Web Design Specialist

(Adobe CS5 Web Edition)

Exam Study Guide

Web Design Series

CCSSG-CDWDGD-PR-1102 • Version 1.0 • rd090210

Web Design Specialist
(Adobe CS5 Web Edition)
Exam Study Guide

President/Chief Certification Architect
James Stanger, Ph.D.

Vice President, Operations
Todd Hopkins

Senior Content Developer
Kenneth A. Kozakis

Managing Editor
Susan M. Lane

Editor
Sarah Skodak

Project Manager/Publisher
Tina Strong

Customer Service
Certification Partners, LLC
1230 W. Washington St., Ste. 111
Tempe, AZ 85281
(602) 275-7700

Web Design Specialist Exam Study Guide

(Adobe CS5 Web Edition)

Developers

Irina Heer; Kenneth A. Kozakis; James Stanger, Ph.D.;
Jeffrey Brown; Chris Minnick; and Susan M. Lane

Contributors

Stephen Schneiter, Brian Danks, Martin Heltai and Robert Barrett

Editors

Susan M. Lane and Sarah Skodak

Project Manager/Publisher

Tina Strong

Trademarks

Disclaimer

Copyright Information

ISBN: 0-7423-2803-1

Table of Contents

CIW Web Design Specialist Exam Study Guide Overview

This exam study guide will help you prepare for the CIW Web Design Specialist exam 1D0-520. The CIW Web Design Specialist exam contains the following four topical domains with associated objectives categories and objectives:

- Site Development Essentials
- Web Design Elements
- Basic Web Technologies
- Advanced Web Technologies

CIW Web Design Specialist exam and certification

Those who pass the CIW Web Design Specialist exam (1D0-520) earn the highly respected CIW Web Design Specialist certification, which validates essential Internet and Web development skills for the workplace. Those who also pass the CIW E-Commerce Specialist exam (1D0-525) earn the CIW Web Design Professional certification, which validates advanced skills in Web site and e-commerce solution development. For information about taking CIW exams, visit *www.CIWcertified.com*.

CIW Candidate Certification Agreement form

All CIW exam candidates must complete and submit the CIW Candidate Certification Agreement online form before taking the CIW Web Design Specialist exam. Exam candidates will not receive a CIW certificate without completing the agreement.

You can access the CIW Candidate Certification Agreement form by visiting the CIW Candidate Information Center at *https://i7lp.integral7.com/ciw*. Once you have registered with the CIW program, you can complete and submit the agreement.

CIW Web Design Specialist exam information

The CIW exam candidate is responsible for earning a passing score on the CIW Web Design Specialist exam. CIW exam items are based on CIW exam objectives. All deliveries of the CIW Web Design Specialist exam will be subject to the following:

- Each delivery of the exam includes a random selection of 70 CIW Web Design Specialist exam items.
- The examination period is 75 minutes.

To earn a passing score on the 1D0-520 CIW Web Design Specialist exam, candidates must correctly answer at least 49 of the 70 exam items. Exam items offer four solutions or options. Exam candidates must select the one best solution for each item.

After completing the CIW Web Design Specialist exam, candidates will receive a score report from the testing center. Candidates who pass the CIW Web Design Specialist exam (1D0-520) are CIW-certified and will receive a CIW Web Design Specialist certificate from the CIW Certification department approximately four to six weeks after completing the exam and the CIW Certification Agreement. Those who also pass the CIW E-Commerce Specialist exam (1D0-525) earn the CIW Web Design Professional certification and will receive the appropriate certificates.

You can take CIW exams at participating Prometric and Pearson VUE testing centers. For registration information, please visit the Pearson VUE (*www.vue.com/ciw*) or Prometric (*www.prometric.com/ciw*) Web sites.

Payment arrangements

Payment is required at the time of exam registration, either by voucher or by credit card. (Prometric also accepts checks.) After the payment has cleared, an exam date and time can be scheduled. When you make an exam appointment, you will receive instructions about cancellation procedures and directions to the testing center location. Testing center information is also available through the Prometric Test Center locator (*www.register.prometric.com*) or the Pearson VUE Testing Center locator (*www7.pearsonvue.com/vtclocator/*). The fee for taking the CIW Web Design Specialist exam is U.S. $150. Please contact your local authorized testing center to determine the exact pricing in your local currency.

Taking the CIW Web Design Specialist exam

The following information provides details about taking CIW exams.

Before the exam

You should arrive at the testing center at least 15 minutes before the test is scheduled to begin.

Identify yourself to the administrator. Two forms of identification are required; both forms of identification must carry your signature. One form of identification must have your photograph. Suitable forms of identification can include:

- Driver's license
- Passport
- Government or company ID card
- Major credit card

You will then have the opportunity to learn how to use the computer-based testing system. This tutorial session is not part of the CIW exam, and will not reduce the amount of time you have to complete the exam. Books, calculators, laptop computers, notes, scratch paper and any reference materials are not allowed during any CIW exam. Because CIW exams are computer-based, you will not need pens, pencils or paper to take the examination. Accommodations for the disabled are available with advance notice and approval from Certification Partners and the testing center.

During the exam

The exam will display a brief opening screen and some initial instructions. You may be asked to complete an optional survey. This survey is not a part of the exam, and will not reduce the amount of time you have to complete the CIW exam.

CIW exam results and security

The following information provides additional details about CIW exam results, security and certificates.

