



College Curriculum

**WEB
ADMINISTRATION
MODULE**



*Funded in part by a grant from the
National Science Foundation.
Grant No. DUE9813446.*

*Opinions expressed herein are those of the
authors and do not reflect those of the
National Science Foundation.*

WEB ADMINISTRATION (WEBMASTER)



For information on NWCET Compliance Review, please contact NWCET.

For more information, contact:

National Workforce Center for Emerging Technologies
3000 Landerholm Circle SE, N258
Bellevue, WA 98007-6484

Web: www.nwcet.org

Email: nwinfo@bcc.ctc.edu

Phone: (425) 564-4215

Fax: (425) 564-6193

Copyright © 2002

National Workforce Center for Emerging Technologies (NWCET)
through Bellevue Community College (BCC). All rights reserved.

Permission to Photocopy and Quote

General permission is granted to educators to photocopy limited material from prepared curriculum for non-commercial instructional or scholarly use. Permission must be sought from BCC in order to charge for photocopies, to quote material in advertising, or to reprint substantial portions of the document in other publications. Credit should always be given to the source of the photocopies or quotes by citing a complete reference.

To order additional copies or for more information on NWCET curriculum products, visit our website at:

www.nwcet.org

or call us at:

(425) 564-4215

TABLE OF CONTENTS

Overview of Webmaster Program Course Descriptions

Course 1: Introduction To The Internet and The World Wide Web

Course 2: Web Page Programming Using Javascript

Course 3: Digital Imaging Foundations

Course 4: Web Site Design Principles

Course 5: Internet Server Technologies

Course 6: Database Design and Development For Web Developers

Course 7: Creating Database-Driven Web Pages Using Open Source

**Course 8: Creating N-Tier Web Applications Using Activeserver Pages and
Activex Data Objects**

Course 9: Introduction to Java Programming

Course 10: Building Web Applications Using XML

This print version of a competency-based curriculum for Web Administration (WebMaster) flows from work begun as an online courseware project between National Workforce Center for Emerging Technologies (NWCET) and DrydenOnline ® Division of Harcourt College Publishers. The curriculum is based on the information technology skill standards published by the NWCET (*Building a Foundation for Tomorrow: Skill Standards for Information Technology, 1999*).

This document includes a program overview (course descriptions) and ten courses. Each course includes the following:

- . ⑩ **Course Overview:** A brief description of the content/topical areas covered in the course and the high-level goal of the course within the Web Administration (WebMaster) Program.
- . ⑩ **Instructional Goals:** A list of the “content obligations” the instructor has for this effective delivery of the course.
- . ⑩ **Course Prerequisites:** A list of the required or recommended courses, knowledge, and skills the learner needs to be successful in the course.
- . ⑩ **Learning Outcomes:** A list of what the learner must know and be able to do as a result of successfully completing the course.
- . ⑩ **Performance Indicators:** A list of specific, observable knowledge and skills that support and lead to the achievement of the learning outcomes.
- . ⑩ **Module Titles:** A list of the modules covered within the course.

Additionally, each module within a course contains learning outcomes, performance indicators, and identified module topics.

We wish to thank Karen Braunstein-Post, the content outline architect/developer, for her thorough research and generation of this information. Ms. Braunstein-Post is a faculty member of the Renton Technical College Computer Science/Information Technology Program.

® DrydenOnline is a registered trademark of Harcourt Corporation

	Course Title	Course Description
1	Introduction To The Internet and The World Wide Web	This ten-module course is designed to be an introduction to the terms, concepts, and development technologies used on the Internet and World Wide Web. Topics covered include: use of the Internet, Internet communication technologies, Internet browsers and search engines, HTML Web authoring and file transfer, and management.
2	Web Page Programming Using Javascript	This course is an introduction to client-side scripting using the popular programming language JavaScript. JavaScript is a powerful programming tool used to create interactivity, animation, and flexibility in Web pages. In addition to basic JavaScript syntax, this course focuses on programming structure and concepts.
3	Digital Imaging Foundations	An understanding of the concepts, terms, and techniques of digital imaging is very important for the Web developer. This course thoroughly covers the creation and modification of digital images using Adobe Photoshop.
4	Web Site Design Principles	This course is intended as an overview of the principles of effective Web site design. Creating effective and successful Web sites requires identification of the site's audience; an understanding of efficient, user-friendly interfaces, and a logical structure.
5	Internet Server Technologies	This ten-module course is designed to introduce the Web development student to the hardware, software, and protocols used on Internet servers. Internet server software and the technologies it supports will determine what features and tools a developer may employ when creating Web applications. This course provides information on the most critical Internet information services such as email, file transfer, and e-commerce.
6	Database Design and Development For Web Developers	This ten-module course is designed to introduce the E-Commerce student to database design, development concepts, and the client-server software model as they apply to Web site development. Database-driven Web applications are central to E-Commerce and business-to-business technology on the Internet.
7	Creating Database-Driven Web Pages Using Open Source	This course is designed to introduce the Web development student to building database-driven pages using "open source" tools available on the Internet. The tools used in this course will be PHP (Hypertext Preprocessor), which is a server-side scripting language, and MySQL, a relational database management system.
8	Creating N-Tier Web Applications Using Activeserver Pages and Activex Data Objects	This ten-module course is designed for students who are already familiar with HTML, Visual Basic scripting, and database concepts. Microsoft's Active Server Page (ASP) technology allows the Internet developer to create browser-independent, dynamic Web pages by combining programmatic code with HTML. Students will complete hands-on projects utilizing these technologies to create a three-tier client-server Web application.

9	Introduction to Java Programming	This ten-module course provides an introduction to the Java programming language for the Web administration student. Course content includes basic syntax, control structures, Java methods, objects and classes, graphics, interfaces, and applets. Students will have the opportunity to practice with projects and examples in all of these areas.
10	Building Web Applications Using XML	This ten-module course is designed for the student who is already familiar with HTML and database concepts. Extensible Markup Language (XML), an extension of Standard Generalized Markup Language (SGML), can be used to create Web pages, structure page layouts with frames, tables, and forms; link to external sources, enhance content using style sheets, and incorporate database content.

COURSE 1: INTRODUCTION TO THE INTERNET

AND THE WORLD WIDE WEB

Overview

.....
.....1

Module 1: Getting Started with the Net
.....5

Module 2: Internet Communications.....6

Module 3: Browsers and Search Engines
.....7

Module 4: Web Authoring Basics

.....8

Module 5: Web File Transfer and Management
.....9

Module 6:
Imaging.....
.....10

Module 7: Web Design
Techniques.....11

Module 8: Creating Web Interactivity using Client-side Scripting
.....12

Module 9: Introduction to Client-side Scripting using VBScript
.....13

Module 10: Using VBScript for
DHTML.....14

This ten-module course is designed to be an introduction to the terms, concepts, and

development technologies used on the Internet and World Wide Web. Topics covered include: use of the Internet, Internet communication technologies, Internet browsers and search engines, HTML Web authoring and file transfer, and management. The course ends with an introduction to client-side Web scripting and basic programming concepts using VBScript. This course is an overview that introduces topics that will be covered in greater detail in later courses.

Instructional Goals

- . ⑩ Explain the fundamental uses of the Internet in today's society
- . ⑩ Describe the role of Internet service providers
- . ⑩ Describe client/server architecture and how it is used on the Internet
- . ⑩ Use email, Internet browsers, search engines, and File Transfer Protocol software
- . ⑩ Create Web pages using HTML and store them on an Internet server
- . ⑩ Describe the most popular digital imaging file formats used on the Internet
- . ⑩ Demonstrate basic programming techniques using client-side scripting tools

Course Prerequisites

- . ⑩ Basic computer skills such as keyboarding and screen navigation
- . ⑩ Fundamental file handling skills such as opening, saving, copying, and pasting

Learning Outcomes

- . ⑩ Describe the current status and future development trends of the Internet
- . ⑩ Categorize major ways in which the Internet is being used
- . ⑩ Describe the basic services of Internet Service Providers (ISP) including dial-up, Web hosting, and secure socket connections
- . ⑩ Conduct an analysis of your needs for online service and select an ISP with services that match your needs
- . ⑩ Select a domain name and create a unique presence on the Internet
- . ⑩ Describe what is meant by "client/server architecture"
- . ⑩ Describe and explain why you would use Telnet
- . ⑩ Describe the three kinds of electronic mail systems—server-side, client-side, and hybrid, and in what situations they are most useful
- . ⑩ Set up and use Pine (server-side email software) and Eudora (client-side email software) to send electronic messages and identify differences between the two kinds of email services
- . ⑩ Distinguish between mailing lists and newsgroups
- . ⑩ Explain netiquette and its importance to email communication on the Internet
- . ⑩ Describe privacy considerations you must make when sending email messages on the Internet
- . ⑩ Use basic features of Microsoft Internet Explorer and Netscape Communicator
- . ⑩ Compare the relative strengths of Internet Explorer and Communicator

- . ⑩ Describe the distinction between a directory service and search engine
- . ⑩ Conduct an information search using both a directory service and a search engine
- . ⑩ Utilize good searching techniques to find information on the Internet
- . ⑩ Describe how Internet Explorer and Netscape browsers display a document on the World Wide Web
- . ⑩ Use HTML (Hypertext Markup Language) to build a Web-ready document that includes images, internal links, and external links
- . ⑩ Describe and demonstrate the difference between absolute, root, relative, and internal referencing
- . ⑩ Build or edit a Web document using server-side scripts written in Pico
- . ⑩ Describe the role and parts of a URL (Uniform Resource Locator) on the World Wide Web

Learning Outcomes, *continued*

- . ⑩ Download, install, and use WS_FTP (PC) or Fetch (Mac) to upload or download files from one computer to another on a network
- . ⑩ Download, install, and use WinZip (PC) or ZipIt (Mac) to compress files for transfer on the Internet
- . ⑩ Utilize UNIX commands to move, copy, delete, and rename files or directories on a server
- . ⑩ Apply the UNIX chmod command to enable permissions for the site owner, network group, and other visitors to read, write, and execute files on a server
- . ⑩ Describe the difference between using encryption and CyberCash for carrying out online Internet financial transactions
- . ⑩ Describe a firewall and how it protects a network from hackers
- . ⑩ Determine image file sizes on Windows and Mac computers
- . ⑩ Distinguish between GIF and JPEG image file formats and when to use them
- . ⑩ Create and manipulate GIF and JPEG images, and use them appropriately within a Web document
- . ⑩ Describe the difference between a client-side image map and server-side image map
- . ⑩ Create three kinds of image maps and incorporate associated HTML, including hyperlinks, within a Web page
- . ⑩ Describe the PNG image file format and why it is favorably considered as a graphic file format for the Internet
- . ⑩ Describe a Flash animation and why it is an efficient way to create animated graphics on the Web
- . ⑩ Use tables to position text and graphics on a Web page
- . ⑩ Describe and utilize nested tables
- . ⑩ Use cell padding and cell spacing attributes to format cells and cell content within a table
- . ⑩ Use tiling to create a custom background on a Web page
- . ⑩ Create a transparent pixel graphic with height and weight attributes to precisely position elements on a Web page

- . ⑩ Build a frameset document in HTML using the <FRAMESET> and <FRAME> tags, and target content to specified screen sections
- . ⑩ Describe a floating frame, and how it is used
- . ⑩ Describe the function of a Cascading Style Sheet
- . ⑩ Describe the difference between an Internal Style Sheet and External Style Sheet
- . ⑩ Articulate how interactivity can work to achieve the goals of a Web site
- . ⑩ Describe CGI (Common Gateway Interface) and implement a simple CGI-based form on a Web page
- . ⑩ Describe JavaScript, how it differs from Java, and its primary role in Web authoring
- . ⑩ Build a JavaScript rollover on a Web page
- . ⑩ Describe Java, its main features, and how it applies to Web authoring
- . ⑩ Implement and demonstrate how a Java applet is used on a Web page
- . ⑩ Describe situations where each of these technologies could be used to create interactive experiences on the Web
- . ⑩ Explain the differences between VBScript and JavaScript
- . ⑩ Explain the basic structure of a VBScript block

Learning Outcomes, *continued*

- . ⑩ Describe the difference between placing a script block in the <head> section vs. the <body> section of a Web page
- . ⑩ Describe the differences between Visual Basic and VBScript
- . ⑩ Demonstrate use of variables, sequence structures, branches, and looping structures using VBScript
- . ⑩ Create and test simple event procedures for controls placed on a form in a Web page
- . ⑩ Define Dynamic HTML and its purpose
- . ⑩ Explain “element naming” and its purpose
- . ⑩ Define the major components of the Document Object Model and give examples of some methods, properties, and events
- . ⑩ Explain what is meant by the term “event handling”
- . ⑩ Describe event bubbling
- . ⑩ Create a simple event procedure using VBScript to modify the style settings of an element
- . ⑩ Use DIV and SPAN tags to create programmable areas in a Web page
- . ⑩ Explain the use of the “innerHTML” and “outerHTML” properties
- . ⑩ Write an event procedure to alter the content of an element at runtime

Performance Indicators

- . ⑩ Current uses of the Internet and the World Wide Web can be clearly explained
- . ⑩ Role of the ISP can be appropriately described
- . ⑩ Concept of client/server architecture and its role on the Internet can be clearly described
- . ⑩ Email, Internet browsers, search engines, and File Transfer Protocol can

be used appropriately

- . ⑩ Web pages can be successfully created and stored on an Internet server using HTML
- . ⑩ The concept of digital imaging and most common image file formats can be clearly described
- . ⑩ Fundamental programming techniques can be appropriately demonstrated using client-side scripting tools (VBScript and JavaScript)

Module Titles

1. 1. Getting Started with the Net
2. 2. Internet Communications
3. 3. Browsers and Search Engines
4. 4. Web Authoring Basics
5. 5. Web File Transfer and Management
6. 6. Imaging
7. 7. Web Design Techniques
8. 8. Creating Web Interactivity using Client-side Scripting
9. 9. Introduction to Client-side Scripting using VBScript
10. 10. Using VBScript for DHTML

Learning Outcomes

- . ⑩ Describe the current status and future development trends of the Internet
- . ⑩ Categorize major ways in which the Internet is being used
- . ⑩ Describe the basic services of ISPs including dial-up, Web hosting, and secure socket connections
- . ⑩ Conduct an analysis of your needs for online service and select an ISP with services that match your needs
- . ⑩ Select a domain name and create a unique presence on the Internet

Performance Indicators

- . ⑩ Current status and future development trends on the Internet can be clearly described
- . ⑩ Primary ways in which the Internet is being used can be categorized
- . ⑩ Basic services of ISPs can be clearly described
- . ⑩ A thorough analysis of an individual's needs for online service can be performed and appropriate service can be selected
- . ⑩ An appropriate domain name can be selected and registered

Module Topics

- . ⑩ 1.1 What are the Internet and the World Wide Web?
- . ⑩ 1.2 Quick Tour
- . ⑩ 1.3 How the Internet Works
- . ⑩ 1.4 Choosing an ISP

Module 2: Internet Communications

Learning Outcomes

- . ⑩ Describe what is meant by "client/server architecture"
- . ⑩ Describe and explain why you would use Telnet
- . ⑩ Describe the three kinds of electronic mail systems—server-side, client-side, and hybrid, and in what situations they are most useful
- . ⑩ Set up and use Pine (server-side email software) and Eudora (client-side email software) to send electronic messages and identify differences between the two kinds of email services
- . ⑩ Distinguish between mailing lists and newsgroups
- . ⑩ Explain netiquette and its importance to email communication on the Internet
- . ⑩ Describe privacy considerations you must make when sending email messages on the Internet

Performance Indicators

- . ⑩ "Client/server architecture" can be clearly defined
- . ⑩ Appropriate uses of Telnet can be clearly described
- . ⑩ Three types of electronic mail systems can be appropriately described
- . ⑩ Client and server-side email software can be appropriately installed and differences between them can be clearly delineated
- . ⑩ Differences between mailing lists and newsgroups can be thoroughly discussed
- . ⑩ Internet etiquette and its importance in email communication can be thoroughly explained
- . ⑩ Privacy issues when sending email messages on the Internet can be clearly described

Module Topics

- . ⑩ 2.1 Client/Server Architecture
- . ⑩ 2.2 Telnet
- . ⑩ 2.3 Electronic Mail
- . ⑩ 2.4 Mailing Lists, News Groups, Netiquette, and Privacy

Learning Outcomes

- . ⑩ Use basic features of Microsoft Internet Explorer and Netscape Communicator
- . ⑩ Compare the relative strengths of Internet Explorer and Communicator
- . ⑩ Describe the distinction between a directory service and search engine
- . ⑩ Conduct an information search using both a directory service and a search engine
- . ⑩ Utilize good searching techniques to find information on the Internet

Performance Indicators

- . ⑩ Basic features of Microsoft Internet Explorer and Netscape Communicator can be appropriately used
- . ⑩ Internet Explorer and Netscape Communicator can be appropriately

compared and contrasted

- . ⑩ Differences between a search engine and a directory service can be clearly delineated
- . ⑩ Efficient search techniques can be used to find appropriate information on the Internet

Module Topics

- . ⑩ 3.1 Internet Browsers
- . ⑩ 3.2 Using Web Browsers for Web Authoring
- . ⑩ 3.3 Searching the Web: Resources and Strategies
- . ⑩ 3.4 Guidelines for Effective Internet Searches
- . ⑩ 3.5 Power Searching Tools

Module 4: Web Authoring Basics

Learning Outcomes

- . ⑩ Describe how Internet Explorer and Netscape browsers display a document on the World Wide Web
- . ⑩ Use HTML (Hypertext Markup Language) to build a Web-ready document that includes images, internal links, and external links
- . ⑩ Describe and demonstrate the difference between absolute, root, relative, and internal referencing
- . ⑩ Build or edit a Web document using server-side scripts written in Pico
- . ⑩ Describe the role and parts of a URL (Uniform Resource Locator) on the World Wide Web

Performance Indicators

- . ⑩ A clear explanation of how Internet Explorer and Netscape browsers display documents on the World Wide Web can be given
- . ⑩ HTML can be used to build a Web-ready document containing images and internal and external links
- . ⑩ Differences between absolute, root, relative, and internal referencing can be clearly discussed
- . ⑩ Web documents can be built and edited using server-side scripts written in Pico
- . ⑩ Components of a Uniform Resource Locator can be clearly described

Module Topics

- . ⑩ 4.1 Web Technologies
- . ⑩ 4.2 HTML
- . ⑩ 4.3 Writing HTML with a Text Editor
- . ⑩ 4.4 Server-side Authoring Using Pico
- . ⑩ 4.5 Using Pico and UNIX to Write Server-side Scripts

Learning Outcomes

- . ⑩ Download, install, and use WS_FTP (PC) or Fetch (Mac) to upload or

download files from one computer to another on a network

- . ⑩ Download, install, and use WinZip (PC) or ZipIt (Mac) to compress files for transfer on the Internet
- . ⑩ Utilize UNIX commands to move, copy, delete, and rename files or directories on a server
- . ⑩ Apply the UNIX chmod command to enable permissions for the site owner, network group, and other visitors to read, write, and execute files on a server
- . ⑩ Describe the difference between using encryption and CyberCash for carrying out online Internet financial transactions
- . ⑩ Describe a firewall and how it protects a network from hackers

Performance Indicators

- . ⑩ Appropriate FTP software can be downloaded and installed to upload and download files across a network
- . ⑩ Appropriate file compression software can be downloaded and installed to compress files for transfer on the Internet
- . ⑩ Appropriate UNIX commands can be used to perform file maintenance operations on a server
- . ⑩ Appropriate UNIX commands can be used to assign permissions on a network server
- . ⑩ Differences between encryption and CyberCash for financial transactions over the Internet can be clearly delineated
- . ⑩ Concept and functions of firewalls can be clearly described

Module Topics

- . ⑩ 5.1 Downloading and Uploading Files
- . ⑩ 5.2 What is Compression?
- . ⑩ 5.3 What is FTP?
- . ⑩ 5.4 Operating Systems
- . ⑩ 5.5 Online Security, Protection, and Privacy

Module 6: Imaging

Learning Outcomes

- . ⑩ Determine image file sizes on Windows and Mac computers
- . ⑩ Distinguish between GIF and JPEG image file formats and when to use them
- . ⑩ Create and manipulate GIF and JPEG images, and use them appropriately within a Web document
- . ⑩ Describe the difference between a client-side image map and server-side image map
- . ⑩ Create three kinds of image maps and incorporate associated HTML, including hyperlinks, within a Web page
- . ⑩ Describe the PNG image file format and why it is favorably considered as a graphic file format for the Internet
- . ⑩ Describe a Flash animation and why it is an efficient way to create

animated graphics on the Web

Performance Indicators

- . ⑩ Image file sizes on Windows and Mac computers can be correctly determined
- . ⑩ Appropriate use of GIF and JPEG file formats can be determined
- . ⑩ GIF and JPEG images can be appropriately used in a Web document
- . ⑩ Differences between client and server-side image maps can be clearly explained
- . ⑩ Three types of image maps can be successfully created and incorporated into a Web page
- . ⑩ Advantages of using the PNG file format for images on Web pages can be clearly explained
- . ⑩ Reasons why Flash animation is efficient for creating animation on Web pages can be clearly described

Module Topics

- . ⑩ 6.1 Using Images on the Web
- . ⑩ 6.2 What are the Differences between GIF and JPEG?
- . ⑩ 6.3 Image Maps
- . ⑩ 6.4 Color Palettes—Adaptive and Web-Safe
- . ⑩ 6.5 Other Graphic Formats—PNG, Animated GIFs, and Flash Animation

Learning Outcomes

- . ⑩ Use tables to position text and graphics on a Web page
- . ⑩ Describe and utilize nested tables
- . ⑩ Use cell padding and cell spacing attributes to format cells and cell content within a table
- . ⑩ Use tiling to create a custom background on a Web page
- . ⑩ Create a transparent pixel graphic with height and weight attributes to precisely position elements on a Web page
- . ⑩ Build a frameset document in HTML using the <FRAMESET> and <FRAME> tags, and target content to specified screen sections
- . ⑩ Describe a floating frame, and how it is used
- . ⑩ Describe the function of a cascading style sheet
- . ⑩ Describe the difference between an Internal Style Sheet and External Style Sheet

Performance Indicators

- . ⑩ Tables and nested tables can be appropriately used to position text and graphics on a Web page
- . ⑩ Cell padding and cell spacing can be used to appropriately format cells and cell content within a table
- . ⑩ Tiling can be used to create a custom background on a Web page
- . ⑩ Transparent graphics with height and width attributes appropriately set can be used in a Web page

- . ⑩ HTML documents can be appropriately built using frames and floating frames
- . ⑩ Functions of cascading style sheets can be clearly described
- . ⑩ Differences between internal style sheets and external style sheets can be clearly described

Module Topics

- . ⑩ 7.1 Layout with Tables
- . ⑩ 7.2 Layout with Frames
- . ⑩ 7.3 Layout with Cascading Style Sheets

