes.
- 

## Course Outline

### Module 1: Overview of HTML and CSS

This module provides an overview of HTML and CSS, and describes how to use Visual Studio 2012 to build a Web application.

#### Lessons

- Overview of HTML
- Overview of CSS
- Creating a Web Application by Using Visual Studio 2012

### Lab : Exploring the Contoso Conference Application

After completing this module, students will be able to:

- Describe basic HTML elements and attributes.
- Explain the structure of CSS.
- Describe the tools available in Visual Studio 2012 for building Web applications.

**Module 2: Creating and Styling HTML5 Pages** This module describes the new features of HTML5, and explains how to create and style HTML5 pages.

#### Lessons

- Creating an HTML5 Page
- Styling an HTML5 Page

### Lab : Creating and Styling HTML5 Pages

After completing this module, students will be able to:

- Create static pages using the new features available in HTML5.
- Use CSS3 to apply basic styling to the elements in an HTML5 page.

**Module 3: Introduction to JavaScript** This module provides an introduction to the JavaScript language, and shows how to use JavaScript to add interactivity to HTML5 pages.

#### Lessons

- Overview of JavaScript Syntax
- Programming the HTML DOM with JavaScript

- Introduction to jQuery

### **Lab : Displaying Data and Handling Events by Using JavaScript**

After completing this module, students will be able to:

- Explain the syntax of JavaScript and describe how to use JavaScript with HTML5.
- Write JavaScript code that manipulates the HTML DOM and handles events.
- Describe how to use jQuery to simplify code that uses many common JavaScript APIs.

**Module 4: Creating Forms to Collect Data and Validate User Input** This module describes the new input types available with HTML5, and explains how to create forms to collect and validate user input by using the new HTML5 attributes and JavaScript code.

#### **Lessons**

- Overview of Forms and Input Types
- Validating User Input by Using HTML5 Attributes
- Validating User Input by Using JavaScript

### **Lab : Creating a Form and Validating User Input**

After completing this module, students will be able to:

- Create forms that use the new HTML5 input types.
- Validate user input and provide feedback by using the new HTML5 attributes.
- Write JavaScript code to validate user input and provide feedback in cases where it is not suitable to use HTML5 attributes

**Module 5: Communicating with a Remote Data Source** This module describes how to send and receive data to and from a remote data source by using an XMLHttpRequest object and by performing jQuery AJAX operations.

- Sending and Receiving Data by Using XMLHttpRequest
- Sending and Receiving Data by Using jQuery AJAX operations

### **Lab : Communicating with a Remote Data Source**

After completing this module, students will be able to:

- Serialize, deserialize, send, and receive data by using XMLHttpRequest objects.
- Simplify code that serializes, deserializes, sends, and receives data by using the jQuery ajax method

**Module 6: Styling HTML5 by Using CSS3** This module describes how to style HTML5 pages and elements by using the new features available in CSS3.

- Styling Text
- Styling Block Elements
- CSS3 Selectors
- Enhancing Graphical Effects by Using CSS3

### **Lab : Styling Text and Block Elements using CSS3**

After completing this module, students will be able to:

- Style text elements on an HTML5 page by using CSS3.
- Apply styling to block elements by using CSS3.
- Use CSS3 selectors to specify the elements to be styled in a Web application.
- Implement graphical effects and transformations by using the new CSS3 properties.

**Module 7: Creating Objects and Methods by Using JavaScript** This module explains how to write well-structured and easily maintainable JavaScript code, and how to apply object-oriented principles to JavaScript code in a Web application.

- Writing Well-Structured JavaScript
- Creating Custom Objects

- Extending Objects

#### **Lab : Refining Code for Maintainability and Extensibility**

After completing this module, students will be able to:

- Describe the benefits of structuring JavaScript code carefully to aid maintainability and extensibility.
- Explain best practices for creating custom objects in JavaScript.
- Describe how to extend custom and native objects to add functionality.