CIW exam score report

As soon as you complete the CIW Web Design Specialist certification exam, you will see your score results immediately on your computer screen. You will also receive a hard copy of the score report from the testing center manager. The score report shows whether you passed or failed the exam, and indicates your score on each section of the exam. The score report does not include any information about specific questions. Partial credit is not awarded on any question.

To pass the CIW Web Design Specialist exam, a candidate must:

- Answer all items on the exam.
- Earn a total score that is equal to or more than the passing score for the examination. The passing score is listed in the CIW Web Design Specialist Exam Information section of this guide, and is clearly

indicated on the Prometric exam screens and the official exam score report that are produced when the candidate completes the CIW examination.

CIW exam security

CIW exams are administered under controlled conditions at authorized testing centers. The testing center confirms the identity of exam candidates and provides a secure environment for collecting and transmitting exam results. The testing center and Certification Partners will keep the candidate's scores and information confidential. The CIW program does not release exam items or answers. This procedure is standard for high-stakes, proctored certification exams, and helps maintain the value of the certification. All CIW examinations are copyrighted material. To maintain the security and value of the program, Certification Partners reserves the right to decertify and/or bar from examinations any individuals who republish or distribute the copyrighted certification exam questions. To make a comment or recommendation, send an e-mail message to *exam@ciwcertified.com*.

CIW exam fee receipts

If you need a receipt for your exam fees, you may request the receipt from the testing center manager at the time you take the exam. Certification Partners cannot provide receipts for testing center fees.

CIW certificates

After you pass the exams required for your CIW certification, you will receive your CIW certificate by first-class postal mail in approximately eight weeks. Postal delivery times may vary outside the United States. Certificates are awarded after a candidate has completed the requirements for a certification. If you have not received your certificate within 90 days of completing the certification requirements, contact *exam@ciwcertified.com*. Please do not contact Prometric or Pearson VUE regarding CIW certificates. Include your full name and postal address, and your exam dates and registration numbers in your message.

CIW business card logos

You will receive access to the appropriate CIW logo for your certification in your welcome packet along with your CIW certificate. Visit the CIW Candidate Information Center at *https://i7lp.integral7.com/ciw* to retrieve your logo.

CIW Exam Retake Policy

The CIW Exam Retake Policy outlines the conditions under which a candidate may retake a CIW exam. A waiting period between retakes of a single certification exam maintains the security of the exam and strengthens the value of the CIW certification. Each CIW exam is defined by a unique exam ID (e.g., 1D0-520). A retake is any subsequent sitting of an exam with the same CIW exam ID by the same candidate at any authorized testing center.

In the event that a candidate passes a CIW exam, the candidate will not be allowed to retake that CIW exam. If the CIW exam objectives change, the exam ID changes to reflect the new exam version. Candidates who have passed an exam may take a newer version of the exam that may have the same title but a different ID number.

In the event that a candidate fails a CIW exam on his or her first attempt, a 24-hour waiting period is required between the first and second sittings of that CIW exam. Candidates who fail a second attempt are required to wait for a period of no less than thirty (30) calendar days from the date of the second sitting before any third or subsequent sitting of the same CIW exam.

Exams administered in any ways that do not comply with the CIW Exam Retake Policy shall be considered invalid and ineligible for a refund.

In the event that a candidate has violated this Exam Retake Policy, the candidate may be deemed ineligible to register for or schedule any CIW exam for a minimum period of twelve (12) months from the date of such determination.

In addition, any candidate determined to have violated the CIW Exam Retake Policy may be subject to any or all of the following:

- Denial of a specific CIW certification for a period of twelve (12) months from the date of such determination;
- Revocation of a specific CIW certification, if such certification had been previously granted to the candidate;
- Revocation of all CIW certifications previously granted to the candidate;
- Ineligibility to receive any CIW certification for a minimum of twelve (12) months from the date of such determination;
- Any other appropriate actions, including legal remedies, deemed necessary or appropriate to enforce the CIW Exam Retake Policy.

Comments regarding the operations of a testing center delivering CIW exams should be referred directly to the testing center.

CIW Web Design Specialist exam retirement

No retirement date has been announced for the CIW Web Design Specialist 1D0-520 exam.

CIW Web Design Specialist exam updates

A subject matter expert (SME) team assembled by Certification Partners regularly reviews the CIW Web Design Specialist 1D0-520 exam. Due to continual advances in Internet technology, exam objectives may require significant additions and deletions over time.

Certification Partners' policy is to inform all program partners about potential changes to the exam objectives three months before releasing any updates. After revised exam objectives are released, Certification Partners will modify exams and/or courseware. This process may take between six and 12 months. Until the new exam is published, the existing objectives and exams will continue to be valid and available.

Any CIW certification that you have earned by taking and passing all required exams (and fulfilling any other requirements at the time of completion) will not expire. Once CIW certified, always CIW certified. Whether you choose to re-certify with a newer exam version is your decision, based upon the needs of your career path. However, you are strongly encouraged to update your certification. Holding the most current version of a technology certification shows your employers and colleagues that your skills and knowledge are up-to-date with the most current trends and technology.

Certification Partners has the option to introduce additional CIW exams focusing on specialized areas, or more specific knowledge of one or more technologies covered generally in 1D0-520. If and when such exams are released, candidates will be required to take and pass new specialized exams to earn any associated CIW certification status. Visit the CIW Web site at *www.CIWcertified.com* for more information about current CIW exams.