Module 8: Creating Web Interactivity using Client-side Scripting

Learning Outcomes

- . ⑩ Articulate how interactivity can work to achieve the goals of a Web site
- . ⑩ Describe CGI (Common Gateway Interface) and implement a simple CGI-based form on a Web page
- . ⑩ Describe JavaScript, how it differs from Java, and its primary role in Web authoring
- . ⑩ Build a JavaScript rollover on a Web page
- . ⑩ Describe Java, its main features, and how it applies to Web authoring
- . ⑩ Implement and demonstrate how a Java applet is used on a Web page
- . ⑩ Describe situations where each of these technologies could be used to create interactive experiences on the Web

Performance Indicators

- . ⑩ Advantages of creating interactive Web pages can be clearly articulated
- . ⑩ CGI (Common Gateway Interface) can be clearly defined and implemented on a Web page
- . ⑩ Differences between JavaScript and Java can be clearly described
- . ⑩ A Web page using a JavaScript rollover can be successfully created
- . ⑩ Java applets can be successfully implemented on a Web page
- . ⑩ Appropriate use of CGI, client-side scripting and Java applets can be clearly defined

Module Topics

- . ⑩ 8.1 The Value of Interactivity
- . ⑩ 8.2 Creating the Interactivity
- . ⑩ 8.3 JavaScript
- . ⑩ 8.4 Java and Java Applets

Learning Outcomes

- . ⑩ Explain the differences between VBScript and JavaScript
- . ⑩ Explain the basic structure of a VBScript block
- . ⑩ Describe the difference between placing a script block in the <head> section vs. the <body> section of a Web page

- . ⑩ Describe the differences between Visual Basic and VBScript
- . ⑩ Demonstrate use of variables, sequence structures, branches, and looping structures using VBScript
- . ⑩ Create and test simple event procedures for controls placed on a form in a Web page

Performance Indicators

- . ⑩ Differences between VBScript and JavaScript can be clearly explained
- . ⑩ Basic structure of a VBScript script block can be clearly explained
- . ⑩ Runtime differences between scripts placed in the <head> section vs. script placed in the <body> can be clearly described
- . ⑩ Differences between Visual Basic and VBScript can be clearly described
- . ⑩ Variables, sequence structures, branches, and looping structures can be appropriately demonstrated using VBScript
- . ⑩ Simple event procedures for HTML controls placed on a form in a Web page can be appropriately created and tested

Module Topics

- . ⑩ 9.1 Client-side Scripting using VBScript vs. JavaScript
- . ⑩ 9.2 Creating Script Blocks using VBScript
- . ⑩ 9.3 VBScript Essentials—Differences between Visual Basic and VBScript
- . ⑩ 9.4 Implementing Programming Logic Structures using VBScript
- . ⑩ 9.5 Writing Event Procedures

Module 10: Using VBScript for DHTML

Learning Outcomes

- . ⑩ Define Dynamic HTML and its purpose
- . ⑩ Explain “element naming” and its purpose
- . ⑩ Define the major components of the Document Object Model and give examples of some methods, properties, and events
- . ⑩ Explain what is meant by the term “event handling”
- . ⑩ Describe event bubbling
- . ⑩ Create a simple event procedure using VBScript to modify the style settings of an element
- . ⑩ Use DIV and SPAN tags to create programmable areas in a Web page
- . ⑩ Explain the use of the “innerHTML” and “outerHTML” properties
- . ⑩ Write an event procedure to alter the content of an element at runtime

Performance Indicators

- . ⑩ Purpose and concept of Dynamic HTML can be clearly defined
- . ⑩ Purpose and advantages of “element naming” can be clearly explained
- . ⑩ Major components of the Document Object Model can be clearly defined and examples of methods, properties and events given
- . ⑩ Term “event handling” can be clearly explained
- . ⑩ The concept of “event bubbling” can be clearly described

- . ⑩ Simple event procedures to modify the style property/settings of an element at runtime can be appropriately created
- . ⑩ DIV and SPAN tags can be appropriately used to create programmable sections of a Web page
- . ⑩ Use of “innerHTML” and “outerHTML” properties can be clearly explained
- . ⑩ Event procedures to alter Web page content at runtime can be appropriately written

Module Topics

- . ⑩ 10.1 Understanding DHTML
- . ⑩ 10.2 Writing Simple Event Procedures for HTML Elements
- . ⑩ 10.3 Using DIV and SPAN
- . ⑩ 10.4 Using DHML to Dynamically Alter Content and Style at Runtime

**COURSE 2: WEB PAGE
PROGRAMMING**

USING JAVASCRIPT

Overview..... 1

Module 1: Introduction to JavaScript 5

Module 2: JavaScript's Object Model 6

Module 3: Variables and Conditional Statements 7

Module 4: Windows and Frames
..... 8

**Module 5: Arrays and
Loops..... 9**

Module 6: JavaScript Cookies
..... 10

Module 7: Detecting the Browser
..... 11

Module 8: Dynamic HTML
..... 12

This course is an introduction to client-side scripting using the popular programming language JavaScript. JavaScript is a powerful programming tool used to create interactivity, animation, and flexibility in Web pages. In addition to basic JavaScript syntax, this course focuses on programming structure and concepts. Students will learn about objects, cookies, layers, and how to use DHTML to create animation. Upon completion of this course, the student will have learned basic programming skills and will also have the capability to add very functional and creative effects to their Web projects.

Instructional Goals

- . ⑩ Explain the difference between a traditional programming language and a scripting language
- . ⑩ Explain the concept of an “event” and how it is used in Web page scripting
- . ⑩ Use the JavaScript Object Model to access forms, create layers, access

- values of controls, detect the browser, and produce output on the user's browser screen
- . ⑩ Create JavaScript code using variables, conditional statements, and other commonly used programming structures
- . ⑩ Manipulate windows and frames using JavaScript coding
- . ⑩ Use JavaScript and DHTML to control styles, animation, and keyboard input

Course Prerequisites

- ⑩ Strong knowledge of HTML and Web page concepts and terms

Learning Outcomes

- . ⑩ Explain the difference between a scripting language and a traditional programming language and the difference between a compiled and an interpreted language
- . ⑩ Define the following terms and describe their roles: cross-platform, platform specific, algorithm, pseudocode, function call, function block, and argument
- . ⑩ List the advantages/disadvantages of using a scripting language over a compiled language
- . ⑩ Explain the role of events in a user-driven interactive environment and how event handlers work in conjunction with events
- . ⑩ Explain why functions are used in programming languages
- . ⑩ Define images array and describe how indexing works in an array structure like images
- . ⑩ Describe the role of the src (source) attribute for images referenced by JavaScript
- . ⑩ Explain the relevance of using objects in a programming environment and the terms property and method
- . ⑩ Explain the concept of object hierarchy and how it is used to reference objects
- . ⑩ Define the term base object, name the base object in the Document Object Model, and list two of its descendant objects
- . ⑩ Explain the role of the write method, the object that contains it, and how to reference it through the Document Object Model
- . ⑩ Explain the significance of indexing and the checked property when accessing a radio button with JavaScript
- . ⑩ Define the difference between defining and assigning a value to a variable
- . ⑩ Define variable scope
- . ⑩ Define the difference between local and global variables with respect to functions
- . ⑩ List four variable data types and what kind of values they may be used for
- . ⑩ Define expression, operator and operand
- . ⑩ Create an expression using at least one operator and explain what the result would be
- . ⑩ Define the role of comparison operators in a conditional statement
- . ⑩ Define and use properties, methods, and event handlers that relate to the

Web browser window

Learning Outcomes, *continued*

- . ⑩ Create an HTML frame that is capable of changing information in the other frames of the frameset
- . ⑩ Define a function that requires arguments and pass values to the function in a function call
- . ⑩ Create a remote window and a handle to the new window
- . ⑩ Update information in a remote window from the window that created it
- . ⑩ Define array and explain how array indexing works
- . ⑩ Create a simple array of data values and access individual values through array indexing
- . ⑩ Define multi-dimensional array and explain how the term applies to data tables
- . ⑩ Explain the significance of the initial value of a looping variable and how it relates to the exit conditional statement
- . ⑩ Use a for loop to access each of the values of an array in order
- . ⑩ Explain the process by which the length of a data array can be retrieved and its significance to a looping structure
- . ⑩ Define the term "*cookie*" and its relation to client-server computing and HTTP
- . ⑩ Use JavaScript to create a cookie file that is stored on the client hard drive
- . ⑩ Manipulate the date object and its methods to query the system clock and set future dates
- . ⑩ Use JavaScript to read and parse a cookie file for its stored values
- . ⑩ Define the JavaScript properties that return the client's browser name, browser version, and platform
- . ⑩ Isolate the browser name and version from a string that defines them
- . ⑩ Write a generic browser-detection function that can be used on a variety of Web pages
- . ⑩ Use cross-platform cascading style sheet features to design a custom style for a Web page
- . ⑩ Manipulate advanced features of cascading style sheets like visibility, z-indexing, and absolute positioning
- . ⑩ Use JavaScript to change advanced features of cascading style sheets through the Document Object Model
- . ⑩ Create a simple DHTML animation with layers and moving page elements
- . ⑩ Create an animation loop that is set to update itself at regular intervals
- . ⑩ Define persistence of vision
- . ⑩ Capture keyboard events and interpret the key code of the event

Performance Indicators

- . ⑩ Differences between traditional programming languages and scripting languages can be clearly explained
- . ⑩ Event-driven programming and how it is implemented using JavaScript can be clearly explained

- . ⑩ JavaScript code can be written by using appropriate variable declarations, conditional statements, loops, and arrays

Performance Indicators, *continued*

- . ⑩ Dynamic output can be created by using windows and frames
- . ⑩ DHTML and JavaScript can be appropriately used to control styles, animation, and keyboard input

Module Titles

1. 1. Introduction to JavaScript
2. 2. JavaScript's Object Model
3. 3. Variables and Conditional Statements
4. 4. Windows and Frames
5. 5. Arrays and Loops
6. 6. JavaScript Cookies
7. 7. Detecting the Browser
8. 8. Dynamic HTML

Learning Outcomes

- . ⑩ Explain the difference between a scripting language and a traditional programming language, and the difference between a compiled and an interpreted language
- . ⑩ Define the following terms and their roles: cross-platform, platform specific, algorithm, pseudocode, function call, function block, and argument
- . ⑩ List the advantages/disadvantages of using a scripting language over a compiled language
- . ⑩ Explain the role of events in a user-driven interactive environment and how event handlers work in conjunction with events
- . ⑩ Explain why functions are used in programming languages
- . ⑩ Define images array
- . ⑩ Describe how indexing works in an array structure
- . ⑩ Describe the role of the src (source) attribute for images referenced by JavaScript

Performance Indicators

- . ⑩ Differences between traditional programming and scripting languages, and differences between compiled and interpreted languages can be clearly explained
- . ⑩ Fundamental programming terms can be clearly defined
- . ⑩ Advantages and disadvantages of using a scripting language over a compiled language can be identified
- . ⑩ The role of events in a user-driven interactive environment can be clearly explained
- . ⑩ The purpose of using functions in programming can be clearly explained
- . ⑩ The function of an array and array indexing can be thoroughly defined
- . ⑩ The role of the source attribute for images referenced by JavaScript can be appropriately described

Module Topics

- . ⑩ 1.1 Scripting Languages
- . ⑩ 2.1 Algorithm: Button Rollovers
- . ⑩ 3.1 Script's Event Model
- . ⑩ 4.1 JavaScript Functions
- . ⑩ 5.1 The JavaScript Images Array
- . ⑩ 6.1 Source Code

Learning Outcomes

- . ⑩ Explain the relevance of using objects in a programming environment and the terms property and method
- . ⑩ Explain the concept of object hierarchy and how it is used to reference objects
- . ⑩ Explain the role of the write method, the object that contains it, and how to reference it through the Document Object Model
- . ⑩ Explain the significance of indexing using controls such as the radio button
- . ⑩ Use the checked property when accessing a radio button with JavaScript

Performance Indicators

- . ⑩ The terms objects, properties, and methods can be clearly defined
- . ⑩ The concepts of object hierarchy and its use can be clearly explained
- . ⑩ The write method and its usage can be clearly explained
- . ⑩ The significance of indexing when accessing grouped controls can be clearly explained
- . ⑩ The checked property of a radio button can be appropriately used with JavaScript

Module Topics

- . ⑩ 2.1 The Document Object Model
- . ⑩ 2.2 Algorithm: Interactive FORM Story
- . ⑩ 2.3 Writing Text to an HTML Page
- . ⑩ 2.4 Other FORM Elements
- . ⑩ 2.5 Source Code

Learning Outcomes

- . ⑩ Define the difference between declaring and assigning a value to a variable
- . ⑩ Define the concept of variable scope
- . ⑩ Define the difference between local and global variables with respect to functions
- . ⑩ List four variable data types and what kind of values they may hold
- . ⑩ Define the terms expression, operator, and operand
- . ⑩ Create an expression using at least one operator and explain what the result would be

- . ⑩ Define the role of logical operators in a conditional statement

Performance Indicators

- . ⑩ Differences between declaring and assigning values to variables can be clearly delineated
- . ⑩ The concept of variable scope can be clearly defined
- . ⑩ The differences between local and global variables can be clearly defined
- . ⑩ Four commonly used variable data types and examples of appropriate values can be listed
- . ⑩ An expression using at least one operator can be appropriately created and analyzed
- . ⑩ The role of logical operators in a conditional statement can be clearly defined

Module Topics

- . ⑩ 3.1 Declaring Variables
- . ⑩ 3.2 Algorithm: Client-side Conversion Calculator
- . ⑩ 3.3 Data Types
- . ⑩ 3.4 Operators and Operands
- . ⑩ 3.5 Conditional Statements
- . ⑩ 3.6 Source Code

Learning Outcomes

- . ⑩ Define and use properties, methods, and event handlers that relate to the Web browser window
- . ⑩ Create an HTML frame that is capable of changing information in the other frames of the frameset
- . ⑩ Define a function that requires arguments and pass values to the function in a function call
- . ⑩ Create a remote window and a handle to the new window
- . ⑩ Update information in a remote window from the window that created it

Performance Indicators

- . ⑩ Properties, methods, and events of the Web browser window object can be clearly defined
- . ⑩ JavaScript code can be written to exchange information between the frames of a frameset
- . ⑩ A JavaScript function can be written that passes arguments and returns a value
- . ⑩ JavaScript code can be written to create a remote window and control it programmatically

Module Topics

- . ⑩ 4.1 Understanding Browser Windows
- . ⑩ 4.2 Creating Algorithms Using the Browser Window
- . ⑩ 4.3 Working With Frames

- . ⑩ 4.4 Working With Remote Windows

Learning Outcomes

- . ⑩ Define the term array and explain how array indexing works
- . ⑩ Create a simple array of data values and access individual values through array indexing
- . ⑩ Define the concept of a multi-dimensional array and how it applies to data tables
- . ⑩ Explain the significance of the initial value of a looping variable and how it relates to the exit conditional statement
- . ⑩ Use a for loop to access each of the values of an array in order
- . ⑩ Explain the process by which the length of a data array can be retrieved and its significance to a looping structure

Performance Indicators

- . ⑩ Concept and function of an array can be clearly defined
- . ⑩ Simple arrays are created and code can be written to access the data values
- . ⑩ The concept of a multi-dimensional array and its relationship to data tables can be clearly explained
- . ⑩ Array values can be accessed using looping algorithms

Module Topics

- . ⑩ 5.1 Arrays
- . ⑩ 5.2 Algorithm: Word Guessing Game
- . ⑩ 5.3 Loop Structures
- . ⑩ 5.4 Arrays and Loops
- . ⑩ 5.5 String Manipulation
- . ⑩ 5.6 Source Code

Learning Outcomes

- . ⑩ Define the term "*cookie*" and its relation to client-server computing and HTTP
- . ⑩ Use JavaScript to create a cookie file that is stored on the client hard drive
- . ⑩ Manipulate the date object and its methods to query the system clock and set future dates
- . ⑩ Use JavaScript to read and parse a cookie file for its stored values

Performance Indicators

- . ⑩ Cookies and their relationship to client-server computing and HTTP can be clearly defined
- . ⑩ Cookies can be appropriately stored on the client hard drive
- . ⑩ Query of the system clock and manipulating the date object using JavaScript can be achieved
- . ⑩ Reading and parsing a cookie for its values using JavaScript can be demonstrated

Module Topics

- . ⑩ 6.1 JavaScript Cookies
- . ⑩ 6.2 Algorithm: Cookies Page
- . ⑩ 6.3 The Date Object
- . ⑩ 6.4 Setting Cookies with JavaScript
- . ⑩ 6.5 Getting Cookies with JavaScript
- . ⑩ 6.6 Source Code

Learning Outcomes

- . ⑩ Define the JavaScript properties that return the client's browser name, browser version, and platform
- . ⑩ Isolate the browser name and version from a string that defines them
- . ⑩ Write a generic browser-detection function that can be used on a variety of Web pages
- . ⑩ Use cross-platform cascading style sheet features to design a custom style for a Web page

Performance Indicators

- . ⑩ JavaScript properties to return the client browser name, version, and platform can be clearly defined
- . ⑩ The browser name and version can be isolated from a string
- . ⑩ A generic browser-detection function that can be used on a variety of Web pages can be written
- . ⑩ A cross-platform cascading style sheet can be used to design a custom style for a Web page

Module Topics

- . ⑩ 7.1 The Navigator Object
- . ⑩ 7.2 Algorithm: Browser Detection Page
- . ⑩ 7.3 Generic Detection Function
- . ⑩ 7.4 Cascading Style Sheets
- . ⑩ 7.5 Source Code

Learning Outcomes

- . ⑩ Demonstrate advanced features of cascading style sheets like visibility, z-indexing, and absolute positioning
- . ⑩ Use JavaScript to change advanced features of cascading style sheets through the Document Object Model
- . ⑩ Create a simple DHTML animation with layers and moving page elements
- . ⑩ Create an animation loop that is set to update itself at regular intervals
- . ⑩ Define persistence of vision
- . ⑩ Capture keyboard events and interpret the key code of the event

Performance Indicators

- . ⑩ Advanced features of cascading style sheets such as visibility, z-indexing, and absolute positioning can be appropriately demonstrated

- . ⑩ JavaScript can be used to change advanced features of cascading style sheets through the Document Object Model
- . ⑩ Simple animations with layers and moving page elements can be created using DHTML
- . ⑩ The concept of persistence of vision can be clearly defined
- . ⑩ Keyboard events can be appropriately used to interpret key codes

Module Topics

- . ⑩ 8.1 Advanced Features of CSS
- . ⑩ 8.2 Algorithm: Tanker Animation
- . ⑩ 8.3 Controlling DHTML with JavaScript
- . ⑩ 8.4 Animation
- . ⑩ 8.5 Keyboard Input
- . ⑩ 8.6 Source Code

COURSE 3: DIGITAL IMAGING FOUNDATIONS

Overview.....
 1

Module 1: Getting Started With Digital Imaging
 5

Module 2: Getting Started With Photoshop
 6

Module 3: Using Selections and Masks
 7

Module 4: Using Layers

..... 8

Module 5: Using Colors and Painting Tools
..... 9

Module 6: Adjusting Images
..... 10

Module 7: Effects, Filters, and Useful Techniques..... 11

Module 8: Preparing Images for the World Wide Web
..... 12

An understanding of the concepts, terms, and techniques of digital imaging is very important for the Web developer. This course thoroughly covers the creation and modification of digital images using Adobe Photoshop. The course progresses through introductory techniques for digital imaging through advanced techniques such as layering, creating, and editing images using selections and masks, and applying filters. Preparation and storage of images for the World Wide Web are also covered.