**Module 8: Creating Interactive Pages using HTML5 APIs** This module describes how to use some common HTML5 APIs to add interactive features to a Web application. This module also explains how to debug and profile a Web application. **Lessons**

- Interacting with Files
- Incorporating Multimedia
- Reacting to Browser Location and Context
- Debugging and Profiling a Web Application

#### **Lab : Creating Interactive Pages by Using HTML5 APIs**

After completing this module, students will be able to:

- Use the Drag and Drop, and the File APIs to interact with files in a Web application.
- Incorporate audio and video into a Web application.
- Detect the location of the user running a Web application by using the Geolocation API.
- Explain how to debug and profile a Web application by using the Web Timing API and the Internet Explorer Developer Tools.

**Module 9: Adding Offline Support to Web Applications** This module describes how to add offline support to a Web application, to enable the application to continue functioning in a user's browser even if the browser is disconnected from the network. **Lessons**

- Reading and Writing Data Locally
- Adding Offline Support by Using the Application Cache

#### **Lab : Adding Offline Support to a Web Application**

After completing this module, students will be able to:

- Save and retrieve data locally on the user's computer by using the Local Storage API.
- Provide offline support for a Web application by using the Application Cache API.

**Module 10: Implementing an Adaptive User Interface** This module describes how to create HTML5 pages that can dynamically detect and adapt to different devices and form factors. **Lessons**

- Supporting Multiple Form Factors
- Creating an Adaptive User Interface

#### **Lab : Implementing an Adaptive User Interface**

After completing this module, students will be able to:

- Describe the need to detect device capabilities and react to different form factors in a Web application.
- Create a Web page that can dynamically adapt its layout to match different form factors.

**Module 11: Creating Advanced Graphics** This module describes how to create advanced graphics for an HTML5 Web application by using a Canvas element, and by using Scalable Vector Graphics. **Lessons**

- Creating Interactive Graphics by Using Scalable Vector Graphics
- Programmatically Drawing Graphics by Using a Canvas

#### **Lab : Creating Advanced Graphics**

After completing this module, students will be able to:

- Use Scalable Vector Graphics to add interactive graphics to an application.
- Draw complex graphics on an HTML5 Canvas element by using JavaScript code.

**Module 12: Animating the User Interface** This module describes how to enhance the user experience in an HTML5 Web application by adding animations. **Lessons**

- Applying CSS Transitions
- Transforming Elements
- Applying CSS Key-frame Animations

**Lab : Animating User Interface Elements**

After completing this module, students will be able to:

- Apply CSS transitions to elements on an HTML5 page, and write JavaScript code to detect when a transition has occurred.
- Describe the different types of 2D and 3D transitions available with CSS3
- Implement complex animations by using CSS key-frames and JavaScript code.

**Module 13: Implementing Real-Time Communications by Using Web Sockets** This module explains how to use Web Sockets to transmit and receive data between an HTML5 Web application and a server. **Lessons**

- Introduction to Web Sockets
- Sending and Receiving Data by Using Web Sockets

**Lab : Implementing Real-Time Communications by Using Web Sockets**

After completing this module, students will be able to:

- Explain how Web Sockets work and describe how to send and receive data through a Web Socket.
- Use the Web Socket API with JavaScript to connect to a Web Socket server, send and receive data, and handle the different events that can occur when a message is sent or received.

**Module 14: Creating a Web Worker Process** This module describes how to use Web Worker Processes to perform long-running operations asynchronously and improve the responsiveness of an HTML5 Web application. **Lessons**

- Introduction to Web Workers
- Performing Asynchronous Processing by Using a Web Worker

**Lab : Creating a Web Worker Process**

After completing this module, students will be able to:

- Describe the purpose of a Web Worker process, and how it can be used to perform asynchronous processing as well as provide isolation for sensitive operations.
- Use the Web Worker APIs from JavaScript code to create, run, and monitor a Web Worker process