CIW Web Design Specialist Exam Objectives

The following table lists the full text of the CIW Web Design Specialist exam objectives by objective number. The corresponding abbreviated CIW Web Design Specialist exam objectives, which are used throughout the remainder of this Exam Study Guide, are also provided. All objectives are categorized by domain (e.g., Domain 1: Site Development Essentials) and subdomain (e.g., Domain 1.1: Web site development process) to encourage the use of this table as a reference and study tool.

Objective Number	Abbreviated Objective	Full Objective
Domain 1: Site Development Essentials		
1.1	**Web site development process**	**Identify and manage elements of the Web site development process.**
1.1.1	Web development job tasks	Identify job responsibilities and tasks of a Web designer or Web development team member.
1.1.2	Web design portfolio	Develop and update your Web design portfolio with demonstration pages and sites.
1.1.3	Web project collaboration	Define the collaborative nature of a Web development project.
1.1.4	Determining site audience	Determine the audience for the site.
1.1.5	Web site vision statement	Develop a Web site vision statement.
1.1.6	Site strategy and tactics	Develop a site strategy and identify strategy implementation tactics.
1.1.7	Mindmapping site structure	Use the mindmapping process to structure a Web site.
1.1.8	Design goals for customer and audience	Set design goals appropriate for the business/organization represented by the site and the site's intended audience.
1.1.9	Site metaphor	Create a site metaphor.
1.1.10	Site design and architecture specifications	Develop site design and architecture specifications.
1.1.11	Site project implementation factors	Determine site project implementation factors (e.g., stakeholder input, time frame, scope, desired functionality, required technologies).
1.1.12	Web project development and rollout plan	Create a Web project plan, including development timetable, site rollout plan.
1.1.13	Site characteristics and resources	Identify Web site characteristics (e.g., interactivity, navigation, database integration) and the project resources they require.
1.1.14	Web site branding	Apply branding to a Web site.
1.1.15	Audience usability tests	Conduct audience usability tests.
1.1.16	Flowcharts and Web wireframes	Use flowcharts and Web wireframes to determine page layout.
1.2	**Customer expectations**	**Meet customer expectations with Web site project and design.**
1.2.1	Needs and usability vs. design and aesthetics	Balance customer needs and usability with site design principles and aesthetics (includes distinguishing site design customer from site audience).
1.2.2	Customer expectations and feedback	Document customer expectations and feedback.
1.2.3	Communicating progress to stakeholders	Communicate plans and progress regularly to ensure that completed project meets stakeholder/customer expectations.

Objective Number	Abbreviated Objective	Full Objective
1.2.4	Changes in project scope	Identify and manage changes in project scope, including scope creep.
1.2.5	Changes in development plan	Document changes in development plan.
1.2.6	Project tracking reports	Create a project tracking report.
1.2.7	Project evaluation	Conduct a project evaluation, including acceptance documentation, summary of technologies used, project style guidelines.
1.2.8	Web page and site templates	Create Web page and site templates that fulfill design specifications.
1.2.9	Designing for PDA-based vs. traditional browsers	Identify challenges involved in designing Web pages for PDA-based versus traditional browsers.
1.3	**Ethical and legal issues**	**Identify ethical and legal issues relevant to Web development and design.**
1.3.1	Ethics, legal vs. ethical issues	Define ethics, and distinguish between legal and ethical issues.
1.3.2	Proper content usage	Use Web content (e.g., text, graphics, code) properly, including original content, misleading/inaccurate information, copyrighted content, licensing, avoiding infringement.
1.3.3	Site strategies and technologies to avoid	Identify site strategies and technologies to avoid, including pop-up windows, single-browser sites, spam.
1.3.4	Strategies for end-user privacy and trust	Use strategies to avoid violating end-user privacy and trust (e.g., refusing to share or sell end-user information, opt-in/opt-out for mailing lists).
1.3.5	Privacy disclaimers	Develop privacy disclaimers appropriate to site purpose and audience.
1.3.6	International legal issues	Identify international legal issues, including fair use, trademarks, contracts.
1.3.7	Nature and purpose of site content	Consider nature and purpose of site content (e.g., audience appropriateness, intended vs. unintended audience, potentially offensive content, offensive vs. illegal content, global and cultural perspectives).
Domain 2: Web Design Elements		
2.1	**Aesthetics and viewer experience**	**Use Web design principles to evaluate and develop a site's aesthetic qualities and its ability to enhance viewer experience.**
2.1.1	Web page design and layout elements	Define and use common Web page design and layout elements (e.g., color, space, font size and style, lines, logos, symbols, pictograms, images, stationary features).
2.1.2	Design and audience participation	Determine ways that design helps and hinders audience participation (includes target audience, stakeholder expectations, cultural issues).
2.1.3	Visually balanced page/site	Manipulate space and content to create a visually balanced page/site that presents a coherent, unified message (includes symmetry, asymmetry, radial balance).
2.1.4	Color and contrast	Use color and contrast to introduce variety, stimulate users and emphasize messages.
2.1.5	Design strategies for user focus	Use design strategies to control a user's focus on a page.