Instructional Goals

- . ⑩ Describe key imaging concepts and terms
- . ⑩ Demonstrate the use of basic Photoshop tools and image manipulation techniques
- . ⑩ Edit images using selections and masks
- . ⑩ Enhance images using colors, painting tools, and effects
- . ⑩ Prepare images for efficient use on the World Wide Web

Course Prerequisites

- . ⑩ Knowledge of basic file processing techniques (opening, saving, etc.)
- . ⑩ Experience creating Web pages using HTML

Learning Outcomes

- . ⑩ Describe the reasons for using images in information design
- . ⑩ Describe the features of the client/producer relationship
- . ⑩ Describe the difference between raster and vector images
- . ⑩ Define and describe these basic imaging concepts: bitmap, vector, saturation, indexing, resolution, color depth, luminance, aliasing, and anti-aliasing
- . ⑩ Describe the most common image file formats and their uses
- . ⑩ Create models for RGB and CMYK color spaces
- . ⑩ Set preferences and manage memory allocation in Photoshop
- . ⑩ Open, close, crop, resize, duplicate, and save an image in Photoshop
- . ⑩ Describe four sources for obtaining images for use in Photoshop
- . ⑩ Describe copyright considerations when looking for a source image
- . ⑩ Utilize navigation tools in Photoshop for scrolling, dragging, and zooming within a screen
- . ⑩ Demonstrate the use of the toolbox and its features
- . ⑩ Demonstrate the access and use of tool palettes, and their coordination with the toolbox
- . ⑩ Utilize primary keyboard shortcut commands and describe their importance for working in Photoshop
- . ⑩ Create a list of image specifications based on information derived in Photoshop, including file size, dimensions, measurements, position, color data, and resolution
- . ⑩ Explain why making selections and creating masks are important techniques for creating or modifying images
- . ⑩ Describe the application of masks and selections in composing images
- . ⑩ Describe the difference between selections and masks
- . ⑩ Use the Marquee, Lasso, and Magic Wand tools to select portions of an image
- . ⑩ Modify a selection by using the following: save, load, move, duplicate, add to, subtract from, transform, and feather
- . ⑩ Use the following masking tools to isolate and modify selected portions of an image: Quick Mask, Magic Eraser, and Extract Image command
- . ⑩ Convert a selection to a mask, and a mask to a selection
- . ⑩ Describe the importance of Photoshop layers in image composition
- . ⑩ Utilize primary features of the Layers Palette: creating, hiding/showing, linking, stacking, moving, copying, naming, and flattening

Learning Outcomes, *continued*

- . ⑩ Utilize advanced features for creating Layers effects: Opacity, Gradient, Magic Eraser, effects, and blends
- . ⑩ Create and position text in Photoshop
- . ⑩ Define the following printing terms: kerning, leading, tracking, and font
- . ⑩ Describe the following principles of good image composition and how

they are applied: Unity, Balance, Contrast, and Alignment

- . ⑩ Set preferences and manage memory allocation in Photoshop
- . ⑩ Compose a presentation image, consisting of multiple layers of graphics, that follows principles of good composition
- . ⑩ Discuss how traditional art techniques have influenced digital imaging technology
- . ⑩ Select colors using multiple methods in Photoshop
- . ⑩ Utilize the primary painting tools to create and edit images—Paintbrush, Airbrush, Line Tool, Pencil Tool, Smudge Tool, Gradient, Eraser, and Fill Bucket
- . ⑩ Create a color scheme
- . ⑩ Describe how transparency, feathering, gradients, and texture filters can be used to modify a digital image
- . ⑩ Discuss how old black and white photographs are enhanced using traditional photo-retouching techniques
- . ⑩ Describe the basic procedure and principles of scanning a photograph or other image
- . ⑩ Identify problem areas in old photographs and use the appropriate Photoshop tool to correct them from the following toolset: Hue/Saturation Adjustment, Brightness/Contrast Adjustment, Levels, Rubber Stamp Tool, Unsharp Mask Filter, Dodge Tool, Burn Tool, and Sponge Tool
- . ⑩ Utilize Adjustment Layers to create editable changes in a layer
- . ⑩ Utilize the Replace Color Tool
- . ⑩ Describe the origins of presentation graphics and the use of standard visual effects
- . ⑩ Describe an image filter and demonstrate how it is accessed and applied in Photoshop
- . ⑩ Utilize Photoshop filters and layer effects and describe situations where they are appropriately used
- . ⑩ Demonstrate how text can be enhanced using filters, layer effects, and the Transform tool
- . ⑩ Demonstrate five techniques for improving work efficiency in Photoshop
- . ⑩ Describe the origins of Web graphics and their applications for the World Wide Web
- . ⑩ Describe the differences among GIF, JPEG, and PNG image file formats and situations in which each format should be used
- . ⑩ Demonstrate primary strategies for optimizing images for use on the World Wide Web: color reduction, compression, and sizing
- . ⑩ Create a simple GIF animation for display on a Web page
- . ⑩ Utilize a text processor and HTML to insert a Web-ready image into a Web page for display on an Internet browser

Performance Indicators

- . ⑩ Key imaging concepts and terms can be clearly described
- . ⑩ Use of basic Photoshop tools and image manipulation techniques can be appropriately demonstrated
- . ⑩ Images can be appropriately edited using selections and masks

- . ⑩ Images can be enhanced using colors, painting tools, and special effects
- . ⑩ Images can be prepared for most efficient use on the World Wide Web

Module Titles

1. 1. Getting Started With Digital Imaging
2. 2. Getting Started With Photoshop
3. 3. Using Selections and Masks
4. 4. Using Layers
5. 5. Using Colors and Painting Tools
6. 6. Adjusting Images
7. 7. Effects, Filters, and Useful Techniques
8. 8. Preparing Images for the World Wide Web

Learning Outcomes

- . ⑩ Describe the reasons for using images in information design
- . ⑩ Describe the features of the client/producer relationship
- . ⑩ Describe the difference between raster and vector images
- . ⑩ Define and describe these basic imaging concepts: bitmap, vector, saturation, indexing, resolution, color depth, luminance, aliasing, and anti-aliasing
- . ⑩ Describe the most common image file formats and their uses
- . ⑩ Create models for RGB and CMYK color spaces
- . ⑩ Set preferences and manage memory allocation in Photoshop
- . ⑩ Open, close, crop, resize, duplicate, and save an image in Photoshop

Performance Indicators

- . ⑩ Reasons for using images in information design can be thoroughly described
- . ⑩ Features of the client/producer relationship can be thoroughly described
- . ⑩ Differences between raster and vector images can be clearly delineated
- . ⑩ Bitmaps, vectors, saturation, indexing, resolution, color depth, luminance, aliasing, and anti-aliasing can be clearly defined
- . ⑩ Most common image file formats and their uses can be clearly described
- . ⑩ Models for RGB and CMYK color spaces can be appropriately created
- . ⑩ Preferences and memory management can be appropriately set in Photoshop
- . ⑩ Images can be opened, closed, cropped, resized, duplicated, and saved in Photoshop

Module Topics

- . ⑩ 1.1 Why Are Images Important?
- . ⑩ 1.2 The Client-Designer Relationship
- . ⑩ 1.3 Understanding Imaging Concepts
- . ⑩ 1.4 Understanding Color
- . ⑩ 1.5 Setting Preferences and Memory Management
- . ⑩ 1.6 Case Study: Getting Started With Images in Photoshop

Learning Outcomes

- . ⑩ Describe four sources for obtaining images for use in Photoshop
- . ⑩ Describe copyright considerations when looking for a source image
- . ⑩ Utilize navigation tools in Photoshop for scrolling, dragging, and zooming within a screen
- . ⑩ Demonstrate the use of the toolbox and its features
- . ⑩ Demonstrate the access and use of tool palettes, and their coordination with the Toolbox
- . ⑩ Utilize primary keyboard shortcut commands and describe their importance for working in Photoshop
- . ⑩ Create a list of image specifications based on information derived in Photoshop, including file size, dimensions, measurements, position, color data, and resolution

Performance Indicators

- . ⑩ Four sources for obtaining images for use in Photoshop can be clearly described
- . ⑩ Copyright considerations can be clearly described as they apply to searching for images
- . ⑩ Navigation tools in Photoshop can be appropriately utilized
- . ⑩ Appropriate use of the toolbox and its features can be demonstrated
- . ⑩ Primary keyboard shortcut commands can be appropriately used
- . ⑩ A list of image specifications can be created from information derived in Photoshop

Module Topics

- . ⑩ 2.1 Acquiring Source Materials
- . ⑩ 2.2 Viewing Tools
- . ⑩ 2.3 Using the Toolbox and Tool Palettes
- . ⑩ 2.4 Keyboard Shortcuts
- . ⑩ 2.5 Case Study: Obtaining Information About Images in Photoshop

Learning Outcomes

- . ⑩ Explain why making selections and creating masks are important techniques for creating or modifying images
- . ⑩ Describe the application of masks and selections in composing images
- . ⑩ Describe the difference between selections and masks
- . ⑩ Use the Marquee, Lasso, and Magic Wand tools to select portions of an image
- . ⑩ Modify a selection by using the following: save, load, move, duplicate, add to, subtract from, transform, and feather
- . ⑩ Use the following masking tools to isolate and modify selected portions of an image: Quick Mask, Magic Eraser, and Extract Image command
- . ⑩ Convert a selection to a mask, and a mask to a selection

Performance Indicators

- . ⑩ The importance of making selections and creating masks can be clearly explained
- . ⑩ The application of masks and selections in composing images can be clearly described
- . ⑩ Differences between selections and masks can be clearly described
- . ⑩ The marquee, lasso, and magic wand tools can be appropriately used to select portions of an image
- . ⑩ A selection can be successfully modified using save, load, move, duplicate, add to, subtract from, transform, and feather
- . ⑩ Selected portions of an image can be appropriately isolated and modified
- . ⑩ A selection can be successfully converted to a mask and vice versa

Module Topics

- . ⑩ 3.1 Why Are Selections and Quick Masks Important Image Editing Tools?
- . ⑩ 3.2 Using Selection and Masking Techniques in Image Composition
- . ⑩ 3.3 Selection Tools: Lassos, Marquees, and Magic Wand
- . ⑩ 3.4 Modifying Selections: Adding, Subtracting, and Transforming
- . ⑩ 3.5 Masking Tools: Quick Mask, Magic Eraser, and Extract Image Command
- . ⑩ 3.6 Case Study: Modifying Images with Selections and Masks

Learning Outcomes

- . ⑩ Describe the importance of Photoshop layers in image composition
- . ⑩ Utilize primary features of the Layers Palette: creating, hiding/showing, linking, stacking, moving, copying, naming, and flattening
- . ⑩ Utilize advanced features for creating Layers effects: Opacity, Gradient, Magic Eraser, effects, and blends
- . ⑩ Create and position text in Photoshop
- . ⑩ Define the following printing terms: kerning, leading, tracking, and font
- . ⑩ Describe the following principles of good image composition and how they are applied: Unity, Balance, Contrast, and Alignment
- . ⑩ Set preferences and manage memory allocation in Photoshop
- . ⑩ Compose a presentation image, consisting of multiple layers of graphics, that follows principles of good composition

Performance Indicators

- . ⑩ The importance of Photoshop layers in image composition can be clearly described
- . ⑩ The primary features of the Layers Palette can be appropriately utilized
- . ⑩ Advanced features for creating layers can be appropriately utilized
- . ⑩ Text can be appropriately created and positioned
- . ⑩ The printing terms kerning, leading, tracking, and fonts can be clearly defined
- . ⑩ Principles of good image composition and how to apply them can be clearly described
- . ⑩ A presentation image consisting of multiple layers of graphics can be

composed using principles of good composition

Module Topics

- . ⑩ 4.1 Why Are Layers Important for Digital Image Construction?
- . ⑩ 4.2 Working with Layers and the Layers Palette
- . ⑩ 4.3 Advanced Layers Features: Opacity, Blends, Gradient, and Magic Eraser and its Effects
- . ⑩ 4.5 The Text Tool
- . ⑩ 4.6 Principles of Composition and How They Apply to Photoshop
- . ⑩ 4.7 Case Study: Constructing a Presentation Graphic

Learning Outcomes

- . ⑩ Discuss how traditional art techniques have influenced digital imaging technology
- . ⑩ Select colors using multiple methods in Photoshop
- . ⑩ Utilize the primary painting tools to create and edit images—Paintbrush, Airbrush, Line Tool, Pencil Tool, Smudge Tool, Gradient, Eraser, and Fill Bucket
- . ⑩ Create a color scheme
- . ⑩ Describe how transparency, feathering, gradients, and texture filters can be used to modify a digital image

Performance Indicators

- . ⑩ Influence of traditional art techniques on digital imaging technology can be thoroughly discussed
- . ⑩ Colors can be appropriately selected using various methods in Photoshop
- . ⑩ Primary painting tools can be appropriately used to create and edit images
- . ⑩ A color scheme can be appropriately created
- . ⑩ Use of transparency, feathering, gradients, and texture filters to modify a digital image can be clearly described

Module Topics

- . ⑩ 5.1 Traditional Art Techniques and Digital Media
- . ⑩ 5.2 Color Selection in Photoshop
- . ⑩ 5.3 Photoshop Paint Tools: Paintbrush, Pencil Tool, Line Tool, Airbrush, Paint Bucket, Eraser, and Smudge Tool
- . ⑩ 5.4 Principles of Color Choice: Building a Color Scheme
- . ⑩ 5.5 Enhancing Color Effects through Transparency, Gradients, Feathering, and Filters
- . ⑩ 5.6 Case Study: Colorizing a Black and White Illustration

Learning Outcomes

- . ⑩ Discuss how old black and white photographs are enhanced using traditional photo-retouching techniques
- . ⑩ Describe the basic procedure and principles of scanning a photograph or other image
- . ⑩ Identify problem areas in old photographs and use the appropriate

Photoshop tool to correct them from the following toolset: Hue/Saturation Adjustment, Brightness/Contrast Adjustment, Levels, Rubber Stamp Tool, Unsharp Mask Filter, Dodge Tool, Burn Tool, and Sponge Tool

- . ⑩ Utilize Adjustment Layers to create editable changes in a layer
- . ⑩ Utilize the Replace Color Tool

Performance Indicators

- . ⑩ Enhancement of old black and white photographs using traditional photo-retouching techniques can be thoroughly discussed
- . ⑩ Basic procedure and principles of scanning a photograph or other image can be clearly described
- . ⑩ Problem areas in old photographs can be identified and appropriately corrected using Photoshop tools
- . ⑩ Adjustment layers can be used to create editable changes in a layer
- . ⑩ The Replace Color Tool can be appropriately used

Module Topics

- . ⑩ 6.1 Enhancing and Restoring Photographs: The Traditional Approach
- . ⑩ 6.2 Basic Scanning Principles and Techniques
- . ⑩ 6.3 Photoshop Image Enhancement Tools: Hue/Saturation, Brightness/Contrast, Levels, Rubber Stamp Tool, and Unsharp Mask Filter
- . ⑩ 6.4 Using Adjustment Layers
- . ⑩ 6.5 Selecting and Replacing Colors
- . ⑩ 6.6 Case Study: Enhancing an Old Photograph

Learning Outcomes

- . ⑩ Describe the origins of presentation graphics and the use of standard visual effects
- . ⑩ Describe an image filter and demonstrate how it is accessed and applied in Photoshop
- . ⑩ Utilize Photoshop filters and layer effects and describe situations where they are appropriately used
- . ⑩ Demonstrate how text can be enhanced using filters, layer effects, and the Transform tool
- . ⑩ Demonstrate five techniques for improving work efficiency in Photoshop

Performance Indicators

- . ⑩ Origins of presentation graphics and use of standard visual effects can be clearly described
- . ⑩ Application of Image filters in Photoshop can be appropriately applied
- . ⑩ Text enhancement using appropriate Photoshop tools can be demonstrated
- . ⑩ Five techniques for improving work efficiency in Photoshop can be demonstrated

Module Topics

- . ⑩ 7.1 Early Presentation Graphics and Effects

- . ⑩ 7.2 What Are Image Filters?
- . ⑩ 7.3 Layer Effects and their Applications: Drop Shadow, Inner Shadow, Outer Glow, Inner Glow, Bevel, and Emboss
- . ⑩ 7.4 Techniques for Enhancing Text
- . ⑩ 7.5 Tips for Working Effectively in Photoshop
- . ⑩ 7.6 Case Study: Creating a Web Site Banner

Learning Outcomes

- . ⑩ Describe the origins of Web graphics and their applications for the World Wide Web
- . ⑩ Describe the differences among GIF, JPEG, and PNG image file formats and situations in which each format should be used
- . ⑩ Demonstrate primary strategies for optimizing images for use on the World Wide Web: color reduction, compression, and sizing
- . ⑩ Create a simple GIF animation for display on a Web page
- . ⑩ Utilize a text processor and HTML to insert a Web-ready image into a Web page for display on an Internet browser

Performance Indicators

- . ⑩ The origins of Web graphics and their applications for the World Wide Web can be clearly described
- . ⑩ The differences between GIF, JPEG, and PNG image file formats and appropriate application can be clearly delineated
- . ⑩ Primary strategies for optimizing images for use on the World Wide Web can be appropriately demonstrated
- . ⑩ A simple GIF animation for display on a Web page can be successfully created
- . ⑩ HTML can be used to insert a Web-ready image into a Web page for display on an Internet browser

Module Topics

- . ⑩ 8.2 Using Images on the World Wide Web
- . ⑩ 8.3 GIF, JPEG, and PNG Image File Formats
- . ⑩ 8.4 Optimizing Images for Use on the World Wide Web—Dithering, Bit-depth, and the Browser-safe Palette
- . ⑩ 8.5 GIF Animation
- . ⑩ 8.6 Inserting a Web-ready Image into an HTML Document
- . ⑩ 8.7 Case Study: Creating a Web Page with Images

COURSE 4: WEB SITE DESIGN PRINCIPLES

Overview.....
 1

Module 1: Creating Effective Content
..... 5

Module 2: Web Site Design For Every Browser and Screen
..... 6

Module 3: Choosing the Most Appropriate Multimedia Formats
..... 7

Module 4: Designing User-Friendly Site Navigation
..... 8

Module 5: Using Server-side Includes and Scripting..... 9

Module 6: Receiving and Organizing Feedback
..... 10

Module 7: Providing Customer Service
..... 11

Module 8: Adding Databases to Your Site..... 12

Module 9: Verifying the Final Product 13

This course is intended as an overview of the principles of effective Web site design. Creating effective and successful Web sites requires identification of the site's audience; an understanding of efficient, user-friendly interfaces, and a logical structure. This course covers creating effective content, designing the pages to look their best on every browser and screen resolution, choosing appropriate multimedia formats, enabling user feedback, providing customer service, adding databases, and verifying the final product.

Instructional Goals

- . ⑩ Create content that is valuable to the intended audience
- . ⑩ Design pages that are viewable on a variety of browsers and screen resolutions
- . ⑩ Choose most appropriate multimedia formats for the Web site
- . ⑩ Design user-friendly site navigation
- . ⑩ Enable and organize user feedback
- . ⑩ Work with server-side includes and scripting
- . ⑩ Design customer service strategies
- . ⑩ Analyze requirements and determine when to add databases
- . ⑩ Verify the site is error-free

Course Prerequisites

- . ⑩ An understanding of Internet and Web page terminology
- . ⑩ Knowledge of HTML and Web page construction

Learning Outcomes

- . ⑩ Identify the purpose of the Web site
- . ⑩ Identify a potential audience for the Web site
- . ⑩ Determine the benefits of content to the audience
- . ⑩ Determine how to maintain user interest in the content
- . ⑩ Develop strategies to solicit user-submitted content
- . ⑩ Research and include content from other sources (such as newspapers)
- . ⑩ Develop strategies for creating original content
- . ⑩ Determine when material can legally be used (copyright material)
- . ⑩ Create content for foreign users
- . ⑩ Determine the scope of the site and what equipment and software will be used to create it

- . ⑩ Compare and contrast the capabilities of the most widely used browser software
- . ⑩ Develop strategies for creating a “browser-independent” site
- . ⑩ Discuss the pros and cons of using frames in a Web site
- . ⑩ Analyze the way in which browsers display colors
- . ⑩ Explain how screen resolution affects the appearance of Web pages
- . ⑩ Research methods to create pages that will show appropriately on different screen resolutions
- . ⑩ Explain the differences between the most common graphics formats
- . ⑩ Explain the importance of graphics file size
- . ⑩ Research appropriate software tools to modify graphics files
- . ⑩ Demonstrate the most appropriate HTML tags and attributes used to load graphics
- . ⑩ Explain the differences between video formats
- . ⑩ Describe the HTML tags used to display video
- . ⑩ List the most popular video player software
- . ⑩ Define streaming video
- . ⑩ Compare and contrast the different audio file formats
- . ⑩ Explain how to play audio files in a Web page
- . ⑩ Explain the importance of user-friendly navigation
- . ⑩ Discuss the pros and cons of text navigation, graphic links, and image maps

Learning Outcomes, *continued*

- . ⑩ Explain the importance of placement and consistency in Web site navigation
- . ⑩ Explain the “3 clicks” rule of site navigation
- . ⑩ Define what a site map is and how it is important
- . ⑩ Explain the importance of using familiar icons as links
- . ⑩ Research how to create custom icons and where to download royalty-free material for navigation
- . ⑩ Explain the importance of listening and responding to user feedback about site navigation
- . ⑩ Explain the purpose and concept of server-side includes
- . ⑩ Explain the advantages of using server-side includes in maintaining Web sites
- . ⑩ Demonstrate how to implement a server-side include in a Web page
- . ⑩ Explain the concept and purpose of using server-side script in Web pages
- . ⑩ List the pros and cons of using server-side script
- . ⑩ Identify the scripting languages and Web servers that support server-side script
- . ⑩ Define the purpose of cascading style sheets
- . ⑩ Analyze the types of feedback your Web site will generate
- . ⑩ Analyze user questions for repetition to determine how to respond
- . ⑩ Discuss different methods for getting user feedback on a Web page
- . ⑩ Discuss different methods for sending responses back to the user

- . ⑩ Design methods to automate responding to a user
- . ⑩ Evaluate feedback-related services offered by Web hosting services
- . ⑩ Analyze the reasons users visit a site
- . ⑩ Develop methods for users to communicate with the Webmaster and with each other
- . ⑩ Develop methods for prompt and accurate response to customer requests
- . ⑩ Investigate tools for providing statistics on customer demographics and habits
- . ⑩ Develop strategies for personal “one-on-one” service
- . ⑩ Explain how databases can enhance a Web site
- . ⑩ Differentiate between client-and server-side processing and which is used for database connectivity
- . ⑩ Discuss different types of database servers and their characteristics
- . ⑩ Discuss server-side scripting languages that can be used to access data in a database server
- . ⑩ Define Extensible Markup Language (XML) and its role in database connectivity
- . ⑩ Verify links and navigational tools for validity and user-friendliness
- . ⑩ Check that the text between the title tags is meaningful
- . ⑩ Verify spelling and grammar on all pages
- . ⑩ Verify that font size is appropriate for most users
- . ⑩ Explain why “off-page” links may drain users from a site
- . ⑩ Inspect pages for appropriate use of white space
- . ⑩ Verify that all graphics have meaningful <ALT> tags for browsers that do not support graphics
- . ⑩ Test page rendering on most commonly used screen resolutions and browsers
- . ⑩ Test database connectivity, client-side scripts, and user feedback functions
- . ⑩ Solicit feedback from colleagues and others

Performance Indicators

- . ⑩ Useable, compelling content can be created for the intended audience
- . ⑩ Web pages that can be viewed on a variety of browsers and screen resolutions can be created
- . ⑩ Graphics, videos, and sound can be used on Web pages in the most efficient formats
- . ⑩ Logical and user-friendly site navigation can be designed
- . ⑩ Customer needs can be met by providing opportunity for feedback and providing customer service
- . ⑩ Server-side includes and scripting can be appropriately demonstrated
- . ⑩ Requirements can be analyzed and addition of database connectivity can be appropriately determined
- . ⑩ Web pages can be verified for appropriate fonts, links, spelling, grammar, and effectiveness

Module Titles

1. 1. Creating Effective Content
2. 2. Web Site Design For Every Browser and Screen
3. 3. Choosing the Most Appropriate Multimedia Formats
4. 4. Designing User-Friendly Site Navigation
5. 5. Using Server-side Includes and Scripting
6. 6. Receiving and Organizing Feedback
7. 7. Providing Customer Service
8. 8. Adding Databases to Your Site
9. 9. Verifying the Final Product

Learning Outcomes

- . ⑩ Identify the purpose of the Web site
- . ⑩ Identify a potential audience for the Web site
- . ⑩ Determine the benefits of content to the audience
- . ⑩ Determine how to maintain user interest in the content
- . ⑩ Develop strategies to solicit user-submitted content
- . ⑩ Research and include content from other sources (such as newspapers)
- . ⑩ Develop strategies for creating original content:
 - . ⑩ Brainstorming
 - . ⑩ Asking colleagues
 - . ⑩ Research
 - . ⑩ Personal experience
 - . ⑩ Determine when material can legally be used (copyrighted material)
- . ⑩ Create content for foreign users:
 - . ⑩ Take cultural differences into account (some material may be highly offensive to some cultures)
 - . ⑩ Make sure language translations are not insulting or incorrect (beware of word-for-word translations)

Performance Indicators

- . ⑩ Purpose and scope of a Web site can be clearly identified
- . ⑩ Potential audience can be clearly identified
- . ⑩ Benefits of the site can be determined based upon its purpose
- . ⑩ Ways to maintain user interest can be thoroughly explored
- . ⑩ Strategies to solicit user-submitted content can be explored
- . ⑩ Content from outside sources can be researched and included appropriately
- . ⑩ Strategies for creating original content can be developed
- . ⑩ Material can be verified to ensure that it is used legally
- . ⑩ Content for potential foreign visitors can be created appropriately

Module Topics

- . ⑩ 1.1 Creating Your Concept
- . ⑩ 1.2 Identifying the Potential Audience
- . ⑩ 1.3 Making Your Site Valuable to Your Audience
- . ⑩ 1.4 Adding User-supplied Content

- . ⑩ 1.5 Creating Original Content
- . ⑩ 1.6 Is it Legal?
- . ⑩ 1.7 Foreign Visitors