Objective Number	Abbreviated Objective	Full Objective
2.1.6	Tools for site's visual consistency	Apply strategies and tools for visual consistency to Web pages and site (e.g., style guides, page templates, image placement, navigation aids).
2.1.7	Site message, culture and tone	Convey a site's message, culture and tone (professional, casual, formal, informal) using images, colors, fonts, content style.
2.1.8	Eliminating unnecessary elements	Eliminate unnecessary elements that distract from a page's message.
2.1.9	Typographical issues in printable content	Design for typographical issues in printable content.
2.1.10	Screen resolution issues	Design for screen resolution issues in online content.
2.1.11	Applying a single W3C standard consistently	Explain the importance of applying a single CSS and HTML standard consistently throughout a site.
2.2	**Navigation, usability and accessibility**	**Use Web design principles to enable navigation, usability and accessibility.**
2.2.1	Site characteristics and strategies	Identify Web site characteristics and strategies to enable them, including interactivity, navigation, database integration.
2.2.2	Site hierarchy/ architecture concepts	Identify Web site hierarchy/architecture concepts, including appropriate page depth for content.
2.2.3	Common navigation conventions	Identify common navigation conventions.
2.2.4	Navigation action plan	Develop and apply a navigation action plan.
2.2.5	Multimedia purpose	Identify purpose and usefulness of multimedia.
2.2.6	User-accessibility standards and laws	Identify and apply user-accessibility standards and laws (e.g., W3C WAI/WCAG, ADA, Section 508, international standards).
2.2.7	User-accessibility challenges and solutions	Identify common user-accessibility challenges and solutions.
2.2.8	Written site content	Develop or obtain written content that conveys the site's message, including clear and concise writing, professional editing, style guides, consistency, jargon, voice and tone (professional, formal, informal).
2.2.9	Audience and end-user capabilities	Identify audience and end-user capabilities (e.g., lowest common denominator in usability).
Domain 3: Basic Web Technologies		
3.1	**Basic X/HTML**	**Use basic HTML and XHTML (X/HTML) to develop a series of Web pages.**
3.1.1	X/HTML origins, standards and versions	Explain the origins of HTML and XHTML, define the X/HTML standards, and distinguish among X/HTML versions.
3.1.2	Basic X/HTML code	Write X/HTML code to create a static Web page with text and images.
3.1.3	Hexadecimal color values	Use hexadecimal values to specify colors in X/HTML.
3.1.4	X/HTML layout elements	Use X/HTML to apply design principles and layout elements (e.g., fonts, space, colors, lines, images) to Web pages.
3.2	**X/HTML and extended technologies**	**Use X/HTML and extended technologies to enhance Web page structure, format and usability.**
3.2.1	X/HTML metadata tags and content	Add metadata tags and content to X/HTML documents to influence search engine placement (includes refining <meta> tags in existing pages).

Objective Number	Abbreviated Objective	Full Objective
3.2.2	Non-standard/proprietary X/HTML code	Identify non-standard X/HTML code and the ways that proprietary code affects Web development.
3.2.3	Web page formatting with CSS1 and CSS2	Identify ways to apply Web page formatting with Cascading Style Sheets (CSS1, CSS2 and CSS3) using various methods (e.g., linking, embedding, inline), and use style sheets to simplify Web site design.
3.2.4	Linking style sheet to X/HTML page	Create an external style sheet and link it to an X/HTML document.
3.2.5	X/HTML tables	Design and develop X/HTML tables to appropriately format data.
3.2.6	X/HTML framesets	Develop X/HTML framesets (including simple, nested, combined, inline), and target frames correctly.
3.3	**Image files and X/HTML**	**Create image files, and use images in X/HTML pages and site design.**
3.3.1	Vector vs. raster graphics	Distinguish between vector and raster graphic types.
3.3.2	Image file formats	Identify and choose appropriate image file formats, including browser-compatibility issues and lowest common denominator in audience usability (e.g., GIF 87a, GIF 89a, JPEG, JPEG 2000, PNG, BMP).
3.3.3	Image-editing software	Use image-editing software to create functional images that complement your page/site.
3.3.4	Image manipulation functions	Perform common image manipulation functions (e.g., cropping, rasterizing, adding text to existing images, modifying height/width dimensions, modifying resolution, choosing bit depths).
3.3.5	Transparent and animated images	Create transparent and animated images, including GIF, PNG.
3.3.6	Image layers	Create image layers using image-editing software.
3.3.7	Image colors and audience cultures	Evaluate image colors to determine effectiveness in various cultures.
3.3.8	Image files in X/HTML	Insert image files in Web pages using X/HTML
3.3.9	Metadata in images	Insert metadata into images to ensure accessibility and to ensure higher page ranking in search engine result pages.
3.3.10	Stock photography in site development	Identify the benefits and drawbacks of using stock photography when developing a site (e.g., license-free vs. licensed stock photos, increase in project speed, reduction in creative control).
3.4	**GUI site development applications**	**Create Web sites using GUI site development applications.**
3.4.1	X/HTML text editors vs. GUI site management applications	Compare site development using X/HTML text editors to using GUI site management applications.
3.4.2	W3C-compliant code with GUI site applications	Configure site development applications to develop W3C-compliant code, including XHTML 1.0 Transitional.
3.4.3	Images with GUI site applications	Add images to Web pages and create image maps using GUI site development applications.
3.4.4	Page text, tables and hyperlinks with GUI applications	Add text, tables and hyperlinks to Web pages using GUI site development applications.
3.4.5	Web forms with GUI site applications	Create Web forms using GUI site development applications.

Objective Number	Abbreviated Objective	Full Objective
3.4.6	Creating templates with GUI site applications	Create page and site templates using GUI site development applications.
3.4.7	CSS in GUI page and site templates	Apply CSS to page and site templates using GUI site development applications.
3.4.8	Applying templates with GUI site applications	Apply page and site templates to new pages using GUI site development applications.
3.4.9	Validating source code with GUI site applications	View and validate source code using GUI site development applications.
3.4.10	Enforcing accessibility standards with GUI site applications	Use GUI site development applications to enforce compliance with accessibility standards.
3.5	**Site publishing and maintenance**	**Publish and maintain a production Web site.**
3.5.1	Staging/mock-up server	Use a staging/mock-up server to test a site, including advantages, hardware/software choices, configurations.
3.5.2	Site testing	Perform site testing (functionality, usability, browser compatibility).
3.5.3	In-house hosting vs. ISP or ASP hosting	Compare in-house Web site hosting to hosting with an Internet Service Provider (ISP) or Application Service Provider (ASP).
3.5.4	Site publishing with FTP	Publish a Web site using an FTP client.
3.5.5	Configuring DNS entries	Create and configure Domain Name System (DNS) entries (includes subdomains, shared domains).
3.5.6	Site security issues	Identify site security issues, including attacks (e.g., social engineering, denial of service, brute force) and ways to thwart them.
3.5.7	Server security	Secure a server, including disabling unnecessary services, updating patch levels, configuring login settings, setting permissions and rights.
3.5.8	Web site maintenance	Maintain the Web site (includes user feedback, auto and manual link checking).
3.5.9	Documenting site changes	Document changes to the site.
Domain 4: Advanced Web Technologies		
4.1	**Multimedia and plug-in technologies**	**Use multimedia and plug-in technologies to enhance a Web site.**
4.1.1	Multimedia Web design principles	Identify multimedia Web design principles, and choose appropriate multimedia technologies for a site based on usability criteria.
4.1.2	Accessibility issues with images and animation	Identify accessibility issues and solutions related to Web images and animation (e.g., text-reader capability, captioning).
4.1.3	SWF technology features and software	Identify Shockwave-Flash (SWF) technology features (e.g., animation, streaming, timelines, layers) and authoring software.
4.1.4	Using SWF-authoring software	Use SWF-authoring software to create animations, add buttons, perform "tweening," create movie clips, apply masks.
4.1.5	SVG characteristics	Identify Scalable Vector Graphics (SVG) characteristics (e.g., XML-based, two-dimensional, searchable, scalable, zoom support).
4.1.6	SWF and SVG files in X/HTML pages	Add SWF animation files and SVG files to X/HTML pages.
4.1.7	SWF and SVG in instructional design	Identify strategies and benefits of using SWF and SVG technologies in training industry/instructional design to facilitate learning.

Objective Number	Abbreviated Objective	Full Objective
4.1.8	Plug-in/viewer technologies	Apply plug-in/viewer technology to Web pages to support various file types (e.g., Portable Document Format [PDF], Scalable Vector Graphics [SVG], Flash/SWF technologies).
4.1.9	X/HTML and downloadable files	Create an X/HTML link to a downloadable file.
4.1.10	Rich media streaming ads	Create rich media streaming ads and compare them to conventional online ads (includes considering bandwidth limitations).
4.1.11	Java applet functionality and use	Define Java applet functionality, and create an animated applet for display on a Web site.
4.2	**Client-side and server-side technologies**	**Use client-side and server-side programming to enhance Web site functionality.**
4.2.1	Client-side vs. server-side technologies	Define and contrast client-side and server-side technologies used to create dynamic content for Web pages.
4.2.2	JavaScript objects, properties and methods	Identify common JavaScript objects, properties and methods.
4.2.3	Using JavaScript	Use JavaScript to detect browsers, redirect pages, preload pages and confirm user choices.
4.2.4	JavaScript dot notation	Use JavaScript dot notation to access X/HTML objects.
4.2.5	Rollover images with scripting technology	Create rollover images on a Web page using scripting technology.
4.2.6	Dynamic HTML (DHTML)	Define Dynamic HTML (DHTML) and the technologies it requires, and write browser-specific DHTML code for use with Firefox, Internet Explorer, and other browsers (e.g., Safari, Chrome, Opera).
4.2.7	XML vs. HTML and XHTML	Define Extensible Markup Language (XML), and distinguish XML from HTML and XHTML.
4.2.8	Well-formed XML documents	Define and create a "well-formed" XML document.
4.2.9	Secure XML	Define Secure XML.
4.2.10	Web form processing with CGI	Use Common Gateway Interface (CGI) to process Web forms.
4.2.11	Site functionality with cookies	Use cookies to enhance Web site functionality.
4.2.12	Pop-up/pop-under windows	Identify the functionality of pop-up/pop-under windows (e.g., creation, benefits, disadvantages, reasons to omit from your site).
4.2.13	CAPTCHA	Define CAPTCHA and create a CAPTCHA for a Web site.
4.2.14	TinyURL	Explain how the TinyURL service works.
4.3	**Web databases**	**Connect Web pages to a database.**
4.3.1	Database elements	Define elements of a database and their functionality.
4.3.2	Database query types	Identify general database query types.
4.3.3	Database Management System (DBMS) types	Define the three Database Management System (DBMS) types.
4.3.4	Adding site search capability	Add search capability to a Web site.
4.3.5	Database information types	Identify information types that can be contained in a database, including X/HTML, images, XML, inventories.