Learning Outcomes

- . ⑩ Determine the scope of the site and what equipment and software will be used to create it:
 - . ⑩ Intranets vs. Internet sites
 - . ⑩ Compare and contrast the capabilities of the most widely used browser software
 - . ⑩ Develop strategies for creating a “browser-independent” site:
 - . ⑩ Alternate pages for different browsers
 - . ⑩ Server-side script
 - . ⑩ Discuss the pros and cons of using frames in a Web site
 - . ⑩ Analyze the way in which browsers display colors
 - . ⑩ Explain how screen resolution affects the appearance of Web pages
 - . ⑩ Research methods to create pages that will show appropriately on different screen resolutions

Performance Indicators

- . ⑩ Scope of the Web site and the equipment and browser software used to create it are determined
- . ⑩ Capabilities of the most widely used browser software can be compared and contrasted
- . ⑩ Strategies for creating “browser-independent” sites can be developed
- . ⑩ Pros and cons of using frames can be discussed
- . ⑩ Differences in the way browsers display colors can be thoroughly analyzed
- . ⑩ The effect of screen resolution on the appearance of the page can be clearly explained
- . ⑩ Methods for creating pages for different screen resolutions can be researched

Module Topics

- . ⑩ 2.1 Determine the Environment
- . ⑩ 2.2 Examining Browser Capabilities
- . ⑩ 2.3 Creating Browser-independent Pages
- . ⑩ 2.4 Pros and Cons of Using Frames
- . ⑩ 2.5 Colors and Browsers
- . ⑩ 2.6 The Importance of Considering Screen Resolution

Learning Outcomes

- . ⑩ Explain the differences between the most common graphics formats:
 - . ⑩ GIF
 - . ⑩ JPG
 - . ⑩ BMP

- . ⑩ Other
- . ⑩ Explain the importance of graphics file size:
 - . ⑩ Effects on page loading
 - . ⑩ Do's and don'ts
- . ⑩ Research appropriate software tools to modify graphics files:
 - . ⑩ Shareware
 - . ⑩ Purchased software
- . ⑩ Demonstrate the most appropriate HTML tags and attributes used to load graphics:
 - . ⑩ Importance of height and width attributes
 - . ⑩ Explain the differences between video formats
 - . ⑩ Describe the HTML tags used to display video:
 - . ⑩ Embedded video
 - . ⑩ Linked video
 - . ⑩ List the most popular video player software
 - . ⑩ Define streaming video
 - . ⑩ Compare and contrast the different audio file formats
 - . ⑩ Explain how to play audio files in a Web page

Performance Indicators

- . ⑩ Differences between graphics file formats can be clearly explained
- . ⑩ The importance and effect of graphics file size can be clearly explained
- . ⑩ Software tools for graphics editing can be thoroughly researched
- . ⑩ HTML tags and attributes for displaying graphics files can be demonstrated
- . ⑩ Differences between video formats can be clearly explained
- . ⑩ The most popular video player software can be listed
- . ⑩ Streaming video can be clearly defined
- . ⑩ Audio file formats can be compared and contrasted
- . ⑩ Methods for playing audio files in a Web page can be clearly explained

Module Topics

- . ⑩ 3.1 Using Graphics Files
- . ⑩ 3.2 Modifying Graphics Files
- . ⑩ 3.3 Adding Video to Your Site
- . ⑩ 3.4 Using Streaming Video
- . ⑩ 3.5 Adding Audio to Your Site

Learning Outcomes

- . ⑩ Explain the importance of user-friendly navigation
- . ⑩ Discuss the pros and cons of text navigation, graphic links, and image maps
- . ⑩ Explain the importance of placement and consistency in Web site navigation
- . ⑩ Explain the “3 clicks” rule of site navigation
- . ⑩ Define what a site map is and how it is important
- . ⑩ Explain the importance of using familiar icons as links

- . ⑩ Research how to create custom icons and where to download royalty-free material for navigation
- . ⑩ Explain the importance of listening and responding to user feedback about site navigation

Performance Indicators

- . ⑩ Importance of user-friendly site navigation can be clearly explained
- . ⑩ Pros and cons of text, graphics, and image maps as navigation tools can be thoroughly discussed
- . ⑩ Importance of placement and consistency in site navigation can be clearly explained
- . ⑩ “3 clicks” rule of site navigation can be clearly explained
- . ⑩ Concept and importance of a site map can be clearly defined
- . ⑩ Importance of using familiar icons as links can be clearly explained
- . ⑩ Software for creating custom icons and sites for downloading royalty-free icons can be researched
- . ⑩ Importance of listening and responding to user feedback regarding your site navigation can be clearly explained

Module Topics

- . ⑩ 4.1 The Importance of Creating User-friendly Site Navigation
- . ⑩ 4.2 Text Links, Graphic Links, and Image Maps
- . ⑩ 4.3 The Importance of Placement and Consistency
- . ⑩ 4.4 Don’t Let Your Users Get Lost or Bored!
- . ⑩ 4.5 Use of Icons as Site Navigation Tools
- . ⑩ 4.6 Listen to Your Users

Learning Outcomes

- . ⑩ Explain the purpose and concept of server-side includes:
- . ⑩ Server must support
- . ⑩ Explain the advantages of using server-side includes in maintaining Web sites:
- . ⑩ Can update items on many pages by changing one include file on the server
- . ⑩ Demonstrate how to implement a server-side include in a Web page
- . ⑩ Explain the concept and purpose of using server-side script in Web pages
- . ⑩ List the pros and cons of using server-side script
- . ⑩ Identify the scripting languages and Web servers that support server-side script
- . ⑩ Define the purpose of cascading style sheets

Performance Indicators

- . ⑩ The concept and purpose of server-side includes can be clearly explained
- . ⑩ Advantages of using server-side includes for site maintenance can be clearly explained
- . ⑩ Implementing a server-side include in a Web page can be successfully demonstrated
- . ⑩ The concept and purpose of server-side script can be clearly explained

- . ⑩ Pros and cons of using server-side script can be completely listed
- . ⑩ Scripting languages and Web servers that support server-side script can be identified
- . ⑩ The concept and purpose of cascading style sheets can be clearly explained

Module Topics

- . ⑩ 5.1 What is a Server-Side Include?
- . ⑩ 5.2 Why Use Server-Side Include Files?
- . ⑩ 5.3 Implementing Server-Side Includes
- . ⑩ 5.4 What is Server-Side Script?
- . ⑩ 5.5 Implementing Server-Side Script
- . ⑩ 5.6 Using Cascading Style Sheets

Learning Outcomes

- . ⑩ Analyze the types of feedback your Web site will generate:
 - . ⑩ Criticism
 - . ⑩ Praise
 - . ⑩ Request for information
 - . ⑩ Membership registration
 - . ⑩ Other
 - . ⑩ Analyze user questions for repetition to determine how to respond
- . ⑩ Discuss different methods for getting user feedback on a Web page:
 - . ⑩ Forms
 - . ⑩ Email
- . ⑩ Discuss different methods for sending responses back to the user:
 - . ⑩ Web page
 - . ⑩ Email
 - . ⑩ Phone
- . ⑩ Design methods to automate responding to a user:
 - . ⑩ Auto response messages
 - . ⑩ Using server-side script to create dynamic pages (using databases, for example)
 - . ⑩ FAQ's
- . ⑩ Evaluate feedback-related services offered by Web hosting services:
 - . ⑩ Forms support
 - . ⑩ Auto response email
 - . ⑩ CGI
 - . ⑩ FTP support
 - . ⑩ Database support

Performance Indicators

- . ⑩ Types of user feedback can be thoroughly analyzed
- . ⑩ Requests from users can be thoroughly analyzed for repetition to determine the most appropriate method of response
- . ⑩ Methods of gathering user feedback can be thoroughly discussed

- . ⑩ Methods of sending responses can be thoroughly discussed
- . ⑩ Methods to automate response to users can be designed
- . ⑩ Web hosting services can be thoroughly evaluated to determine whether they offer appropriate tools for your feedback requirements

Module Topics

- . ⑩ 6.1 Analyzing User Feedback
- . ⑩ 6.2 Gathering User Input
- . ⑩ 6.3 Responding to the User
- . ⑩ 6.4 Automating Your Responses
- . ⑩ 6.5 Evaluate Web Hosting Services

Learning Outcomes

- . ⑩ Analyze the reasons users visit a site:
 - . ⑩ Products
 - . ⑩ Information
 - . ⑩ Chat
 - . ⑩ Other
- . ⑩ Develop methods for users to communicate with the Webmaster and with each other:
 - . ⑩ Planned discussion
 - . ⑩ Impromptu chats
 - . ⑩ Forms
 - . ⑩ Email
 - . ⑩ Develop methods for prompt and accurate response to customer requests
 - . ⑩ Investigate tools for providing statistics on customer demographics and habits
- . ⑩ Develop strategies for personal one-on-one service:
 - . ⑩ Cookies to customize the site to the user
 - . ⑩ Databases to store customer preferences
 - . ⑩ Emails

Performance Indicators

- . ⑩ Customer motivations for visiting a site can be thoroughly analyzed
- . ⑩ Methods for allowing customers easy communication with Webmaster/company and with each other can be appropriately developed
- . ⑩ Methods for prompt and accurate response to customer requests can be developed
- . ⑩ Tools for providing customer analysis can be thoroughly investigated
- . ⑩ Strategies from providing one-on-one personal service can be developed

Module Topics

- . ⑩ 7.1 The Importance of Using Databases in a Web Application
- . ⑩ 7.2 Connecting to the Database Using ADO and ASP
- . ⑩ 7.3 Reading and Displaying Data from a Database
- . ⑩ 7.4 Inserting, Updating, and Deleting Database Records

Learning Outcomes

- . ⑩ Explain how databases can enhance a Web site
- . ⑩ Differentiate between client-and server-side processing and which is used for database connectivity
- . ⑩ Discuss different types of database servers and their characteristics:
 - . ⑩ Oracle
 - . ⑩ SQL Server
 - . ⑩ MySQL
- . ⑩ Discuss server-side scripting languages that can be used to access data in a database server:
 - . ⑩ PHP
 - . ⑩ ASP
 - . ⑩ JAVA Server Pages
- . ⑩ Define Extensible Markup Language (XML) and its role in database connectivity

Performance Indicators

- . ⑩ The importance of using databases in Web sites can be clearly explained
- . ⑩ Differences between client-side and server-side processing can be clearly differentiated as they relate to database connectivity
- . ⑩ Different types of database servers and their characteristics can be thoroughly discussed
- . ⑩ Server-side scripting languages capable of database access can be thoroughly discussed
- . ⑩ Extensible Markup Language (XML) and its role in database connectivity can be clearly defined

Module Topics

- . ⑩ 8.1 The Importance of Adding Databases to Web Sites
- . ⑩ 8.2 Client-side vs. Server-side and Databases
- . ⑩ 8.3 Database Servers
- . ⑩ 8.4 Choosing a Server-side Scripting Language
- . ⑩ 8.5 Extensible Markup Language (XML) and Database Connectivity

Learning Outcomes

- . ⑩ Verify links and navigational tools for validity and user-friendliness
- . ⑩ Check that the text between the title tags is meaningful:
- . ⑩ This is what will appear in user bookmarks and in search engines
- . ⑩ Verify spelling and grammar on all pages
- . ⑩ Verify that font size is appropriate for most users:
 - . ⑩ Not too small or large
 - . ⑩ Use fonts without serifs for reading-disabled people
 - . ⑩ Explain why “off-page” links may drain users from a site
 - . ⑩ Inspect pages for appropriate use of white space
- . ⑩ Verify that all graphics have meaningful <ALT> tags for browsers that do

not support graphics

- . ⑩ Test page rendering on most commonly used screen resolutions and browsers
- . ⑩ Test database connectivity, client-side scripts, and user feedback functions
- . ⑩ Solicit feedback from colleagues and others

Performance Indicators

- . ⑩ Links and navigational design can be thoroughly verified
- . ⑩ Spelling and grammar can be thoroughly verified
- . ⑩ Appearance of pages can be tested on most commonly used browsers and screen resolutions
- . ⑩ Fonts can be verified for appropriate size and type
- . ⑩ Graphics can be tested to make sure they contain ALT text
- . ⑩ Purpose and necessity of any off-page links can be verified
- . ⑩ Database connectivity, client-and server-side components and user feedback functions can be thoroughly tested
- . ⑩ Feedback can be solicited from colleagues and other neutral parties

Module Topics

- . ⑩ 9.1 Verifying Links and Navigation
- . ⑩ 9.2 Verifying Content for Spelling and Grammar
- . ⑩ 9.3 Test the Appearance and Effectiveness of Your Site
- . ⑩ 9.4 Testing Client and Server Components
- . ⑩ 9.5 Try Your Product Out on Others

COURSE 5: INTERNET SERVER TECHNOLOGIES

Overview

.....
1

Module 1: Overview of Internet Services

..... **5**

Module 2: Internet and Web

Infrastructure..... 6

Module 3: Web Server Services..... 7

Module 4: Planning and Developing Web Sites 8

Module 5: Hosting an Internet Site 9

Module 6: Installation of Web Server Software 10

Module 7: Database Storage 11

Module 8: Web Server Security 12

Module 9: Optimizing Web Server Performance 13

Module 10: Managing the Web Server

..... 14

This ten-module course is designed to introduce the Web development student to the hardware, software, and protocols used on Internet servers. Internet server software and the technologies it supports will determine what features and tools the developer may employ when creating Web applications. This course provides information on the most critical Internet information services such as email, file transfer, and e-commerce. It will also provide an overview of intranets, extranets, and Web portals. Features of some of the most popular Internet server software will be covered. In addition, issues of security, database support, Web site design considerations, server management, and performance tuning will be discussed. Students will have an opportunity to complete projects and assignments that will reinforce these topics.

Instructional Goals

- . ⑩ Describe the role of the server on the Internet, along with the different types of networks and services currently supported by these servers
- . ⑩ Describe the protocols, hardware, and software that enable data to be transferred over the Internet
- . ⑩ Explain how email, newsgroups, and other Internet services are provided
- . ⑩ Discuss the issues of compatibility and supported technologies related to hosting a Web application on an Internet server
- . ⑩ Describe the features and capabilities of the most popular Internet server software
- . ⑩ Demonstrate basic installation procedures for Internet servers
- . ⑩ Discuss issues related to security on Internet servers
- . ⑩ Analyze performance and implement procedures for server monitoring and maintenance

Course Prerequisites

- . ⑩ An understanding of basic computer concepts, file structure, and storage
- . ⑩ Knowledge of the Internet and the World Wide Web

Learning Outcomes

- . ⑩ Explain the evolution of the Internet and the role of the server
- . ⑩ Differentiate between the following types of networks: Internet, intranet, extranet, and Web portal
- . ⑩ Explain the concept of “Business to Business” (B2B)
- . ⑩ Explain the concept of “Business to Customer” (B2C)
- . ⑩ List other Internet information services currently in use
- . ⑩ Explain the current technologies in Internet communications
- . ⑩ Explain the role of Internet Service Providers (ISP)
- . ⑩ Define TCP/IP protocol and its role on the Internet
- . ⑩ Explain the function of routers, firewalls, and switches
- . ⑩ Explain the functions and purpose of “Dynamic Naming Service” (DNS)

- . ⑩ Define how DHCP works to assign IP addresses over a network
- . ⑩ Explain how Hypertext Transfer Protocol works and how it is used on the Internet
- . ⑩ Explain the function and purpose of File Transfer Protocol on the Internet
- . ⑩ Explain how email is transported on the Internet using SMTP
- . ⑩ Explain the functions of NNTP on the Internet
- . ⑩ Explain how streaming audio and video are implemented over the Internet
- . ⑩ List other important Web server services currently in use on the Internet (chat, credit card processing)
- . ⑩ List the steps that must be taken when planning a Web site that will be hosted on a Web server
- . ⑩ Explain client-side technologies and how they relate to the Web server
- . ⑩ Explain server-side technologies and how they are implemented on Web servers
- . ⑩ Describe XML and its various forms in use on the Internet today
- . ⑩ Explain the issues that must be considered when choosing a service provider to host a Web site

Learning Outcomes, *continued*

- . ⑩ Discuss the issues related to choosing an operating system on a Web server
- . ⑩ Explain the basic functions and capabilities of Web servers
- . ⑩ Compare and contrast the features of the most popular Web servers: IIS, Apache, Solaris, Netscape
- . ⑩ Discuss the hardware requirements for a Web server
- . ⑩ Discuss the issues related to selecting the appropriate hardware to run the Web server software
- . ⑩ Determine the components needed for Web server software installation
- . ⑩ Demonstrate the steps for basic installation and configuration of the most popular Web servers
- . ⑩ Add and remove components from a Web server
- . ⑩ Demonstrate advanced configuration such as setting up MTS, MMQS, and indexing
- . ⑩ Discuss issues related to supporting new Web technologies on the most popular Web servers
- . ⑩ Discuss troubleshooting strategies for Web server installations
- . ⑩ Discuss current relational database technologies and their importance
- . ⑩ Compare and contrast the most popular database management systems with emphasis on server requirements
- . ⑩ Demonstrate methods to connect to a database on an Internet server
- . ⑩ Demonstrate methods to retrieve data stored on an Internet server
- . ⑩ Discuss authentication methods used on the Internet—LDAP, NT/AD, and NDS
- . ⑩ Discuss file and directory security technologies
- . ⑩ Explain the concept and functions of encryption technologies
- . ⑩ Discuss the use of firewalls and proxy servers on the Internet

- . ⑩ Monitor and troubleshoot security on Internet servers
- . ⑩ Discuss performance issues related to server hardware
- . ⑩ Optimize server configuration
- . ⑩ Discuss optimization of cache engines for efficient performance
- . ⑩ Explain the importance of load balancing and clustering for optimum throughput
- . ⑩ Discuss the use of switching technologies in optimizing performance
- . ⑩ Monitor and troubleshoot common performance problems
- . ⑩ Explain the fundamental issues involved in managing a Web server installation
- . ⑩ Discuss management of users, rights, and directory security
- . ⑩ Analyze the requirements of Web sites hosted on the server for advanced components/technologies—data mining, site server, and e-commerce
- . ⑩ Discuss data backup strategies on the Web server
- . ⑩ Back up Web sites hosted on a Web server

Performance Indicators

- . ⑩ Primary role of Internet servers and the types of networks and services supported by those servers can be clearly described
- . ⑩ Protocols, hardware, and software used to transfer data over the Internet can be thoroughly described
- . ⑩ Protocols and methods used to transport email, and create and maintain newsgroups and other Internet services can be thoroughly discussed
- . ⑩ Issues related to compatibility and supported technologies for Web applications on an Internet server can be discussed
- . ⑩ Major features and capabilities of the most popular Internet server software can be clearly described
- . ⑩ Issues of data and user security on Internet servers can be thoroughly discussed
- . ⑩ Server performance and maintenance issues can be appropriately analyzed

Module Titles

1. 1. Overview of Internet Services
2. 2. Internet and Web Infrastructure
3. 3. Web Server Services
4. 4. Planning and Developing Web Sites
5. 5. Hosting an Internet Site
6. 6. Installation of Web Server Software
7. 7. Database Storage
8. 8. Web Server Security
9. 9. Optimizing Web Server Performance
10. 10. Managing the Web Server

Learning Outcomes

- . ⑩ Explain the evolution of the Internet and the role of the server
- . ⑩ Differentiate between the following types of networks: Internet, intranet,

extranet, and Web portal

- . ⑩ Explain the concept of “Business to Business” (B2B)
- . ⑩ Explain the concept of “Business to Customer” (B2C)
- . ⑩ List other Internet information services currently in use

Performance Indicators

- . ⑩ Evolution of the Internet and the role of servers can be clearly explained
- . ⑩ Differences between Internet networks, intranets, extranets, and Web portals can be clearly delineated
- . ⑩ The concept and purpose of B2B can be thoroughly explained
- . ⑩ The concept and purpose of B2C can be thoroughly explained
- . ⑩ Other current Internet information services can be appropriately listed

Module Topics

- . ⑩ 1.1 Evolution of the Internet
- . ⑩ 1.2 Overview of Internet Web Sites
- . ⑩ 1.3 E-commerce and other Internet Information Services

Module 2: Internet and Web Infrastructure

Learning Outcomes

- . ⑩ Explain the current technologies in Internet communications
- . ⑩ Explain the role of ISPs
- . ⑩ Define TCP/IP protocol and its role on the Internet
- . ⑩ Explain the function of routers, firewalls, and switches
- . ⑩ Explain the functions and purpose of “Dynamic Naming Service” (DNS)
- . ⑩ Define how DHCP works to assign IP addresses over a network

Performance Indicators

- . ⑩ Current technologies in Internet communications can be thoroughly explained
- . ⑩ The role of ISPs can be clearly explained
- . ⑩ TCP/IP protocol and its role on the Internet can be appropriately defined
- . ⑩ The functions of routers, firewalls, and switches can be clearly explained
- . ⑩ The functions and purpose of DNS are clearly explained
- . ⑩ Assignment of IP addresses using DHCP is clearly defined

Module Topics

- . ⑩ 2.1 Overview of Internet Communications
- . ⑩ 2.2 ISPs
- . ⑩ 2.3 Understanding TCP/IP
- . ⑩ 2.4 Routers, Firewalls, Switches
- . ⑩ 2.5 Overview of DNS and DHCP

Learning Outcomes

- . ⑩ Explain how Hypertext Transfer Protocol works and how it is used on the

Internet

- . ⑩ Explain the function and purpose of File Transfer Protocol on the Internet
- . ⑩ Explain how email is transported on the Internet using SMTP
- . ⑩ Explain the functions of NNTP on the Internet
- . ⑩ Explain how streaming audio and video are implemented over the Internet
- . ⑩ List other important Web server services currently in use on the Internet (chat, credit card processing, etc.)