Objective Number	Abbreviated Objective	Full Objective
4.3.6	Connecting Web pages to databases	Connect a Web page to a database using various methods.
4.4	**Internet marketing**	**Conduct effective Internet marketing**
4.4.1	SEO vs. PPC strategies	Discuss organic and non-organic strategies for improving hit rates and search engine ranking (e.g., search engine optimization [SEO] vs. pay per click [PPC} strategies).
4.4.2	SEO terms	Define common search engine optimization (SEO) terms (e.g., keywords, keyword density, stop words, results pages, reciprocal links, black-hat SEO).
4.4.3	Common SEO techniques	Identify common valid SEO techniques (e.g., writing quality Web copy, structuring pages using validated HTML and CSS, using page titles, using metadata, using on-site and off-site practices).
4.4.4	Web analytics	Define Web analytics, including discussing key events to review on a Web server.
4.4.5	Generating revenue with search engines	Identify ways that search engines generate revenue by processing search entries from users.
4.4.6	Blogging	Create a blog-writing strategy to support a brand, including key elements of a successful blog entry (e.g., brevity, central idea, clever title, links to relevant articles, configuring trackbacks).
4.4.7	Web marketing campaigns	Identify elements of a successful Web marketing campaign (e.g., working with marketing team members to determine audience and central message, creating HTML e-mail messages with images, scheduling Webcasts).
4.5	**Syndicated feeds**	**Create syndicated feeds using feed management services.**
4.5.1	Definition of syndication	Define syndication.
4.5.2	RSS and Atom	Use RSS and Atom to create a news feed.
4.5.3	Aggregators	Describe the purpose of an aggregator in a feed.
4.5.4	RSS in e-mail conversion	Identify the benefit of RSS to e-mail conversion, including push-based and pull-based technologies.

Domain 1: Site Development Essentials

This section will discuss CIW Web Design Specialist Domain 1 exam objectives covering fundamental Web site development skills and concepts.

1.1: Web site development process

This subdomain includes skills and knowledge required to identify and manage elements of the Web site development process.

1.1.1: Web development job tasks

The primary job of a Web designer is usually to design a visually appealing Web site that satisfies the requirements specified by the project manager or client. Depending on the organization, the job description of a Web designer will vary widely. No one can possibly be an expert in every aspect of Web development, thus the trend in Web development jobs is toward specialization. However, Web designers are often still expected to have a very wide range of skills. The skills required of a Web designer may include any combination of the following:

- Web site planning based on management or client goals and priorities
- Web graphics design
- Site structuring and navigation scheme design
- XHTML coding
- Programming and scripting languages coding
- Customer service and client relations
- Web site testing, troubleshooting and debugging

For more information about Web development job tasks, see *Web Design Specialist*, Lesson 2: Web Development Teams.

1.1.2: Web design portfolio

A Web design portfolio is a single Web location that allows you to show off your best work to friends, family or potential employers. It is also a helpful tool for tracking your own progress as a Web designer. The purpose of a portfolio is to demonstrate your Web design skills and display your work. The goal is generally to impress your visitors — often people who may decide to hire you. Considering the purpose and goal of your Web design portfolio, you should make sure that it is attractive, easy to use, and up to date. Consider including the following items in your Web portfolio:

- Your résumé
- Samples of graphics you have created
- Links to Web pages or sites you have designed or contributed to
- Demonstrations or samples of any other relevant special skills you want to emphasize, such as drawing or painting, programming, photography, or print design

Your portfolio must also be accurate. Do not link to sites that you have not contributed to or that you did very little work on, unless you have a good reason and you explain it. Also consider the quality of the work you make available. Works in progress or rough draft versions may not be the best work for you to show off, unless you have a specific reason to provide examples of your creation process. Clearly label any older work meant to demonstrate a "before and after" process. Be sure to maintain your portfolio regularly and update it as quickly as possible when anything on it changes.

For more information and a hands-on lab about Web design portfolios, see *Web Design Specialist*, Lesson 2: Web Development Teams.

1.1.3: Web project collaboration

The team approach to Web development is critical to create a well-rounded design and a functional Web site. In Web development, the team's composition often fluctuates from one project to the next, depending on the skills required. Various skill sets are needed on the team depending on the phase of development in the Web project. Also, Web teams are frequently dispersed geographically; it is not uncommon for some members of a Web project team never to meet in person.

Although Web development projects vary, and one person may serve several roles, a Web project team will typically have people performing each of the jobs described in the following table.

Web Development Job	Description
Project management	The project manager is involved with the project from start to finish. He or she is in charge of determining and documenting the requirements of a project, creating specifications for the project, assembling a team, and ensuring that the project stays on schedule and on budget.
Information architecture	The information architect is responsible for translating customer requirements into a blueprint for site development. This may include planning the site's navigation and determining whether database integration is needed.
Graphic design	The job of the graphic designer is to create a consistent and visually appealing look for a Web site. Graphic designers may also create Flash animations or design user interfaces for Web applications.
Information technology	This group includes people with the more technical jobs involved in Web development, such as programmers, database administrators and networking professionals. The technologies these people enable in the site make it the interactive medium that it is.
Marketing	Marketing professionals are responsible for developing the strategy for communicating a company's message, image and brand identity. As such, they are often involved in developing the site's look and feel, as well as its content.
Writing and editing	These professionals are responsible for writing, editing and proofreading the content on the Web site. Web site content requires concise information and adherence to a consistent style that appropriately reflects the company's desired message, tone and image.

For more information about Web project collaboration, see *Web Design Specialist*, Lesson 2: Web Development Teams.