Performance Indicators

- . ⑩ Function of Hypertext Transfer Protocol and its role on the Internet can be clearly explained
- . ⑩ Function and purpose of File Transfer Protocol can be clearly explained
- . ⑩ Method used to transport email over the Internet using SMTP can be clearly explained
- . ⑩ Functions of NNTP for newsgroups on the Internet can be clearly explained
- . ⑩ Methods of transporting streaming audio and video over the Internet can be clearly explained
- . ⑩ Other important Web server services can be thoroughly listed

Module Topics

- . ⑩ 3.1 WWW—HTTP Protocol
- . ⑩ 3.2 File Transfer—FTP
- . ⑩ 3.3 Sending Mail—SMTP
- . ⑩ 3.4 Newsgroups—NNTP Servers
- . ⑩ 3.5 Streaming Audio/Video
- . ⑩ 3.6 Other Services and Applications

Module 4: Planning and Developing Web Sites

Learning Outcomes

- . ⑩ List the steps that must be taken when planning a Web site that will be hosted on a Web server
- . ⑩ Explain client-side technologies and how they relate to the Web server
- . ⑩ Explain server-side technologies and how they are implemented on Web servers
- . ⑩ Describe XML and its various forms in use on the Internet today

Performance Indicators

- . ⑩ The most important steps that must be taken when designing a Web site to be hosted on a Web server can be listed
- . ⑩ Web page client-side technologies and their relationship to the Web server can be clearly explained
- . ⑩ Web page server-side technologies and their relationship to the Web server can be clearly explained
- . ⑩ XML and its various forms can be clearly described

Module Topics

- . ⑩ 4.1 Planning your Web site
- . ⑩ 4.2 Client-side Technology—JavaScript, VBScript, Flash, and Dynamic HTML
- . ⑩ 4.3 Server-side Technology—CGI, ASP, PHP, and Python
- . ⑩ 4.4 XML—XML, XSL, WDDX, XSLT, and SOAP

Learning Outcomes

- . ⑩ Explain the issues that must be considered when choosing a service provider to host a Web site
- . ⑩ Discuss the issues related to choosing an operating system on a Web server
- . ⑩ Explain the basic functions and capabilities of Web servers
- . ⑩ Compare and contrast the features of the most popular Web servers: IIS, Apache, Solaris, and Netscape
- . ⑩ Discuss the hardware requirements for a Web server
- . ⑩ Discuss the issues related to selecting the appropriate hardware to run the Web server software

Performance Indicators

- . ⑩ Issues to be considered when choosing a service provider to host an Internet Web site can be clearly explained
- . ⑩ Issues related to choosing an operating system for a Web server can be appropriately discussed
- . ⑩ Basic functions and features of Web servers can be clearly explained
- . ⑩ Features of the most popular Web server software can be appropriately compared and contrasted
- . ⑩ Hardware requirements for Web servers can be thoroughly discussed
- . ⑩ Issues related to selecting appropriate hardware to run the Web server software can be discussed

Module Topics

- . ⑩ 5.1 Choosing Where to Host Your Site
- . ⑩ 5.2 Choosing an Operating System
- . ⑩ 5.3 Web Server Basics
- . ⑩ 5.4 Overview of the Available Web Servers—IIS, Apache, Netscape, and Solaris
- . ⑩ 5.5 Web Server Hardware
- . ⑩ 5.6 Choosing the Right Combination of Hardware and Software

Module 6: Installation of Web Server Software

Learning Outcomes

- . ⑩ Determine the components needed for Web server software installation
- . ⑩ Demonstrate the steps for basic installation and configuration of the most

popular Web servers

- . ⑩ Add and remove components from a Web server
- . ⑩ Demonstrate advanced configuration such as setting up MTS, MMQS, and Indexing
- . ⑩ Discuss issues related to supporting new Web technologies on the most popular Web servers
- . ⑩ Discuss troubleshooting strategies for Web server installations

Performance Indicators

- . ⑩ Components necessary for installation of Web server software can be appropriately determined
- . ⑩ Basic installation and configuration of Web servers can be successfully demonstrated
- . ⑩ Components can be successfully added and removed from Web servers
- . ⑩ Advanced configuration of Web servers can be appropriately demonstrated
- . ⑩ Issues related to support new Web technologies can be thoroughly discussed
- . ⑩ Troubleshooting strategies and methods for Web servers can be thoroughly discussed

Module Topics

- . ⑩ 6.1 Determining Components Needed for Installation
- . ⑩ 6.2 Basic Installation and Configuration
- . ⑩ 6.3 Adding and Removing Components
- . ⑩ 6.4 Advanced Configuration—Setting up MTS, MMQS, and Indexing
- . ⑩ 6.5 Supporting New Web Technologies
- . ⑩ 6.6 Troubleshooting Installation

Learning Outcomes

- . ⑩ Discuss current relational database technologies and their importance
- . ⑩ Compare and contrast the most popular database management systems with emphasis on server requirements
- . ⑩ Demonstrate methods to connect to a database on an Internet server
- . ⑩ Demonstrate methods to retrieve data stored on an Internet server

Performance Indicators

- . ⑩ Current relational database technologies and their importance on the Internet can be discussed
- . ⑩ Features and requirements of popular database management systems can be compared and contrasted with emphasis on server requirements
- . ⑩ Methods used to connect to a database system on an Internet sever can be demonstrated
- . ⑩ Methods used to retrieve data stored on an Internet server can be demonstrated

Module Topics

- . ⑩ 7.1 Database Technology—Access, RDBMS, and XML Databases
- . ⑩ 7.2 Choosing the Right Database Product—MSSQL and Oracle
- . ⑩ 7.3 Connecting to a Database—ADO and ODBC
- . ⑩ 7.4 Retrieving the Data You Need—SQL and Java

Module 8: Web Server Security

Learning Outcomes

- . ⑩ Discuss authentication methods used on the Internet—LDAP, NT/AD, and NDS
- . ⑩ Discuss file and directory security technologies
- . ⑩ Explain the concept and functions of encryption technologies
- . ⑩ Discuss the use of firewalls and proxy servers on the Internet
- . ⑩ Monitor and troubleshoot security on Internet servers

Performance Indicators

- . ⑩ Authentication methods used on the Internet can be thoroughly discussed
- . ⑩ File and directory security technologies can be thoroughly discussed
- . ⑩ The appropriate use of firewalls and proxy servers can be thoroughly discussed
- . ⑩ Internet server security can be appropriately monitored and handled

Module Topics

- . ⑩ 8.1 Choosing Authentication Methods—LDAP, NT/AD, NDS, and Others
- . ⑩ 8.2 File and Directory Security—NTFS
- . ⑩ 8.3 SSL and Other Encryption Technology
- . ⑩ 8.4 Firewalls and Proxies
- . ⑩ 8.5 Monitoring and Troubleshooting Security

Learning Outcomes

- . ⑩ Discuss performance issues related to server hardware
- . ⑩ Optimize server configuration
- . ⑩ Discuss optimization of cache engines for efficient performance
- . ⑩ Explain the importance of load balancing and clustering for optimum throughput
- . ⑩ Discuss the use of switching technologies in optimizing performance
- . ⑩ Monitor and troubleshoot common performance problems

Performance Indicators

- . ⑩ Performance issues related to server hardware can be thoroughly discussed
- . ⑩ Web server configuration can be appropriately optimized
- . ⑩ Optimization of cache engines can be appropriately discussed
- . ⑩ Importance of load balancing and clustering for optimum throughput can be clearly explained
- . ⑩ Use of switching technologies can be thoroughly discussed

- . ⑩ Common performance problems can be monitored and investigated

Module Topics

- . ⑩ 9.1 Overview of Performance Issues
- . ⑩ 9.2 Server Hardware—SMP, Memory, and NIC
- . ⑩ 9.3 Server Configuration
- . ⑩ 9.4 Cache Engines
- . ⑩ 9.5 Load Balancing and Clustering
- . ⑩ 9.6 Switching Technology—Layer 2-7 Switching and QOS
- . ⑩ 9.7 Monitoring and Troubleshooting Performance

Module 10: Managing the Web Server

Learning Outcomes

- . ⑩ Explain the fundamental issues involved in managing a Web server installation
- . ⑩ Discuss management of users, rights, and directory security
- . ⑩ Analyze the requirements of Web sites hosted on the server for advanced components/technologies—data mining, site server, and E-commerce
- . ⑩ Discuss data backup strategies on the Web server
- . ⑩ Back up Web sites hosted on a Web server

Performance Indicators

- . ⑩ Fundamental issues involved in managing a Web server installation can be clearly explained
- . ⑩ Management of users, rights, and directory security can be thoroughly discussed
- . ⑩ Requirements of Web sites hosted on a server can be appropriately analyzed for use of additional components/technologies—data mining, site server, and E-commerce
- . ⑩ Data back up can be discussed and successfully performed on a Web server

Module Topics

- . ⑩ 10.1 Overview of Managing a Web Server
- . ⑩ 10.2 Managing Users/Rights and Directories—LDAP, ADSI, AD, NDS, and MMC
- . ⑩ 10.3 Analyzing Your Web Site—Data Mining, Site Server, and Marketing
- . ⑩ 10.4 Backing Up Your Web Site

COURSE 6: DATABASE DESIGN AND DEVELOPMENT FOR

WEB DEVELOPERS

Overview

.....
..... **1**

Module 1: Understanding Client-Server and N-tier Applications..... 5

Module 2: Creating the Database Mission Statement and Objectives 6

Module 3: Creating the Database Structure and Performing Database Normalization . 7

Module 4: Writing Structured Query Language Code 8

Module 5: Working with Database Indexes 9

Module 6: Writing Stored Procedures..... 10

Module 7: Creating Database Cursors
..... 11

Module 8: Working With Database Transactions..... 12

Module 9: Working with Users and Security..... 13

Module 10: Introduction to Data Warehousing..... 14

This ten-module course is designed to introduce the E-Commerce student to database design, development concepts, and the client-server software model as they apply to Web site development. Database-driven Web applications are central to E-Commerce and business-to-business technology on the Internet. It is imperative that the Web developer be well grounded in current database technology and n-tier development methods. Since this course is intended for beginning Web developers, its focus is on database concepts and programming rather than administration. This course covers database design and normalization methods and Structured Query Language (SQL) syntax including cursors, indexes, and stored procedures. In addition, database transactions and user security are addressed. The course also includes an introduction to data warehousing.

Instructional Goals

- . ⑩ Explain the evolution of client-server database computing, up to and including n-tier development and enterprise systems
- . ⑩ Create database mission statements and objectives using effective systems analysis techniques
- . ⑩ Design and normalize databases using mission statements, objectives and systems analysis techniques
- . ⑩ Write Structured Query Language code for database queries, maintenance, and creation of database objects such as indexes, cursors, and stored procedures
- . ⑩ Bundle database operations using transactions

- . ⑩ Explain methods used to enforce database user security
- . ⑩ Explain the concept of data warehousing and appropriate terminology

Course Prerequisites

- . ⑩ Introduction to the Internet and the World Wide Web
- . ⑩ Introduction to E-Commerce
- . ⑩ Introduction to Networking
- . ⑩ An understanding of basic computer file structure and storage
- . ⑩ Basic knowledge of programming is desirable but not required

Learning Outcomes

- . ⑩ Explain what is meant by mainframe computing
- . ⑩ Explain PC/File Server computing
- . ⑩ Compare and contrast Client/Server Database computing and the PC/File Server model
- . ⑩ Define the term Relational Database System
- . ⑩ Explain the purpose of the *n-tier* business logic component of the database computing environment
- . ⑩ Describe an *Enterprise Network*
- . ⑩ Explain the importance of completing a thorough design process
- . ⑩ Conduct user interviews to determine the overall purpose of the database and the tasks it must support
- . ⑩ Create a well-written database mission statement as a guide for the rest of the process
- . ⑩ Compose database mission objectives representing the general tasks supported by the data in the database
- . ⑩ Analyze any existing databases for additions and changes that need to be made
- . ⑩ Become familiar with the way in which data is processed in the existing system
- . ⑩ Compile a complete list of data elements from analysis of the current system and new requirements
- . ⑩ Define a list of tables from subjects in the mission objectives and analysis of the data elements list
- . ⑩ Determine field names and characteristics for each table
- . ⑩ Establish primary and foreign keys to create relationships between tables
- . ⑩ Normalize the database schema to ensure that it will be viable
- . ⑩ Determine business rules to support the way the organization functions
- . ⑩ Define the purpose and evolution of Structured Query Language

Learning Outcomes, *continued*

- . ⑩ Retrieve data using simple select statements, joins, aggregates, and sorting
- . ⑩ Create totals and subtotals using group by, cube, rollup, and compute
- . ⑩ Create complex SQL statements using sub queries
- . ⑩ Use data maintenance language to insert, update, and delete records
- . ⑩ Create temporary tables and views

- . ⑩ Explain the concept and purpose of a database index
- . ⑩ Define when indexes should be used to optimize queries
- . ⑩ Differentiate between clustered and non-clustered indexes
- . ⑩ Analyze query optimization using database tools (showplan)
- . ⑩ Create indexes on database tables using Structured Query Language
- . ⑩ Explain the concept and benefits of using stored procedures
- . ⑩ Write and edit a simple stored procedure using SQL
- . ⑩ Create stored procedures that use input parameters
- . ⑩ Write stored procedures using defaults and customized error messages
- . ⑩ Create stored procedures that send output parameters
- . ⑩ Return status values from stored procedures
- . ⑩ Compile stored procedures
- . ⑩ Optimize stored procedures
- . ⑩ Define database cursors
- . ⑩ Create a simple cursor using SQL
- . ⑩ Differentiate between client-side and server-side cursors
- . ⑩ Differentiate between set processing and row processing
- . ⑩ Create scrollable cursors
- . ⑩ Explain the limitations of cursors
- . ⑩ Discuss how cursors effect performance
- . ⑩ Define Database Transactions
- . ⑩ Explain the benefits of using transactions in database processing
- . ⑩ Differentiate between implicit and explicit transactions
- . ⑩ Write transactions using SQL
- . ⑩ Create transactions using transaction control statements
- . ⑩ Group transaction statements in batches
- . ⑩ Define locking and types of locking
- . ⑩ Explain the purpose and concept of database security
- . ⑩ Define what database users and logins are
- . ⑩ Define the concept of database permissions
- . ⑩ Assign security permissions to groups of users
- . ⑩ Create database backups and restores
- . ⑩ Explain the concept and purpose of a data warehouse
- . ⑩ Differentiate between warehouse data and operational data
- . ⑩ Explain data warehouse components (operational data, data transformations, metadata, data stores, end-user tools)
- . ⑩ Define what a data mart is
- . ⑩ Explain how data is transformed into the data warehouse
- . ⑩ Use *Stars* and *Snowflakes* in data warehouse design

Learning Outcomes, continued

- ⑩ Describe the tasks needed to manage the data warehouse (backup/recovery, scheduling the loading of information, and replication)

Performance Indicators

- . ⑩ Evolution of client-server computing through n-tier and enterprise systems can be clearly explained
- . ⑩ Database mission statements and objectives can be appropriately created
- . ⑩ Databases can be successfully designed and normalized to user specifications
- . ⑩ Structured Query Language code can be written to perform database queries and create database objects
- . ⑩ Database transactions can be appropriately used
- . ⑩ Database security can be clearly explained
- . ⑩ Data warehousing concepts and terminology can be clearly explained

Module Titles

1. 1. Understanding Client-Server and N-tier Applications
2. 2. Creating the Database Mission Statement and Objectives
3. 3. Creating the Database Structure and Performing Database Normalization
4. 4. Writing Structured Query Language Code
5. 5. Working with Database Indexes
6. 6. Writing Stored Procedures
7. 7. Creating Database Cursors
8. 8. Working With Database Transactions
9. 9. Working with Users and Security
10. 10. Introduction to Data Warehousing

Learning Outcomes

- . ⑩ Explain what is meant by mainframe computing
- . ⑩ Explain PC/File Server computing
- . ⑩ Compare and contrast Client/Server Database computing and the PC/File Server model.
- . ⑩ Define the term *Relational Database System*
- . ⑩ Explain the purpose of the *n-tier* business logic component of the database computing environment
- . ⑩ Describe an *Enterprise Network*

Performance Indicators

- . ⑩ Mainframe and PC/File Server computing can be clearly explained
- . ⑩ Differences between the Client/Server Database computing and the PC/File Server model can be clearly delineated
- . ⑩ Relational Database System can be clearly defined
- . ⑩ The function and advantages of the *n-tier* business logic component can be clearly explained
- . ⑩ The concept of an *enterprise network* can be thoroughly described

Module Topics

- . ⑩ 1.1 Mainframe Computing
- . ⑩ 1.2 PC/File Server Computing

- . ⑩ 1.3 Client/Server Database Computing
- . ⑩ 1.4 Relational Database Systems as the Standard for Client/Server Computing
- . ⑩ 1.5 Business Logic and the *N-Tier*
- . ⑩ 1.6 The Corporate System—Enterprise Networking

Learning Outcomes

- . ⑩ Explain the importance of completing a thorough design process
- . ⑩ Conduct user interviews to determine the overall purpose of the database and the tasks it must support
- . ⑩ Create a well-written database mission statement as a guide for the rest of the process
- . ⑩ Compose database mission objectives representing the general tasks supported by the data in the database
- . ⑩ Analyze any existing databases for additions/changes that need to be made
- . ⑩ Become familiar with the way in which data is processed in the existing system

Performance Indicators

- . ⑩ User interviews can be conducted using open-ended questions to determine the purpose for the database in the business
- . ⑩ A well-written database mission statement can be created
- . ⑩ Mission objectives can be composed to describe general tasks the data will support
- . ⑩ Existing databases and processes can be thoroughly analyzed

Module Topics

- . ⑩ 2.1 Conducting Effective User Interviews
- . ⑩ 2.2 Building Database Mission Statements
- . ⑩ 2.3 Composing Database Mission Objectives
- . ⑩ 2.4 Analyzing Existing Databases

Learning Outcomes

- . ⑩ Compile a complete list of data elements from analysis of the current system and new requirements
- . ⑩ Define a list of tables from subjects in the mission objectives and analysis of the data elements list
- . ⑩ Determine field names and characteristics for each table
- . ⑩ Establish primary and foreign keys to create relationships between tables
- . ⑩ Normalize the database schema to ensure that it will be viable
- . ⑩ Determine business rules to support the way the organization functions

Performance Indicators

- . ⑩ A complete list of data elements can be generated from existing databases, input and output documents
- . ⑩ Table subjects can be appropriately derived from mission objectives and

data elements lists

- . ⑩ Appropriate field names can be created and data characteristics assigned for each database table
- . ⑩ Primary and foreign keys can be established to create appropriate database relationships
- . ⑩ Databases can be normalized to the third normal form
- . ⑩ Appropriate business rules for the organization can be determined

Module Topics

- . ⑩ 3.1 Creating Database Tables
- . ⑩ 3.2 Creating Field Specifications
- . ⑩ 3.3 Establishing Relationships
- . ⑩ 3.4 Database Normalization
- . ⑩ 3.5 Creating Business Rules

Learning Outcomes

- . ⑩ Define the purpose and evolution of Structured Query Language
- . ⑩ Retrieve data using simple select statements, joins, aggregates, and sorting
- . ⑩ Create totals and subtotals using group by, cube, rollup, and compute
- . ⑩ Create complex SQL statements using sub queries
- . ⑩ Use data maintenance language to insert, update, and delete records
- . ⑩ Create temporary tables and views

Performance Indicators

- . ⑩ Purpose and evolution of Structured Query Language can be clearly defined
- . ⑩ Data can be retrieved from database tables using appropriate SQL select statement syntax
- . ⑩ Total, subtotals and grouping data can be accomplished using appropriate SQL syntax
- . ⑩ Data maintenance language can be appropriately used to insert, update and delete database records
- . ⑩ Temporary tables and views can be appropriately created

Module Topics

- . ⑩ 4.1 Overview of Structured Query Language
- . ⑩ 4.2 Writing SQL Select Statements
- . ⑩ 4.3 Generating Totals and Subtotals
- . ⑩ 4.4 Using Data Maintenance Language
- . ⑩ 4.5 Working with Temporary Tables and Views

Learning Outcomes

- . ⑩ Explain the concept and purpose of a database index
- . ⑩ Define when indexes should be used to optimize queries
- . ⑩ Differentiate between clustered and non-clustered indexes
- . ⑩ Analyze query optimization using database tools (showplan)

- . ⑩ Create indexes on database tables using Structured Query Language

Performance Indicators

- . ⑩ The concept and purpose of database indexes can be clearly explained
- . ⑩ Usage of indexes for query optimization can be clearly defined
- . ⑩ Differences and appropriate usage of clustered vs. non-clustered indexes can be clearly delineated
- . ⑩ Queries can be analyzed for optimal performance using appropriate database tools
- . ⑩ Indexes can be created on database tables using correct SQL syntax

Module Topics

- . ⑩ 5.1 What is a Database Index and How Does it Work?
- . ⑩ 5.2 Determining When to Add an Index for Optimization
- . ⑩ 5.3 Types of Indexes
- . ⑩ 5.4 Analyzing Query Optimization
- . ⑩ 5.5 Creating Indexes using SQL

Learning Outcomes

- . ⑩ Explain the concept and benefits of using stored procedures
- . ⑩ Write and edit a simple stored procedure using SQL
- . ⑩ Create stored procedures that use input parameters
- . ⑩ Write stored procedures using defaults and customized error messages
- . ⑩ Create stored procedures that send output parameters
- . ⑩ Return status values from stored procedures
- . ⑩ Compile stored procedures
- . ⑩ Optimize stored procedures

Performance Indicators

- . ⑩ Concept and benefits of using stored procedures can be clearly explained
- . ⑩ Simple stored procedures can be written, compiled and executed using correct SQL syntax
- . ⑩ Stored procedures using input parameters can be successfully created, compiled and executed
- . ⑩ Parameter defaults and customer error messages can be appropriately used in stored procedures
- . ⑩ Stored procedures using output parameters can be successfully created, compiled and executed
- . ⑩ Stored procedures that return status values can be successfully created, compiled and executed
- . ⑩ Stored procedures can be appropriately optimized using database optimization tools

Module Topics

- . ⑩ 6.1 Introduction to Stored Procedures
- . ⑩ 6.2 Writing a Simple Procedure

- . ⑩ 6.3 Stored Procedures and Input Parameters
- . ⑩ 6.4 Parameter Defaults and Custom Error Messages
- . ⑩ 6.5 Stored Procedures and Output Parameters
- . ⑩ 6.6 Returning Status Values
- . ⑩ 6.7 Stored Procedure Optimization

Learning Outcomes

- . ⑩ Define database cursors
- . ⑩ Create a simple cursor using SQL
- . ⑩ Differentiate between client-side and server-side cursors
- . ⑩ Differentiate between set processing and row processing
- . ⑩ Create scrollable cursors
- . ⑩ Explain the limitations of cursors
- . ⑩ Discuss how cursors effect performance

Performance Indicators

- . ⑩ Database cursors can be clearly defined
- . ⑩ Simple database cursors can be created using SQL
- . ⑩ Client and server-side cursors can be compared and contrasted
- . ⑩ Set and row processing can be clearly differentiated
- . ⑩ Scrollable cursors can be appropriately created
- . ⑩ The limitations of cursors can be clearly explained
- . ⑩ The ways in which cursors affect performance can be clearly explained

Module Topics

- . ⑩ 7.1 What is a Database Cursor
- . ⑩ 7.2 Cursor Syntax
- . ⑩ 7.3 Client-side vs. Server-side Cursors
- . ⑩ 7.4 When is Row Processing Appropriate?
- . ⑩ 7.5 Scrollable Cursors
- . ⑩ 7.6 Cursor Limitations and Performance

Learning Outcomes

- . ⑩ Define database transactions
- . ⑩ Explain the benefits of using transactions in database processing
- . ⑩ Differentiate between implicit and explicit transactions
- . ⑩ Write transactions using SQL
- . ⑩ Create transactions using transaction control statements
- . ⑩ Group transaction statements in batches
- . ⑩ Define locking and types of locking

Performance Indicators

- . ⑩ Concept and purpose of database transactions can be clearly defined
- . ⑩ Differences between implicit and explicit transactions can be clearly explained
- . ⑩ Transactions can be written using correct SQL syntax

- . ⑩ Transactions can be written using appropriate control statements
- . ⑩ Transaction statements can be written using appropriate grouping syntax
- . ⑩ Locking and types of locks can be clearly defined

Module Topics

- . ⑩ 8.1 Using the Recordset Object
- . ⑩ 8.2 Working with Database Fields
- . ⑩ 8.3 Creating Custom Recordsets with User Input
- . ⑩ 8.4 Displaying Results Using ADO Code

Learning Outcomes

- . ⑩ Explain the purpose and concept of database security
- . ⑩ Define what database users and logins are
- . ⑩ Define the concept of database permissions
- . ⑩ Assign security permissions to groups of users
- . ⑩ Create database backups and restores

Performance Indicators

- . ⑩ Purpose and concept of database security can be clearly explained
- . ⑩ Database users and logins can be clearly defined
- . ⑩ Concept of database and object permissions/rights can be clearly defined
- . ⑩ Security rights/permissions can be assigned to groups of users
- . ⑩ Database backups and restores can be successfully done

Module Topics

- . ⑩ 9.1 Introduction to Database Security
- . ⑩ 9.2 Users and Logins
- . ⑩ 9.3 Setting Database and Database Object Rights/Permissions
- . ⑩ 9.4 Assigning Rights/Permissions to Groups of Users
- . ⑩ 9.5 Performing Database Backups and Restores

Learning Outcomes

- . ⑩ Explain the concept and purpose of a data warehouse
- . ⑩ Differentiate between warehouse data and operational data
- . ⑩ Explain data warehouse components (operational data, data transformations, metadata, data stores, and end-user tools)
- . ⑩ Define what a data mart is
- . ⑩ Explain how data is transformed into the data warehouse
- . ⑩ Use *Stars* and *Snowflakes* in data warehouse design
- . ⑩ Describe the tasks needed to manage the data warehouse (backup/recovery, scheduling the loading of information, and replication)

Performance Indicators

- . ⑩ Concept and purpose of a data warehouse are clearly explained
- . ⑩ Warehouse data and operational data are clearly differentiated
- . ⑩ Data warehouse components are clearly explained

- . ⑩ The term *data mart* is clearly defined
- . ⑩ Transforming operational data into a data warehouse is clearly explained
- . ⑩ Design appropriate data warehouse models using Star and Snowflake schema
- . ⑩ Tasks necessary for managing the data warehouse are clearly described

Module Topics

- . ⑩ 10.1 What is a Data Warehouse?
- . ⑩ 10.2 Operational Data vs. Warehouse Data
- . ⑩ 10.3 Data Warehouse Components
- . ⑩ 10.4 What is a Data Mart?
- . ⑩ 10.5 Transforming Data
- . ⑩ 10.6 Data Warehouse Design
- . ⑩ 10.7 Managing the Data Warehouse

COURSE 7: CREATING DATABASE-DRIVEN WEB PAGES

USING OPEN SOURCE

Overview

.....
 1

Module 1:History of PHP, Downloading, Installing, and Configuring PHP..... 5

Module 2:PHP Basics..... 6

Module 3:Programming Control Structures using PHP 7

Module 4: Using Forms and Controls to Build Interactive Web Pages Using PHP..... 8

**Module 5: Downloading, Installing, and Configuring MySQL
..... 9**

**Module 6: Creating Database Objects Using
MySQL..... 10**

**Module 7: Maintaining and Retrieving Data Using
MySQL..... 11**

**Module 8: Creating Database-driven Web Pages Using PHP and
MySQL..... 12**

This course is designed to introduce the Web development student to building database-driven pages using “open source” tools available on the Internet. The tools used in this course will be PHP (Hypertext Preprocessor), which is a server-side scripting language, and MySQL, a relational database management system. Both of these products are available free or for a nominal fee and can be installed on UNIX-based servers. Upon completion of this course, students will be familiar with where and how to download, install, and configure the software. In addition, database creation and maintenance using MySQL will be covered. Building Web pages to capture user input, query, display, and maintain MySQL databases will be accomplished using PHP server-side scripting. Students will complete exercises and projects to practice these technologies.

Instructional Goals

- . ⑩ Download and install PHP (Hypertext Preprocessor)
- . ⑩ Download and install MySQL relational database software
- . ⑩ Create and maintain MySQL databases
- . ⑩ Write PHP server-side script to capture user input from a Web page
- . ⑩ Write PHP server-side script to send queries to MySQL databases and

display the results on the browser

Course Prerequisites

- . ⑩ Knowledge of Web page development and Internet terminology
- . ⑩ Fundamental knowledge of programming concepts
- . ⑩ Knowledge of database design concepts
- . ⑩ Knowledge of Structured Query Language

Learning Outcomes

- . ⑩ Explain the history and evolution of the PHP language
- . ⑩ Explain where and how to download a copy of PHP for UNIX and for Windows operating systems
- . ⑩ Download and install an appropriate version of PHP (UNIX or Windows)
- . ⑩ Complete the appropriate configuration steps for the appropriate operating system (UNIX or Windows)
- . ⑩ Complete the configuration process for MySQL database support
- . ⑩ Demonstrate basic PHP syntax:
 - . ⑩ PHP scripting characters
 - . ⑩ Analyze the source code that is returned to the user's browser from PHP server-side script:
 - . ⑩ Note that HTML is returned to the browser
- . ⑩ Demonstrate the use of the "echo" function to send output to the browser
- . ⑩ Demonstrate the use of variables in simple assignment statements
- . ⑩ Demonstrate the use of the math operators in PHP statements
- . ⑩ Demonstrate sending output to the browser using variables
- . ⑩ Demonstrate the use of PHP logical operators in creating Boolean expressions
- . ⑩ Demonstrate a simple IF THEN branch using PHP syntax
- . ⑩ Demonstrate an IF THEN ELSE branch using PHP syntax
- . ⑩ Demonstrate the use of conditional operators (AND and OR) in branching statements using PHP syntax
- . ⑩ Demonstrate the use of WHILE loops using PHP syntax
- . ⑩ Explain how data is sent in the header and body of an HTTP message using HTML forms:
 - . ⑩ Action = GET data sent in the header
 - . ⑩ Action = POST data sent in the body
 - . ⑩ Use of URL query string
 - . ⑩ Write HTML syntax using the URL query string to send data to the server in a link tag
- . ⑩ Explain how data in the URL query string is read by PHP scripts on the server
- . ⑩ Write PHP server-side scripts to process the data sent in the query string

Learning Outcomes, *continued*

- . ⑩ Demonstrate use of the URL query string to pass multiple values to PHP server-side scripts

⑩ Create HTML forms that pass their data to the server in the header of an HTTP request message:

⑩ Action = GET

. ⑩ Create PHP server-side scripts to process data sent by an HTML form

. ⑩ Explain where and how to download a copy of MySQL for UNIX and for Windows operating systems

. ⑩ Demonstrate starting the MySQL service on the Web server

. ⑩ Set a root password to restrict access to the MySQL service

. ⑩ Test the root password by logging in to MySQL

. ⑩ Shut down the SQL service on the Web server

. ⑩ Determine where to run the MySQL client software (local machine or remote Web server)

. ⑩ Connect to the MySQL client

. ⑩ Write simple commands using MySQL syntax at the MySQL command line prompt

. ⑩ Show the databases currently installed on the server

. ⑩ Create a new database on the MySQL server

. ⑩ Create database tables and fields on the MySQL server

. ⑩ Display MySQL database tables and fields on the user screen

. ⑩ Delete a table and/or database from MySQL

. ⑩ Insert data into a MySQL database table

. ⑩ Write simple queries to browse data in a MySQL database table

. ⑩ Write joining queries using MySQL tables

. ⑩ Update data in a MySQL database table

. ⑩ Delete data from a MySQL database table

. ⑩ Connect to a MySQL server using PHP syntax

. ⑩ Perform SQL queries on MySQL databases using PHP syntax

. ⑩ Return results of an SQL query into a PHP variable

. ⑩ Use “mysql_query” to detect success or failure of SQL query

. ⑩ Use “mysql_affected_rows” to determine number of rows affected by SQL insert and update or delete statements

. ⑩ Display data returned by SQL queries in an HTML page using PHP row-processing syntax embedded in HTML

Performance Indicators

. ⑩ PHP and MySQL software can be downloaded and installed appropriately on Web servers

. ⑩ Databases can be appropriately created and maintained using MySQL

. ⑩ Server-side script to capture user input can be correctly written using PHP

. ⑩ SQL queries can be executed and results can be displayed on the browser using PHP server-side scripting and MySQL databases

Module Titles

1. 1. History of PHP, Downloading, Installing, and Configuring PHP

2. 2. PHP Basics

3. 3. Programming Control Structures Using PHP

4. 4. Using Forms and Controls to Build Interactive Web Pages Using PHP
5. 5. Downloading, Installing, and Configuring MySQL
6. 6. Creating Database Objects Using MySQL
7. 7. Maintaining and Retrieving Data Using MySQL
8. 8. Creating Database-driven Web Pages Using PHP and MySQL

Learning Outcomes

- . ⑩ Explain the history and evolution of the PHP language
- . ⑩ Explain where and how to download a copy of PHP for UNIX and for Windows operating systems
- . ⑩ Download and install an appropriate version of PHP (UNIX or Windows)
- . ⑩ Complete the appropriate configuration steps for the appropriate operating system (UNIX or Windows)
- . ⑩ Complete the configuration process for MySQL database support

Performance Indicators

- . ⑩ History and evolution of PHP can be clearly explained
- . ⑩ The purpose of PHP and how it is “browser independent” can be clearly explained
- . ⑩ Location and download instructions for PHP UNIX and Windows versions can be clearly explained
- . ⑩ Appropriate version of PHP can be downloaded and installed on the operating system being used
- . ⑩ Configuration of PHP for appropriate operating system can be completed
- . ⑩ Configuration for MySQL database support can be successfully completed

Module Topics

- . ⑩ History and Purpose of PHP
- . ⑩ Downloading and Installing PHP
- . ⑩ Configuring PHP
- . ⑩ Adding Support for MySQL databases

Module 2: PHP Basics

Learning Outcomes

- . ⑩ Demonstrate basic PHP syntax:
- . ⑩ PHP scripting characters
- . ⑩ Analyze the source code that is returned to the user’s browser from PHP server-side script:
- . ⑩ Note that HTML is returned to the browser
- . ⑩ Demonstrate the use of the “echo” function to send output to the browser
- . ⑩ Demonstrate the use of variables in simple assignment statements
- . ⑩ Demonstrate the use of the math operators in PHP statements
- . ⑩ Demonstrate sending output to the browser using variables

Performance Indicators

- . ⑩ Basic PHP syntax can be appropriately demonstrated
- . ⑩ The source of a Web page created in PHP and sent to the browser can be thoroughly analyzed
- . ⑩ Echo functions to create output on the browser can be appropriately demonstrated
- . ⑩ Appropriate use of variables in assignment statements can be demonstrated
- . ⑩ Appropriate use of math operators for calculation statements can be demonstrated
- . ⑩ Appropriate use of variables in sending output to the browser can be demonstrated

Module Topics

- . ⑩ 2.1 PHP Basic Syntax
- . ⑩ 2.2 What is Returned to the Browser When PHP Script is Executed?
- . ⑩ 2.3 Variables and Operators
- . ⑩ 2.4 Using Variables in Browser Output

Learning Outcomes

- . ⑩ Demonstrate the use of PHP logical operators in creating Boolean expressions
- . ⑩ Demonstrate a simple IF THEN branch using PHP syntax
- . ⑩ Demonstrate an IF THEN ELSE branch using PHP syntax
- . ⑩ Demonstrate the use of conditional operators (AND and OR) in branching statements using PHP syntax
- . ⑩ Demonstrate the use of WHILE loops using PHP syntax

Performance Indicators

- . ⑩ Boolean expressions can be created using PHP logical operators
- . ⑩ Simple IF THEN branches can be appropriately coded using PHP syntax
- . ⑩ IF THEN ELSE branches can be appropriately coded using PHP syntax
- . ⑩ Conditional operators can be used to create complex branching statements using PHP syntax
- . ⑩ WHILE loops can be appropriately created using PHP syntax

Module Topics

- . ⑩ 3.1 Creating Boolean Statements Using Logical Operators
- . ⑩ 3.2 Implementing Branching Logic Using IF Statements
- . ⑩ 3.3 Creating Complex Branches Using Conditional Operators
- . ⑩ 3.4 Performing Iteration Using WHILE Loops

Module 4: Using Forms and Controls to Build Interactive Web Pages Using PHP

Learning Outcomes

- . ⑩ Explain how data is sent in the header and body of an HTTP message using

HTML forms:

- . ⑩ Action = GET data sent in the header
- . ⑩ Action = POST data sent in the body
- . ⑩ Use of URL query string
- . ⑩ Write HTML syntax using the URL query string to send data to the server in a link tag
- . ⑩ Explain how data in the URL query string is read by PHP scripts on the server
- . ⑩ Write PHP server-side scripts to process the data sent in the query string
- . ⑩ Demonstrate use of the URL query string to pass multiple values to PHP server-side scripts
- . ⑩ Create HTML forms that pass their data to the server in the header of an HTTP request message:
 - . ⑩ Action = GET
- . ⑩ Create PHP server-side scripts to process data sent by an HTML form

Performance Indicators

- . ⑩ Format and method of how data is sent in an HTTP request method can be thoroughly explained
- . ⑩ HTML syntax can be appropriately written to send a URL query string to a Web server
- . ⑩ The way in which PHP reads data from the URL query string can be clearly explained
- . ⑩ PHP server-side scripts can be appropriately written to process data sent in a URL query string
- . ⑩ HTML syntax passing multiple values to a PHP script can be appropriately demonstrated
- . ⑩ HTML forms can be created and the ACTION parameter can be appropriately set to pass the form data to a PHP server-side script
- . ⑩ PHP server-side scripts can be created to appropriately process the data sent in an HTML form

Module Topics

- . ⑩ 4.1 Anatomy of an HTTP Message
- . ⑩ 4.2 Using the URL Query String to Send Data to the Server
- . ⑩ 4.3 Creating PHP Server-side Scripts to Process User Data
- . ⑩ 4.4 Using HTML Forms to Pass User Data

Learning Outcomes

- . ⑩ Explain where and how to download a copy of MySQL for UNIX and for Windows operating systems
- . ⑩ Demonstrate starting the MySQL service on the Web server
- . ⑩ Set a root password to restrict access to the MySQL service
- . ⑩ Test the root password by logging into MySQL
- . ⑩ Shut down the SQL service on the Web server

Performance Indicators

- . ⑩ An appropriate copy of MySQL can be downloaded and installed for the operating system being used
- . ⑩ The MySQL service can be successfully started on the Web server
- . ⑩ A root password can be set and tested to restrict access to the MySQL service
- . ⑩ The MySQL service can be successfully shut down on the Web server

Module Topics

- . ⑩ 5.1 Getting a Copy and Installing the MySQL software
- . ⑩ 5.2 Starting and Stopping the MySQL Service on the Server
- . ⑩ 5.3 Setting Security on MySQL

Module 6: Creating Database Objects Using MySQL

Learning Outcomes

- . ⑩ Determine where to run the MySQL client software (local machine or remote Web server)
- . ⑩ Connect to the MySQL client
- . ⑩ Write simple commands using MySQL syntax at the MySQL command line prompt
- . ⑩ Show the databases currently installed on the server
- . ⑩ Create a new database on the MySQL server
- . ⑩ Create database tables and fields on the MySQL server
- . ⑩ Display MySQL database tables and fields on the user screen
- . ⑩ Delete a table and/or database from MySQL

Performance Indicators

- . ⑩ Client software can be successfully run either by connecting to a remote server or on the local machine
- . ⑩ Simple commands can be written using MySQL syntax at the MySQL command prompt
- . ⑩ A list of currently installed MySQL databases can be shown on the user screen
- . ⑩ New databases can be appropriately created on the MySQL server
- . ⑩ Database tables and fields can be appropriately created on the MySQL server
- . ⑩ A list of the tables and fields in a MySQL database can be shown on the user screen
- . ⑩ MySQL databases and tables can be successfully deleted from the server

Module Topics

- . ⑩ 6.1 Starting the MySQL Client
- . ⑩ 6.2 Writing Simple MySQL Commands
- . ⑩ 6.3 Creating Databases, Tables, and Fields
- . ⑩ 6.4 Deleting Databases and Tables

Learning Outcomes

- . ⑩ Insert data into a MySQL database table
- . ⑩ Write simple queries to browse data in a MySQL database table
- . ⑩ Write joining queries using MySQL tables
- . ⑩ Update data in a MySQL database table
- . ⑩ Delete data from a MySQL database table

Performance Indicators

- . ⑩ Data can be appropriately inserted into a MySQL database table
- . ⑩ Simple queries to extract and browse data in a MySQL database table can be successfully written
- . ⑩ Joining queries can be appropriately written to extract data from multiple MySQL database tables
- . ⑩ Data in a MySQL table can be successfully updated
- . ⑩ Data in a MySQL table can be successfully deleted

Module Topics

- . ⑩ 7.1 Writing Simple Select Queries and Joins
- . ⑩ 7.2 Using Data Maintenance Language

Module 8: Creating Database-Driven Web Pages Using PHP and MySQL

Learning Outcomes

- . ⑩ Connect to a MySQL server using PHP syntax
- . ⑩ Perform SQL queries on MySQL databases using PHP syntax
- . ⑩ Return results of a SQL query into a PHP variable
- . ⑩ Use “mysql_query” to detect success or failure of SQL query
- . ⑩ Use “mysql_affected_rows” to determine number of rows affected by SQL insert, update or delete statements
- . ⑩ Display data returned by SQL queries in an HTML page using PHP row-processing syntax embedded in HTML

Performance Indicators

- . ⑩ Connection to a MySQL database can be accomplished using PHP server-side syntax
- . ⑩ SQL queries can be submitted to a MySQL database using PHP syntax
- . ⑩ Data returned from SQL queries can be stored in a PHP variable
- . ⑩ The “mysql_query” function can be used to determine success or failure of a SQL query
- . ⑩ The “mysql_affected_rows” function can be used to determine the number of rows returned by a SQL data maintenance query (insert, update, delete)
- . ⑩ Data returned by a SQL query can be displayed in an HTML page using PHP row-processing syntax embedded in HTML

Module Topics

- . ⑩ 8.1 Using PHP Syntax to Connect to the MYSQL Server
- . ⑩ 8.2 Executing SQL Queries Using PHP Syntax
- . ⑩ 8.3 Working with PHP Functions
- . ⑩ 8.4 Display MYSQL Data in HTML Pages Using PHP

**COURSE 8: CREATING N-TIER WEB APPLICATIONS USING
ACTIVESERVER PAGES AND ACTIVEX DATA OBJECTS**

Overview.....
..... 1

Module 1: ASP Basics
..... 5

**Module 2: Introduction to Active Server Page
Objects..... 6**

Module 3: Writing Output with the Response Object
..... 7

**Module 4: Sending Information to the Server with the Request
Object 8**

Module 5: Maintaining State in a Web Page

..... 9

Module 6: Working with Built-in ASP Components
..... 10

Module 7: Adding Database Connectivity using ActiveX Data Objects and ASP.....
11

Module 8: Using the Recordset Object and SQL to Query Data..... 12

Module 9: Using SQL Statements and Advanced Database Techniques..... 13

Module 10: Building the “Middle Tier”—Creating and Calling Visual Basic COM Components from an ASP Page..... 14

This ten-module course is designed for students who are already familiar with HTML, Visual Basic scripting, and database concepts. Microsoft’s Active Server Page (ASP) technology allows the Internet developer to create browser-independent, dynamic Web pages by combining programmatic code with HTML. Active Server Page technology offers the capability to present and modify data from a variety of different database formats using Microsoft ActiveX Data Objects (ADO). In addition, COM components (independently compiled “mini-applications”) can be used to enforce business rules and perform complicated tasks at the Internet server level. Students will complete hands-on projects utilizing these technologies to create a three-tier client-server Web application. The development tools used in this course will be Microsoft Visual Interdev and

Microsoft Visual Basic.