1.1.4: Determining site audience

Among the most valuable information a Web developer can have is knowledge of his or her audience. Knowing the audience requires you to learn demographics about its members: age, education, income and location. In addition, you need to determine the technology these users can support with their current computer systems, including connection speed, browser versions and available plug-ins. When you know these factors, you can tailor your content to maximize user support. Knowing your audience is crucial to the success of a site.

However, it is impossible to completely know your audience on the Web because the Web makes your site available to anyone with a browser. This fact makes accessibility important. To make your Web site usable to your entire audience, you need to consider all users, including users with various system capabilities and users with disabilities.

For more information about determining the Web site audience, see *Web Design Specialist*, Lesson 4: Web Site Development Process, and Lesson 6: Web Site Usability and Accessibility.

1.1.5: Web site vision statement

Defining a Web site vision statement is a key activity that takes place during the conceptualization phase of a Web development project. The vision statement is the fundamental framework that defines the scope and intent of a Web site. This statement should be concise enough that everyone involved in the development process can focus on fulfilling the intended vision. The vision statement should include a value and a measurable goal.

The following example can be considered a good vision statement for an online business such as Amazon:

> *We will become the world's pre-eminent Internet book retailer, selling $1 million in books per day by the end of next year.*

Inherent in this statement is a value ("pre-eminent Internet book retailer") and a measurable goal ("$1 million per day by the end of next year").

A vision statement provides a solid goal toward which to work. Vision statements are just as useful for a small site as for a large e-commerce business because they help all developers focus on achieving goals of any size. Many companies post some version of their vision statements (also often called mission statements) on their Web sites. These statements may include other details to appeal to their customers, but generally define the scope and intent of the businesses they represent.

For more information and a hands-on lab about Web site vision statements, see *Web Design Specialist*, Lesson 4: Web Site Development Process.

1.1.6: Site strategy and tactics

After developing a vision statement that will guide your efforts to create a successful Web site, you need a site strategy to determine how you will achieve that vision. Your strategy must be focused on persuading users to spend time on your Web site and return to it in the future. The goal of electronic commerce — and any business — is to attract and keep customers. To accomplish this goal, you can use the unique features of the Internet (such as the ability to address individual needs) to your advantage.

A tactic is a method used to implement your strategy. For example, if you want to gather personal information from your users, you need a tactic. A widely used tactic is to offer something (e.g., a screen saver or a discount) in exchange for that information. Another common tactic rewards the first-time buyer with an additional discount or bonus. The net effect is that you now have the customer's information. The next time he or she visits your site, you can use this data to complete the customer's purchase or transaction much more quickly. You can also market to users based on their preferences from previous purchases.

For more information and a hands-on lab about site strategy and tactics, see *Web Design Specialist*, Lesson 4: Web Site Development Process.

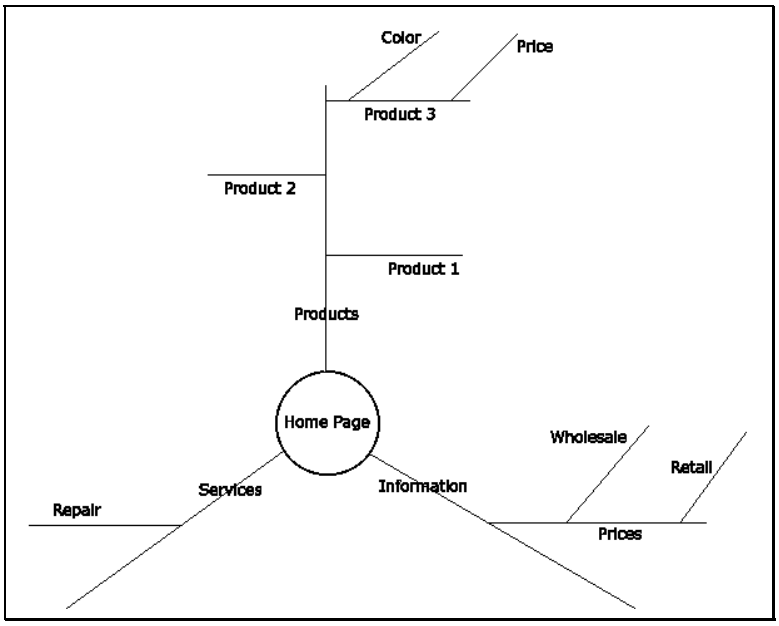
1.1.7: Mindmapping site structure

Mindmapping is a process that allows you to structure ideas on paper in the order your brain follows, rather than the linear process normally used when documenting ideas.

Instead of beginning in the upper-left corner of the paper and proceeding down line-by-line, place your subject in the middle of the page and circle it. From there, draw branches, which are ideas about your topic. If any topics are related in a more definitive way, create another branch off the current idea branch. Within minutes, you will see your mindmap develop into a dynamic sketch.

Mindmapping is a process of moving ideas from thought to document. Do not judge whether ideas are good or bad; just write them down and move on to the next thought. After you complete your mindmapping process, you can go back and refine or eliminate ideas.

Some techniques for mindmapping include using a large chalkboard or whiteboard. You can also use different colors to designate specific categories or items, or use thick markers and thin markers — whatever you can find to help stimulate the process. The following figure illustrates the way you can use mindmapping to develop a Web site. This example is simplistic. Your mindmap might have dozens more branches, which will help you develop a better Web site.



For more information and a hands-on lab about mindmapping Web site structure, see *Web Design Specialist*, Lesson 4: Web Site Development Process.