Instructional Goals

- . ⑩ Explain basic Active Server Page concepts including differences between client-and server-side scripting, scripting languages, client-server model, and the ASP process
- . ⑩ Demonstrate appropriate use of ASP syntax
- . ⑩ Build functional Web pages using ASP objects such as Response, Request, Application, Server, and Session
- . ⑩ Create dynamic Web pages that maintain information using cookies, session, and application level variables
- . ⑩ Demonstrate database connectivity using ActiveX Data Objects
- . ⑩ Demonstrate the use of an independently compiled Visual Basic COM object in a data-driven Web page

Course Prerequisites

- . ⑩ Basic HTML
- . ⑩ Database design and theory
- . ⑩ Web page client-side scripting/programming (including VBScript)

Learning Outcomes

- . ⑩ Explain the concept of an Active Server Page versus a normal HTML page
- . ⑩ Describe the client-server model
- . ⑩ Demonstrate the use of Microsoft Visual Interdev to create Active Server Pages
- . ⑩ Write an ASP page that produces output to the user's browser
- . ⑩ Demonstrate the use of comments in an ASP page
- . ⑩ Explain what is returned to the user browser when an ASP page is returned from the server
- . ⑩ Write an ASP page using variables, control structures, and built-in functions using VBScript
- . ⑩ Define a programming object, its attributes, and usage
- . ⑩ Differentiate between a property, a method, and an event
- . ⑩ List the built-in ASP objects
- . ⑩ Explain the concept of a collection
- . ⑩ Use the Response object to send output to the user
- . ⑩ Explain the purpose of a form on a Web page
- . ⑩ Create a Web page capable of submitting information to the server
- . ⑩ Use the Action property of a form to submit data to the server
- . ⑩ Use client-side validation of form data
- . ⑩ Differentiate between the different types of form fields
- . ⑩ Retrieve the results of a form using the Request object
- . ⑩ Use the query string to send information
- . ⑩ Explain the function of HTTP headers
- . ⑩ Read the HTTP header with Request.ServerVariables
- . ⑩ Read environment variables with Request.ServerVariables

- . ⑩ Explain the use, advantages, and disadvantages of Cookies
- . ⑩ Read and write cookies using the Request object
- . ⑩ Explain the concept of maintaining “state” in a “stateless” environment
- . ⑩ List various methods of maintaining information on the Web

Learning Outcomes, *continued*

- . ⑩ Define the Session object
- . ⑩ Use session variables to maintain data
- . ⑩ Define the Application object
- . ⑩ Use application level variables to maintain data
- . ⑩ Initialize application and session variables using the Global.asa file
- . ⑩ Explain the advantages of using components in application development
- . ⑩ Demonstrate the syntax to instantiate a component in an ASP application
- . ⑩ Use the Microsoft Browser Capabilities component to detect the user’s browser type and properties
- . ⑩ Use the Microsoft FileObject component to read and write text files
- . ⑩ Use the Microsoft AdRotator component to create an advertising banner
- . ⑩ Explain the importance of using databases in Web applications
- . ⑩ Differentiate between using a DSN vs. a DSN-less connection to the database
- . ⑩ Use the properties of the ADO connection object to customize the database connection
- . ⑩ Connect to a relational database using Microsoft ActiveX Data Objects (ADO)
- . ⑩ Read and display data from a database using ASP and Microsoft designer controls
- . ⑩ Insert, update, and delete records from a database using ADO
- . ⑩ Use the Recordset object to return and display records from a database query
- . ⑩ Use the fields collection to examine column values in a query
- . ⑩ Explain the importance of the cursor type and cursor location properties
- . ⑩ Filter Recordsets based on user input
- . ⑩ Display the results of a Recordset on the ASP page using ADO code
- . ⑩ Execute SQL statements using ASP and ADO
- . ⑩ Create SQL statements using the where clause to filter row selection
- . ⑩ Create SQL statements using joins and calculated fields
- . ⑩ Iterate through a Recordset generated by an SQL statement
- . ⑩ Use stored procedures in Microsoft SQL Server and Access
- . ⑩ Use the command object to call stored procedures from an ASP page
- . ⑩ Explain the concept of a reusable component to enforce business rules
- . ⑩ Create a simple COM component using Microsoft Visual Basic
- . ⑩ Compile and register a COM component on the Web server
- . ⑩ Instantiate a COM component in an ASP page
- . ⑩ Test the COM component in the Web application

Performance Indicators

- . ⑩ Dynamic Web pages can be built incorporating correct use of Active Server Page objects for user input and output
- . ⑩ Dynamic Web pages can be designed and created that maintain information through effective use of cookies, and application and session variables
- . ⑩ Database-driven Web applications can be created using ADO and appropriate ASP objects
- . ⑩ Visual Basic COM components can be successfully added to a data-driven Web application to create a three-tier client-server system

Module Titles

1. 1. ASP Basics
2. 2. Introduction to Active Server Page Objects
3. 3. Writing Output with the Response Object
4. 4. Sending Information to the Server with the Request Object
5. 5. Maintaining State in a Web Page
6. 6. Working with Built-in ASP Components
7. 7. Adding Database Connectivity using ActiveX Data Objects and ASP
8. 8. Using the Recordset Object and SQL to Query Data
9. 9. Using SQL Statements and Advanced Database Techniques
10. 10. Building the “Middle Tier”—Creating and Calling Visual Basic COM Components from an ASP Page

Learning Outcomes

- . ⑩ Explain the concept of an Active Server Page versus a normal HTML page
- . ⑩ Explain the client-server model
- . ⑩ Demonstrate the use of Microsoft Visual Interdev to create Active Server Pages
- . ⑩ Write an ASP page that produces output to the user’s browser
- . ⑩ Demonstrate the use of comments in an ASP page
- . ⑩ Explain what is returned to the user browser when an ASP page is returned from the server
- . ⑩ Write an ASP page using variables, control structures, and built-in functions using VBScript

Performance Indicators

- . ⑩ Purpose and concept of Active Server Pages can be clearly explained
- . ⑩ Client-server model can be clearly defined
- . ⑩ Microsoft Visual Interdev can be used to create and browse a simple ASP page
- . ⑩ ASP page can be created to display appropriate output on the user’s browser screen
- . ⑩ ASP pages can be written to demonstrate appropriate use of variables, control structures, and built-in functions using VBScript

Module Topics

- . ⑩ 1.1 What are Active Server Pages?

- . ⑩ 1.2 Using Microsoft Visual Interdev to Create Active Server Pages
- . ⑩ 1.3 Creating and Testing a Simple ASP Page Using VBScript
- . ⑩ 1.4 Using Variables, Control Structures, and Functions in VBScript

Module 2: Introduction to Active Server Page Objects

Learning Outcomes

- . ⑩ Define a programming object, its attributes, and usage
- . ⑩ Differentiate between a property, a method, and an event
- . ⑩ List the built-in ASP objects
- . ⑩ Explain the concept of a collection

Performance Indicators

- . ⑩ Attributes and usage of programming objects can be clearly defined
- . ⑩ Properties, methods, and events can be explained and demonstrated
- . ⑩ Built-in ASP objects can be listed and explained
- . ⑩ Collections can be clearly explained

Module Topics

- . ⑩ 2.1 Introduction to Programming Objects
- . ⑩ 2.2 Built-in ASP Objects
- . ⑩ 2.3 Collections
- . ⑩ 2.4 Object Events

Learning Outcomes

- . ⑩ Use the Response object to send output to the user
- . ⑩ Explain the purpose of a form on a Web page
- . ⑩ Create a Web page capable of submitting information to the server
- . ⑩ Use the Action property of a form to submit data to the server
- . ⑩ Use client-side validation of form data
- . ⑩ Differentiate between the different types of form fields

Performance Indicators

- . ⑩ ASP pages writing output on the user screen can be created using the Response object
- . ⑩ The purpose of a form on a Web page can be clearly explained
- . ⑩ Web pages using forms for user input can be created appropriately
- . ⑩ Form data can be validated using client-side script
- . ⑩ Different types of form fields can be clearly differentiated

Module Topics

- . ⑩ 3.1 Using the Response Object
- . ⑩ 3.2 Creating Forms for User Input
- . ⑩ 3.3 Validating Form Data
- . ⑩ 3.4 Identifying Data Using Form Fields

Module 4: Sending Information to the Server with the Request Object

Learning Outcomes

- . ⑩ Retrieve the results of a form using the Request object
- . ⑩ Use the query string to send information
- . ⑩ Explain the function of HTTP headers
- . ⑩ Read the HTTP header with Request.ServerVariables
- . ⑩ Read environment variables with Request.ServerVariables
- . ⑩ Explain the use, advantages, and disadvantages of Cookies
- . ⑩ Read and write cookies using the Request object

Performance Indicators

- . ⑩ ASP pages can be written to retrieve user input values on a form using the Request object
- . ⑩ The query string can be used to retrieve user information in a Web page
- . ⑩ HTTP headers can be read using the Request.ServerVariables object
- . ⑩ The use, advantages, and disadvantages of cookies can be clearly explained
- . ⑩ ASP pages can be written using the Request object to read and write cookies

Module Topics

- . ⑩ 4.1 Using the Request Object to Retrieve User Input
- . ⑩ 4.2 Using Query String to Retrieve User Input
- . ⑩ 4.3 Working with HTTP Headers
- . ⑩ 4.4 Creating and Testing a Simple ASP Page Using VBScript
- . ⑩ 4.5 Working with Cookies

Learning Outcomes

- . ⑩ Explain the concept of maintaining “state” in a “stateless” environment
- . ⑩ List various methods of maintaining information on the Web
- . ⑩ Define the Session object
- . ⑩ Use session variables to maintain data
- . ⑩ Define the Application object
- . ⑩ Use application level variables to maintain data
- . ⑩ Initialize application and session variables using the Global.asa file

Performance Indicators

- . ⑩ The concept of maintaining “state” in a Web page can be thoroughly explained
- . ⑩ The most commonly used methods of maintaining information on a Web page can be listed
- . ⑩ The Session object can be clearly defined
- . ⑩ ASP pages can be built demonstrating the correct use of session variables
- . ⑩ The Application object can be clearly defined
- . ⑩ ASP pages can be built demonstrating the correct use of application level variables
- . ⑩ Application and session variables can be correctly initialized in the

Global.asa file

Module Topics

- . ⑩ 5.1 Maintaining “State” in a “Stateless” Environment
- . ⑩ 5.2 Methods for Creating Persistent Information in a Web Page
- . ⑩ 5.3 Working with the Session Object
- . ⑩ 5.4 Working with the Application Object
- . ⑩ 5.5 Using the Global.asa File

Module 6: Working with Built-in ASP Components

Learning Outcomes

- . ⑩ Explain the advantages of using components in application development
- . ⑩ Demonstrate the syntax to instantiate a component in an ASP application
- . ⑩ Use the Microsoft Browser Capabilities component to detect the user’s browser type and properties
- . ⑩ Use the Microsoft FileObject component to read and write text files
- . ⑩ Use the Microsoft AdRotator component to create an advertising banner

Performance Indicators

- . ⑩ Advantages of using components for application development can be clearly explained
- . ⑩ Appropriate syntax to instantiate a component in an ASP application can be demonstrated
- . ⑩ The Microsoft Browser Capabilities component can be appropriately utilized in an ASP page to detect the user’s browser
- . ⑩ The Microsoft File Object component can be appropriately utilized in an ASP page to create a file containing a hit counter
- . ⑩ The Microsoft Ad Rotator component can be appropriately utilized in an ASP page

Module Topics

- . ⑩ 6.1 Using Components in Application Development
- . ⑩ 6.2 Detecting the User’s Browser
- . ⑩ 6.3 Reading and Writing Text Files from a Web Page
- . ⑩ 6.4 Adding Advertising Banners

Learning Outcomes

- . ⑩ Explain the importance of using databases in Web applications
- . ⑩ Differentiate between using a DSN vs. a DSN-less connection to the database
- . ⑩ Use the properties of the ADO connection object to customize the database connection
- . ⑩ Connect to a relational database using Microsoft ActiveX Data Objects (ADO)
- . ⑩ Read and display data from a database using ASP and Microsoft designer

controls

- . ⑩ Insert, update, and delete records from a database using ADO

Performance Indicators

- . ⑩ Importance of database connectivity in Web applications can be clearly explained
- . ⑩ The properties of an ADO connection object can be set appropriately to create DSN and DSN-less connections
- . ⑩ A connection to a database can be created using the ADO connection object and ADO code in an ASP page
- . ⑩ Data from database tables can be read and displayed in a Web page using ASP and Microsoft Designer Controls
- . ⑩ Data can be inserted, changed, and deleted from database tables using ADO code in an ASP page

Module Topics

- . ⑩ 7.1 The Importance of Using Databases in a Web Application
- . ⑩ 7.2 Connecting to the Database Using ADO and ASP
- . ⑩ 7.3 Reading and Displaying Data from a Database
- . ⑩ 7.4 Inserting, Updating, and Deleting Database Records

Module 8: Using the Recordset Object and SQL to Query Data

Learning Outcomes

- . ⑩ Use the Recordset object to return and display records from a database query
- . ⑩ Use the fields collection to examine column values in a query
- . ⑩ Explain the importance of the cursor type and cursor location properties
- . ⑩ Filter Recordsets based on user input
- . ⑩ Display the results of a Recordset on the ASP page using ADO code

Performance Indicators

- . ⑩ Records from a database query can be displayed using the Recordset object and ADO code
- . ⑩ Recordset fields collection can be appropriately utilized to manipulate column values
- . ⑩ The importance of the cursor type and cursor location properties of a Recordset can be clearly explained
- . ⑩ Custom Recordsets can be created based on user input
- . ⑩ Recordset results can be set to display in an ASP page using ADO code

Module Topics

- . ⑩ 8.1 Using the Recordset Object
- . ⑩ 8.2 Working with Database Fields
- . ⑩ 8.3 Creating Custom Recordsets with User Input
- . ⑩ 8.4 Displaying Results using ADO Code

Learning Outcomes

- . ⑩ Execute SQL statements using ASP and ADO
- . ⑩ Create SQL statements using the where clause to filter row selection
- . ⑩ Create SQL statements using joins and calculated fields
- . ⑩ Iterate through a Recordset generated by a SQL statement
- . ⑩ Use stored procedures in Microsoft SQL Server and Access
- . ⑩ Use the command object to call stored procedures from an ASP page

Performance Indicators

- . ⑩ Syntactically correct SQL statements can be created and executed using ASP and ADO
- . ⑩ SQL select statements using the where clause can be created to filter row selection
- . ⑩ SQL select statements can be appropriately created to create joins and calculated fields
- . ⑩ The command object can be used to call stored procedures in an ASP page

Module Topics

- . ⑩ 9.1 Executing SQL Statements Using ASP and ADO
- . ⑩ 9.2 Working with SQL Select Statements
- . ⑩ 9.3 Displaying the Data Returned by a Query
- . ⑩ 9.4 Working with Stored Procedures and the Command Object

Module 10: Building the “Middle Tier”—Creating and Calling Visual

Basic COM Components from an ASP Page

Learning Outcomes

- . ⑩ Explain the concept of a reusable component to enforce business rules
- . ⑩ Create a simple COM component using Microsoft Visual Basic
- . ⑩ Compile and register a COM component on the Web server
- . ⑩ Instantiate a COM component in an ASP page
- . ⑩ Test the COM component in the Web application

Performance Indicators

- . ⑩ Business rules and the purpose of COM components in an n-tier application can be clearly explained
- . ⑩ A simple COM component can be coded and compiled using Microsoft Visual Basic
- . ⑩ A compiled component can be successfully registered and instantiated in an ASP page
- . ⑩ The Web application can be tested to ensure the component is working appropriately

Module Topics

- . ⑩ 10.1 Determining Business Rules and Working with N-tier Applications
- . ⑩ 10.2 Building COM Components in Visual Basic
- . ⑩ 10.3 Incorporating the Component in an ASP Page
- . ⑩ 10.4 Testing Your Application

COURSE 9: INTRODUCTION TO JAVA PROGRAMMING

Overview.....
..... 1

Module 1: Java
Basics..... 5

Module 2: Java Variables and Data Types
..... 6

Module 3: Control Structures
..... 7

Module 4: Java
Methods..... 8

Module 5: Using Objects and
Classes..... 9

**Module 6: Arrays and
Strings..... 10**

**Module 7: Class
Inheritance..... 11**

**Module 8: Programming
Graphics..... 12**

**Module 9: Creating User Interfaces
..... 13**

**Module 10: Applets and Advanced Graphics
..... 14**

This ten-module course provides an introduction to the Java programming language for the Web administration student. Java is a full-featured, general purpose language that can be used to create stand-alone applications. It has become the de facto standard for cross-platform Internet applications and programming on the World Wide Web. Upon completion, students will have a thorough understanding of the basic components of the Java language. Course content includes basic syntax, control structures, Java methods, objects and classes, graphics, interfaces, and applets. Students will have the opportunity to practice with projects and examples in all of these areas.

Instructional Goals

- . ⑩ Create and compile a simple Java program using variables and control structures
- . ⑩ Create, call, and debug Java methods
- . ⑩ Use built-in Java objects and classes

- . ⑩ Create Java applications using arrays and strings
- . ⑩ Develop applications using graphics
- . ⑩ Create user interfaces using Java
- . ⑩ Create Java applets

Course Prerequisites

- . ⑩ Fundamental knowledge of programming concepts and terms
- . ⑩ Knowledge of Web page development and Internet terminology

Learning Outcomes

- . ⑩ Explain the history of the Java language
- . ⑩ List the major features of Java:
 - . ⑩ Object-oriented
 - . ⑩ Platform independence
 - . ⑩ Others
- . ⑩ Discuss the similarities and differences between Java applications and applets
- . ⑩ Examine simple examples of Java applications and applets
- . ⑩ Describe the procedure for viewing applets on a Web browser
- . ⑩ Discuss the terms identifier and variable
- . ⑩ Define numeric and character data types supported by Java
- . ⑩ Declare and initialize syntactically correct variables
- . ⑩ Define the Java programming operators and operator precedence
- . ⑩ Discuss examples of compilation, runtime, and logic errors
- . ⑩ Discuss programming style and documentation procedures
- . ⑩ Explain program flow control
- . ⑩ Create a simple If...Then branch using Java code
- . ⑩ Create an If...Then...Else branch using Java code
- . ⑩ Create complex logic using nested IF statements
- . ⑩ Use the “:” and “?” shortcut syntax to create branching statements
- . ⑩ Create case logic using the switch statement
- . ⑩ Demonstrate looping structures using for, while, and do
- . ⑩ Create exit logic using break and continue keywords
- . ⑩ Explain the concept of methods
- . ⑩ Create a method using Java code
- . ⑩ Call a method and pass parameters
- . ⑩ Create methods that use different types of parameters by using method overloading
- . ⑩ Create methods in separate classes
- . ⑩ Use method abstraction to hide programming complexity from the user
- . ⑩ Debug Java methods
- . ⑩ Explain the relationship between objects and classes
- . ⑩ Define a class and create an object of that class
- . ⑩ Explain the roles of constructors and modifiers

Learning Outcomes, *continued*

- . ⑩ Pass objects to methods
- . ⑩ Organize classes into packages
- . ⑩ Demonstrate the use of instance and class variables and methods
- . ⑩ Define the organization of the Java API
- . ⑩ Demonstrate the use of the Math class
- . ⑩ Explain the concept of an array
- . ⑩ Demonstrate array declaration, creation, initialization, and processing
- . ⑩ Create sorting and searching algorithms
- . ⑩ Use objects as array elements
- . ⑩ Create multi-dimensional arrays
- . ⑩ Demonstrate processing of strings
- . ⑩ Use the string class, String Buffer class, and StringTokenizer class
- . ⑩ Use command-line arguments
- . ⑩ Explain the concept of class inheritance and the relationship between superclasses and subclasses
- . ⑩ Create new classes from existing ones
- . ⑩ Use the keywords and modifiers in creating classes
- . ⑩ Use casting with objects
- . ⑩ Use number class and its subclasses
- . ⑩ Use the Java date and time classes
- . ⑩ Discuss the concept of interfaces
- . ⑩ Describe the Abstract Window Toolkit class hierarchy
- . ⑩ Describe the concept of event-driven programming
- . ⑩ Explain Java event delegation model
- . ⑩ Use frames, panels, and simple user interface components
- . ⑩ Explain the role of layout managers
- . ⑩ Use various layout managers
- . ⑩ Use the Graphics class and its methods
- . ⑩ Use the color and font classes
- . ⑩ Explain components used to create user interfaces (buttons, labels, etc.).
- . ⑩ Create user interfaces using interface components
- . ⑩ Use the listener interface for UI components
- . ⑩ Create multiple windows in an application
- . ⑩ Explain how the browser controls and executes the applet
- . ⑩ Use the init, start, stop, and destroy methods of the applet class
- . ⑩ Pass parameters to applets
- . ⑩ Convert applications to applets
- . ⑩ Handle mouse events and keyboard events
- . ⑩ Explain when to use layout managers

Performance Indicators

- . ⑩ Simple Java programs can be created and compiled using variables and control structures
- . ⑩ Java methods can be created, called, and debugged
- . ⑩ Java applications can be built using objects and classes
- . ⑩ Java applications can be created using strings and arrays

- . ⑩ Graphics can be programmed using appropriate Java syntax
- . ⑩ User interfaces can be developed using Java
- . ⑩ Java applets can be successfully created

Module Titles

1. 1. Java Basics
2. 2. Java Variables and Data Types
3. 3. Control Structures
4. 4. Java Methods
5. 5. Using Objects and Classes
6. 6. Arrays and Strings
7. 7. Class Inheritance
8. 8. Programming Graphics
9. 9. Creating User Interfaces
10. 10. Applets and Advanced Graphics

Learning Outcomes

- . ⑩ Explain the history of the Java language
- . ⑩ List the major features of Java:
 - . ⑩ Object-oriented
 - . ⑩ Platform independence
 - . ⑩ Others
- . ⑩ Discuss the similarities and differences between Java applications and applets
- . ⑩ Examine simple examples of Java applications and applets
- . ⑩ Describe the procedure for viewing applets on a Web browser

Performance Indicators

- . ⑩ History of Java language can be clearly explained
- . ⑩ Major features of the Java language can be listed
- . ⑩ Similarities and differences between Java applications and applets can be thoroughly discussed
- . ⑩ Simple examples of Java applications and applets can be thoroughly examined
- . ⑩ Procedure for viewing applets on a Web browser can be clearly described

Module Topics

- . ⑩ 1.1 History of Java
- . ⑩ 1.2 Characteristics of the Java Language
- . ⑩ 1.3 Applications vs. Applets
- . ⑩ 1.4 Viewing Applets on the Web Browser

Module 2: Java Variables and Data Types

Learning Outcomes

- . ⑩ Discuss the terms identifier and variable

- . ⑩ Define numeric and character data types supported by Java
- . ⑩ Declare and initialize syntactically correct variables
- . ⑩ Define the Java programming operators and operator precedence
- . ⑩ Discuss examples of compilation, runtime, and logic errors
- . ⑩ Discuss programming style and documentation procedures

Performance Indicators

- . ⑩ The terms identifier and variable can be thoroughly discussed
- . ⑩ Numeric and character data types can be clearly defined
- . ⑩ Syntactically correct variables can be declared and initialized
- . ⑩ Java programming operators and operator precedence can be clearly defined
- . ⑩ Examples of compilation, runtime, and logic errors can be clearly discussed
- . ⑩ Issues of programming style and program documentation procedures can be thoroughly discussed

Module Topics

- . ⑩ 2.1 Variables and Identifiers
- . ⑩ 2.2 Numeric and Character Data Types
- . ⑩ 2.3 Declaring and Assigning Values to Variables
- . ⑩ 2.4 Working with Programming Operators and Precedence
- . ⑩ 2.5 Compiling and Running Java Applications
- . ⑩ 2.6 Programming Style and Documentation

Learning Outcomes

- . ⑩ Explain program flow control
- . ⑩ Create a simple If...Then branch using Java code
- . ⑩ Create an If...Then...Else branch using Java code
- . ⑩ Create complex logic using nested IF statements
- . ⑩ Use the “:?” shortcut syntax to create branching statements
- . ⑩ Create case logic using the switch statement
- . ⑩ Demonstrate looping structures using for, while, and do
- . ⑩ Create exit logic using break and continue keywords

Performance Indicators

- . ⑩ Program flow control can be clearly explained
- . ⑩ Program code using simple If...Then branching logic can be appropriately created
- . ⑩ Program code using If...Then...Else branching logic can be appropriately created
- . ⑩ Program code using nested IF branching logic can be appropriately created
- . ⑩ The “:?” shortcut branching syntax can be used appropriately
- . ⑩ Case logic using the switch statement can be appropriately created
- . ⑩ Looping structures using for, while, and do can be appropriately demonstrated

- . ⑩ Looping structures can be appropriately exited using break and continue keywords

Module Topics

- . ⑩ 3.1 Understanding Program Flow Control
- . ⑩ 3.2 Developing Decision Structures Using the If... Statement
- . ⑩ 3.3 Developing Case Structures Using Switch
- . ⑩ 3.4 Developing Looping Logic

Module 4: Java Methods

Learning Outcomes

- . ⑩ Explain the concept of methods
- . ⑩ Create a method using Java code
- . ⑩ Call a method and pass parameters
- . ⑩ Create methods that use different types of parameters by using method overloading
- . ⑩ Create methods in separate classes
- . ⑩ Use method abstraction to hide programming complexity from the user
- . ⑩ Debug Java methods

Performance Indicators

- . ⑩ The concept and purpose of Java methods can be clearly explained
- . ⑩ Methods can be appropriately created using Java code
- . ⑩ Methods can be called and parameters can be appropriately passed
- . ⑩ Methods of the same name using different parameter types can be created to achieve method overloading
- . ⑩ Methods can be created in separate classes
- . ⑩ Method abstraction can be appropriately used
- . ⑩ Java methods can be thoroughly debugged

Module Topics

- . ⑩ 4.1 Introduction to Methods
- . ⑩ 4.2 Creating a Method
- . ⑩ 4.3 Calling a Method
- . ⑩ 4.4 Method Overloading
- . ⑩ 4.5 Developing Methods in Separate Classes
- . ⑩ 4.6 Implementing Method Abstraction
- . ⑩ 4.7 Debugging Methods

Learning Outcomes

- . ⑩ Explain the relationship between objects and classes
- . ⑩ Define a class and create an object of that class
- . ⑩ Explain the roles of constructors and modifiers
- . ⑩ Pass objects to methods
- . ⑩ Organize classes into packages

- . ⑩ Demonstrate the use of instance and class variables and methods
- . ⑩ Define the organization of the Java API
- . ⑩ Demonstrate the use of the Math class

Performance Indicators

- . ⑩ Relationship between objects and classes is clearly explained
- . ⑩ Class modules and objects of class modules are appropriately defined
- . ⑩ The roles of constructors and modifiers are clearly explained
- . ⑩ Objects are appropriately passed to methods
- . ⑩ Classes are organized in packages
- . ⑩ Use of instance and class variables and methods is appropriately demonstrated
- . ⑩ Organization of the Java API is clearly defined
- . ⑩ The Java Math class is appropriately demonstrated

Module Topics

- . ⑩ 5.1 Understanding Objects and Classes
- . ⑩ 5.2 Creating Classes and Objects
- . ⑩ 5.3 Creating Packages
- . ⑩ 5.4 Using Instance and Class Variables
- . ⑩ 5.5 Java API
- . ⑩ 5.6 Using the Java Math Class

Module 6: Arrays and Strings

Learning Outcomes

- . ⑩ Explain the concept of an array
- . ⑩ Demonstrate array declaration, creation, initialization, and processing
- . ⑩ Create sorting and searching algorithms
- . ⑩ Use objects as array elements
- . ⑩ Create multi-dimensional arrays
- . ⑩ Demonstrate processing of strings
- . ⑩ Use the string class, StringBuffer class, and StringTokenizer class
- . ⑩ Use command-line arguments

Performance Indicators

- . ⑩ Concept of an array can be clearly explained
- . ⑩ Array declaration, creation, initialization, and processing can be clearly demonstrated
- . ⑩ Sorting and searching algorithms can be appropriately created
- . ⑩ Objects can be appropriately used as array elements
- . ⑩ Multi-dimensional arrays can be appropriately created
- . ⑩ Processing strings can be correctly demonstrated
- . ⑩ Use of string classes can be appropriately demonstrated
- . ⑩ Command line arguments can be correctly used

Module Topics

- . ⑩ 6.1 Introducing Arrays
- . ⑩ 6.2 Working with Arrays
- . ⑩ 6.3 Sorting and Searching Algorithms
- . ⑩ 6.4 String Processing
- . ⑩ 6.5 Using Command-Line Arguments

Learning Outcomes

- . ⑩ Explain the concept of class inheritance and the relationship between super and subclasses
- . ⑩ Create new classes from existing ones
- . ⑩ Use the keywords and modifiers in creating classes
- . ⑩ Use casting with objects
- . ⑩ Use number class and its subclasses
- . ⑩ Use the Java date and time classes
- . ⑩ Discuss the concept of interfaces

Performance Indicators

- . ⑩ The concept of class inheritance and super and sub class relationship can be clearly explained
- . ⑩ New classes can be appropriately created from existing classes
- . ⑩ Keywords and modifiers can be appropriately used in creating classes
- . ⑩ Casting can be appropriately used to change the type of classes
- . ⑩ Number class and its subclasses can be used appropriately
- . ⑩ Java date and time classes can be used appropriately
- . ⑩ The concept of user interfaces can be thoroughly discussed

Module Topics

- . ⑩ 7.1 Introduction to Class Inheritance
- . ⑩ 7.2 Creating New Classes from Existing Classes
- . ⑩ 7.3 Using Keywords and Modifiers
- . ⑩ 7.4 Casting and Classes
- . ⑩ 7.5 Working With Number Classes and Date/Time Classes
- . ⑩ 7.6 User Interface Concepts

Module 8: Programming Graphics

Learning Outcomes

- . ⑩ Describe the Abstract Window Toolkit class hierarchy
- . ⑩ Describe the concept of event-driven programming
- . ⑩ Explain Java event delegation model
- . ⑩ Use frames, panels, and simple user interface components
- . ⑩ Explain the role of layout managers
- . ⑩ Use various layout managers
- . ⑩ Use the Graphics class and its methods
- . ⑩ Use the color and font classes

Performance Indicators

- ⑩ Abstract Window Toolkit class hierarchy can be clearly described
- ⑩ Concept of event-driven programming can be clearly described
- ⑩ Java event delegation model can be clearly explained
- ⑩ Frames, panels, and user interface components can be appropriately used
- ⑩ Layout managers can be appropriately used
- ⑩ Java graphics class, font, and color classes can be appropriately used

Module Topics

- ⑩ 8.1 Abstract Window Toolkit Concepts
- ⑩ 8.2 Event-driven Programming
- ⑩ 8.3 Handling Events
- ⑩ 8.4 Using Layout Managers
- ⑩ 8.5 Programming Frames and Panels
- ⑩ 8.6 Using the Java Graphics Class
- ⑩ 8.7 Using the Font and Color Classes

Learning Outcomes

- ⑩ Explain components used to create user interfaces:
 - ⑩ Buttons
 - ⑩ Labels
 - ⑩ Etc.
- ⑩ Create user interfaces using interface components
- ⑩ Use the listener interface for UI components
- ⑩ Create multiple windows in an application

Performance Indicators

- ⑩ Commonly used components for creating use interfaces can be clearly explained
- ⑩ User-friendly interfaces can be created using UI components
- ⑩ The listener interface can be created for UI components
- ⑩ Multiple windows can be created for an application

Module Topics

- ⑩ 9.1 Introduction to User Interface Components
- ⑩ 9.2 Using Components to Build Interfaces
- ⑩ 9.3 Creating Interfaces with Multiple Windows

Module 10: Applets and Advanced Graphics

Learning Outcomes

1. ⑩ Explain how the browser controls and executes the applet
2. ⑩ Use the init, start, stop and destroy methods of the applet class
3. ⑩ Pass parameters to applets

4. ⑩ Convert applications to applets
5. ⑩ Handle mouse events and keyboard events
6. ⑩ Explain when to use layout managers

Performance Indicators

1. ⑩ Web browser applet execution and control can be clearly explained
2. ⑩ Common methods of the applet class can be appropriately used
3. ⑩ Parameters can be passed appropriately to Java applets
4. ⑩ Mouse and keyboard events can be correctly handled
5. ⑩ Use of layout managers can be clearly explained

Module Topics

1. ⑩ 10.1 Introduction to the Applet Class
2. ⑩ 10.2 Using Common Methods of the Applet Class
3. ⑩ 10.3 Passing Parameters to Applets
4. ⑩ 10.4 Responding to Mouse and Keyboard Events
5. ⑩ 10.5 When to Use Layout Managers

COURSE 10: BUILDING WEB APPLICATIONS USING XML

Overview.....
..... 1

Module 1:XML Basics
..... 5

Module 2: Overview of XML Software
..... 6

Module 3: Developing Document Type Definitions
..... 7

Module 4: Working with Existing Document Type Definitions
..... 8

Module 5: Writing an XML Document..... 9

Module 6: Converting HTML documents to XML..... 10

Module 7: Creating an XML Database
..... 11

Module 8: Using XML Links..... 12

Module 9: Creating Output using XML Styles
..... 13

Module 10: Understanding XSL
..... 14

This ten-module course is designed for the student who is already familiar with HTML

and database concepts. Extensible Markup Language (XML), an extension of Standard Generalized Markup Language (SGML), can be used to create Web pages, structure page layouts with frames, tables, and forms; link to external sources, enhance content using style sheets, and incorporate database content. In addition, XML can be used to create custom languages. This course incorporates all of these technologies and gives the student the opportunity to practice them through projects and exercises. An introduction to Extensible Stylesheet Language (XSL), which incorporates style sheet properties to enhance XML documents, is also included in the course.

Instructional Goals

1. ⑩ Demonstrate XML basic syntax including elements, tags, and attributes using XML software
2. ⑩ Create new and modify existing Document Type Definitions and to create custom page formats and database definitions
3. ⑩ Write XML documents to include tables, frames, and database information
4. ⑩ Create XML documents capable of connecting to other documents using links and pointers (including link groups)
5. ⑩ Create enhanced output using styles and XSL language

Course Prerequisites

1. ⑩ Basic HTML
2. ⑩ Database design and theory
3. ⑩ Web page client-side scripting/programming

Learning Outcomes

- ⑩ Explain the purpose and basic structure of an XML document
- ⑩ Describe an XML element and its purpose
- ⑩ Describe the anatomy of XML syntax including valid characters, strings, and groups
- ⑩ Demonstrate the usage and syntax of XML element attributes
- ⑩ Differentiate between markup and character data in an XML document
- ⑩ Demonstrate the usage and syntax of XML entities
- ⑩ Describe document type definitions (DTDs)
- ⑩ Explain the functions and purpose of XML software
- ⑩ Differentiate between the major functions of XML parsers and styling software
- ⑩ Use an XML editor to create XML code
- ⑩ Analyze (parse) the text in an XML document to find references to external documents
- ⑩ Use styles to format an XML document
- ⑩ Describe the purpose and usage of a document type definition
- ⑩ Plan a document type definition for one document or a document set
- ⑩ Create an outline for a document type definition for a single document using conditional sections
- ⑩ Create a document type definition using elements, attributes, and entities

- ⑩ Analyze the differences between creating document type definitions for individual documents and document sets
- ⑩ Outline document type definitions for document sets
- ⑩ Search for libraries of existing document type definitions on the Internet
- ⑩ Evaluate an existing document type definition for adoption in another application
- ⑩ Convert an SGML document type definition to an XML document type definition
- ⑩ Edit an existing XML document type definition
- ⑩ Edit a document to fit an existing document type definition
- ⑩ Explain the difference between a document type definition and an XML document
- ⑩ Create an XML declaration to communicate version information to the browser
- ⑩ Create a root element, child elements, and empty elements
- ⑩ Specify attributes to customize an XML element
- ⑩ Utilize internal and external general entities to place content into a document

Learning Outcomes, *continued*

- ⑩ Create empty elements to add graphics to the document
- ⑩ Create lists, tables, and frames in the XML document
- ⑩ Differentiate between an HTML document and an XML document
- ⑩ Edit the HTML document to include XML declaration information
- ⑩ Specify a root element in order to create a well-formed or valid XML document
- ⑩ Establish proper syntax for Unicode compliance
- ⑩ Modify HTML attributes and entities to conform to XML standards
- ⑩ Explain the purpose of a database document type definition
- ⑩ Create a database document type definition
- ⑩ Convert existing database information into an XML document
- ⑩ Use JavaScript and DHTML to enhance an interactive document
- ⑩ Use forms and controls in an XML document to capture user input
- ⑩ Define the use of extended pointers and extended links
- ⑩ Explain how links work in an XML document
- ⑩ Evaluate the difference between inline and out-of-line links
- ⑩ Describe link attributes and attribute values
- ⑩ Analyze different types of link behaviors
- ⑩ Explain the purpose and format of a style sheet
- ⑩ Analyze page layout and effective design
- ⑩ Associate a style sheet with an XML document
- ⑩ Create style sheets to set page background, margins, and borders
- ⑩ Create style sheets to format paragraphs and page elements
- ⑩ Create style sheets to affect case, and apply spacing and alignment to text
- ⑩ Apply style sheet techniques to change the font, size, weight, and insertion of lines in text

- ⑩ Apply style sheet techniques to embellish and enhance ordered and unordered lists
- ⑩ Apply style sheet techniques to format the borders, size, and content of tables
- ⑩ Explain the purpose and usage of Extensible Stylesheet Language (XSL)
- ⑩ Define the use of declarations in creating a document using the XSL language
- ⑩ Explain the usage of XSL patterns to process and style an XML element
- ⑩ Explain the use of qualifiers to restrict the elements that will be affected by an XSL style
- ⑩ Describe the four types of rectangular areas used by XSL to contain formatting objects

Performance Indicators

- ⑩ Basic XML documents can be written utilizing correct syntactical rules
- ⑩ Document Type Definitions can be created using syntactically and structurally correct database definitions and user-friendly page formats
- ⑩ Existing XML document type definitions can be researched and appropriately applied to existing Web page specifications
- ⑩ XML documents can be written with appropriate use of elements, attributes, frames, tables, and links
- ⑩ Documents can be successfully created using XSL language to enhance output style

Module Titles

1. XML Basics
2. Overview of XML Software
3. Developing Document Type Definitions
4. Working with Existing Document Type Definitions
5. Writing an XML Document
6. Converting HTML Documents to XML
7. Creating an XML Database
8. Using XML Links
9. Creating Output Using XML Styles
10. Understanding XSL

Learning Outcomes

- ⑩ Explain the purpose and basic structure of an XML document
- ⑩ Describe an XML element and its purpose
- ⑩ Describe the anatomy of XML syntax including valid characters, strings, and groups
- ⑩ Demonstrate the usage and syntax of XML element attributes
- ⑩ Differentiate between markup and character data in an XML document
- ⑩ Demonstrate the usage and syntax of XML entities
- ⑩ Describe document type definitions (DTDs)

Performance Indicators

- ⑩ XML documents can be written utilizing correct syntactical rules
- ⑩ Components of XML syntax such as valid characters, strings, entities, and element attributes can be clearly explained
- ⑩ Character data can be clearly differentiated from XML markup syntax
- ⑩ Purpose of a document type definition can be clearly explained

Module Topics

- ⑩ 1.1 Basic XML Structure
- ⑩ 1.2 Creating XML Elements
- ⑩ 1.3 Defining XML Attributes
- ⑩ 1.4 Character Data vs. XML Markup Tags
- ⑩ 1.5 Defining XML Entities
- ⑩ 1.6 Introduction To Document Type Definitions

Module 2: Overview of XML Software

Learning Outcomes

- ⑩ Explain the functions and purpose of XML software
- ⑩ Differentiate between the major functions of XML parsers and styling software
- ⑩ Use an XML editor to create XML code
- ⑩ Analyze (parse) the text in an XML document to find references to external documents
- ⑩ Use styles to format an XML document

Performance Indicators

- ⑩ Functions and purpose of XML software can be clearly explained
- ⑩ Differences between XML parsers and styling software can be clearly established
- ⑩ XML editors can be used to create syntactically correct XML code
- ⑩ XML documents can be thoroughly analyzed for references to external documents
- ⑩ XML styles can be appropriately applied in an XML document

Module Topics

- ⑩ 2.1 Introduction to XML Software
- ⑩ 2.2 Creating an XML Document Using an Editor
- ⑩ 2.3 Using External References
- ⑩ 2.4 Using Styles in an XML Document

Learning Outcomes

- ⑩ Describe the purpose and usage of a document type definition
- ⑩ Plan a document type definition for one document or a document set
- ⑩ Create an outline for a document type definition for a single document using conditional sections

- ⑩ Create a document type definition using elements, attributes, and entities
- ⑩ Analyze the differences between creating document type definitions for individual documents and document sets
- ⑩ Outline document type definitions for document sets

Performance Indicators

- ⑩ Document type definitions can be clearly defined
- ⑩ Document type definitions for single and multiple documents can be appropriately planned
- ⑩ Syntactically correct document type definitions can be created using elements, attributes, and entities
- ⑩ Differences between document type definitions for individual document and document sets can be clearly listed

Module Topics

- ⑩ 3.1 Overview of Document Type Definitions
- ⑩ 3.2 Creating DTDs Using Conditional Sections
- ⑩ 3.3 Defining Elements, Attributes, and Entities in a DTD
- ⑩ 3.4 Creating DTDs Using Document Sets

Module 4: Working with Existing Document Type Definitions

Learning Outcomes

- ⑩ Search for libraries of existing document type definitions on the Internet
- ⑩ Evaluate an existing document type definition for adoption in another application
- ⑩ Convert an SGML document type definition to an XML document type definition
- ⑩ Edit an existing XML document type definition
- ⑩ Edit a document to fit an existing document type definition

Performance Indicators

- ⑩ Successful search of the Internet for existing document type libraries can be conducted
- ⑩ Existing DTDs can be thoroughly analyzed and adopted into a new application
- ⑩ Existing SGML and XML DTDs can be successfully edited
- ⑩ Existing documents can be edited to fit an existing DTD

Module Topics

- ⑩ 4.1 Finding Existing DTDs on the Internet
- ⑩ 4.2 Evaluating and Converting DTDs
- ⑩ 4.3 Converting SGML DTDs to XML
- ⑩ 4.4 Editing DTDs and Documents

Learning Outcomes

- ⑩ Explain the difference between a document type definition and an XML document
- ⑩ Create an XML declaration to communicate version information to the browser
- ⑩ Create a root element, child elements, and empty elements
- ⑩ Specify attributes to customize an XML element
- ⑩ Utilize internal and external general entities to place content into a document
- ⑩ Create empty elements to add graphics to the document
- ⑩ Create lists, tables, and frames in the XML document

Performance Indicators

- ⑩ DTDs and XML documents can be clearly differentiated
- ⑩ An XML declaration can be created to communicate version information to the Web browser and is appropriately applied
- ⑩ Root, child, empty, and graphical elements can be created and used appropriately
- ⑩ Element attributes can be created and successfully applied to an XML element
- ⑩ Lists, tables, and frames can be created and applied to an XML document

Module Topics

- ⑩ 5.1 DTDs vs. XML Documents
- ⑩ 5.2 Communicating Version Information to the Browser
- ⑩ 5.3 Creating Root and Child Elements
- ⑩ 5.4 Using Internal and External Entities
- ⑩ 5.5 Adding Graphics to a Document
- ⑩ 5.6 Using Lists, Tables, and Frames

Module 6: Converting HTML Documents to XML

Learning Outcomes

- ⑩ Differentiate between an HTML document and an XML document
- ⑩ Edit the HTML document to include XML declaration information
- ⑩ Specify a root element in order to create a well-formed or valid XML document
- ⑩ Establish proper syntax for Unicode compliance
- ⑩ Modify HTML attributes and entities to conform to XML standards

Performance Indicators

- ⑩ Differences between an HTML and XML document can be clearly delineated
- ⑩ An HTML document can be edited to include appropriate XML declaration information
- ⑩ Appropriate root elements for well-formed or valid XML documents can be created
- ⑩ Documents using Unicode-compliant syntax can be successfully created

- ⑩ Existing HTML attributes and entities can be modified to conform to XML standards

Module Topics

- ⑩ 6.1 HTML Documents vs. XML Documents
- ⑩ 6.2 Creating the XML Declaration and Root Element
- ⑩ 6.3 Writing Syntax for Unicode Compliance
- ⑩ 6.4 Modifying HTML to Conform to XML Standards

Learning Outcomes

- ⑩ Explain the purpose of a database document type definition
- ⑩ Create a database document type definition
- ⑩ Convert existing database information into an XML document
- ⑩ Use JavaScript and DHTML to enhance an interactive document
- ⑩ Use forms and controls in an XML document to capture user input

Performance Indicators

- ⑩ Purpose of a database DTD can be clearly explained
- ⑩ Database DTD can be successfully created and applied
- ⑩ Existing database information can be appropriately converted into an XML document
- ⑩ Interactive documents can be created using correct JavaScript and DHTML syntax
- ⑩ User input can be successfully captured and stored using forms and controls in an XML document

Module Topics

- ⑩ 7.1 Overview of Database DTD
- ⑩ 7.2 Creating the Database DTD
- ⑩ 7.3 Converting Database Information into XML
- ⑩ 7.4 Creating Interactive Documents

Module 8: Using XML Links

Learning Outcomes

- ⑩ Define the use of extended pointers and extended links
- ⑩ Explain how links work in an XML document
- ⑩ Evaluate the difference between inline and out-of-line links
- ⑩ Describe link attributes and attribute values
- ⑩ Analyze different types of link behaviors

Performance Indicators

- ⑩ Extended pointers and extended links can be used to successfully link to external sources
- ⑩ Differences between inline and out-of-line links can be clearly delineated
- ⑩ Link attributes and attribute values can be thoroughly described

⑩ Link behaviors can be analyzed appropriately and their appropriate uses clearly explained

Module Topics

- ⑩ 8.1 Using Extended Pointers and Extended Links
- ⑩ 8.2 In-line vs. Out-of-line Links
- ⑩ 8.3 Using Link Attributes and Attribute Values
- ⑩ 8.4 Analyzing Link Behaviors

Learning Outcomes

- ⑩ Explain the purpose and the format of a style sheet
- ⑩ Analyze page layout and effective design
- ⑩ Associate a style sheet with an XML document
- ⑩ Create style sheets to set page background, margins, and borders
- ⑩ Create style sheets to format paragraphs and page elements
- ⑩ Create style sheets to affect case, apply spacing and alignment to text
- ⑩ Apply style sheet techniques to change the font, size, weight, and insertion of lines in text
- ⑩ Apply style sheet techniques to embellish and enhance ordered and unordered lists
- ⑩ Apply style sheet techniques to format the borders, size, and content of tables

Performance Indicators

- ⑩ Purpose and format of a style sheet can be clearly explained
- ⑩ Page layout and design can be appropriately analyzed and applied
- ⑩ Style sheets can be appropriately associated with XML documents
- ⑩ Style sheets can be created to format paragraphs, text, pages, lists, and tables to conform to design specifications

Module Topics

- ⑩ 9.1 Introduction to Formatting Output Using Style Sheets
- ⑩ 9.2 Associating the Style Sheet with the XML Document
- ⑩ 9.3 Formatting Page Attributes
- ⑩ 9.4 Formatting Text
- ⑩ 9.5 Enhancing Lists and Tables with Styles

Module 10: Understanding XSL

Learning Outcomes

- ⑩ Explain the purpose and usage of Extensible Stylesheet Language (XSL) language
- ⑩ Define the use of declarations in creating a document using the XSL language
- ⑩ Explain the usage of XSL patterns to process and style an XML element
- ⑩ Explain the use of qualifiers to restrict the elements that will be affected by an XSL style

⑩ Describe the four types of rectangular areas used by XSL to contain formatting objects

Performance Indicators

- ⑩ Purpose and usage of XSL can be clearly explained
- ⑩ XSL declarations can be thoroughly defined
- ⑩ XSL patterns and their relationship to document styles can be clearly explained
- ⑩ Use of qualifiers to restrict affected elements in an XSL style can be clearly articulated
- ⑩ Four types of rectangular formatting container areas can be thoroughly described

Module Topics

- ⑩ 10.1 Introduction To Extensible Stylesheet Language (XSL)
- ⑩ 10.2 Using Patterns and Styles
- ⑩ 10.3 Using Qualifiers in an XSL Style
- ⑩ 10.4 Using XSL Formatting Areas