1.1.8: Design goals for customer and audience

The role of the Web designer is becoming more complex. In the past, Web designers could suffice with the skills to create a Web presence for the customer organization. Today, having a Web presence is not enough. The Internet is no longer a repository of electronic brochures, but rather a collection of increasingly sophisticated technologies that offer the ability to develop lucrative Internet-based businesses.

The concepts for these successful electronic businesses are varied. Generally, they can be placed into two categories: sites that deliver products intrinsically dependent on the Internet (such as interactive games or tools for searching the Internet), and sites that deliver existing products and services to a global market via the Internet. What the known successful Web sites have in common is that they focused on the goal of fulfilling unmet needs for Web site users.

It is important for Web designers to learn the customer's needs and the site's audience before forming ideas about the design and objectives for the site. These factors determine all the components needed for the site to be successful, from its look and feel to its functionality.

You can establish a set of desired user behaviors for a Web site. Each of these desired behaviors introduces possible scenarios to be considered. Scenarios greatly facilitate the process of developing a Web site. They first provide you with a clear image of the project scope, and then they serve as a tool during site development to keep the project on track, budget and schedule. The look, feel and

functionality of a Web site emerge as the various user scenarios are developed from the user's point of view. Knowing your audience is crucial to the success of your site.

For more information about design goals for customer and audience, see *Web Design Specialist*, Lesson 4: Web Site Development Process.

1.1.9: Site metaphor

A metaphor suggests a likeness or pre-existing identification with other things or experiences. The most common representations used by Web sites are brochures, prospectuses and catalogs — all metaphors of the print medium. Some sites use the familiar metaphor of television. However, the Internet offers different opportunities for user interaction. Even when it is appropriate to use print medium as the metaphor, it should be done intentionally, using the best principles from that medium.

Consider the concept of metaphor you may recall from language classes. A metaphor suggests analogy between two ideas or objects without using the words "like" or "as." In site design, a site can be presented as a particular object or experience to emphasize or stylize the site's purpose. One design metaphor that has become very widespread in Web design is the tabbed folder metaphor. Resembling a stack of tabbed folders in which a tab can be clicked to "open" each folder, the tabbed folder format has long been a standard in software design — most notably in the design of software preferences dialog boxes. Tabs are used for navigation on many of the biggest sites, including Amazon.com, Expedia and PayPal.

Creating a site metaphor can be beneficial because it helps you focus on a strong, controlling visual theme for the site and helps create a consistent look for the site.

Consider the following guidelines when developing a metaphor for your site.

- Consider whether a metaphor is needed to express the desired idea.
- Select a metaphor that is familiar to the chosen audience.
- Use the familiar to explain the unfamiliar.
- Keep metaphors light and effective.
- Be sure that the comparison shares characteristics with your theme.
- Use the metaphor consistently in the design.
- Do not overuse the metaphor.
- Do not use a metaphor that may have any negative connotation.
- Choose a metaphor that is easy to remember.
- Do not mix metaphors.

Remember that it is not always necessary to choose a site metaphor, and that strictly adhering to a metaphor might hinder creativity, resulting in a boring site.

For more information about Web site metaphors, see *Web Design Specialist*, Lesson 4: Web Site Development Process.

1.1.10: Site design and architecture specifications

After you have determined the audience for the site, and you have created a strategy for your Web business and tactics to achieve that strategy, it is time to create specifications. Specifications define the features, content, functionality and structure (or architecture) that the site requires in order to meet the customer's needs and achieve the site's goals. Specifications can be divided into the four types described in the following table.

Site Specification Type	Description
Functionality	Functionality specifications indicate the functions or activities that the site should be able to perform. For example, the functionality specifications for an e-commerce site will usually include search features, a shopping cart, credit card processing and so forth.
Content	Content specifications are usually created before the architecture specifications, because you generally need to know what information the site will contain before you can structure and organize that information. Content specifications indicate the general types and topics of content that the site will include, such as text describing the company, product descriptions and images, shipping information, and frequently asked questions.
Architecture	Architecture specifications indicate the way that the site will be structured and ways that users will navigate it. For example, an online shoe store might organize the shoes into a group for men and a group for women, with additional subgroups of dress shoes, casual shoes and athletic shoes. The shoes might also be separately grouped by price range, brand, color and so forth. The site architecture specifications help you to determine ways to organize and relate these types of information for navigation and searching.
Design	Design specifications are used to plan the general look and feel of the site, including the fonts, colors and images that will be used. For example, many companies have standards for the sizes in which their logos can appear for designated uses, as well as the colors and fonts that should be used for specified purposes and areas on the site. These design requirements are discovered and recorded during the requirements gathering phase of the project. When the design specifications are created, the Web designer determines how to satisfy these requirements. Generally, site design templates are most useful for this purpose.

Web site specifications do not need to be complicated. Often, they can be created on a white board or developed through a series of e-mail messages. The important point is that the final document — whatever its form — should answer this question: What are we going to create?

The customer and any other stakeholders should review and approve all specifications before the project moves forward to the design phase.

For more information and a hands-on lab about Web site design and architecture specifications, see *Web Design Specialist*, Lesson 4: Web Site Development Process.

1.1.11: Site project implementation factors

Site project implementation factors (such as desired functionality, required technologies, stakeholder input, time frame and scope) must be determined before you begin development on a Web project.

Once the need for a Web site is determined, the next job is to determine the scope of the project: the goals of the Web site and the tasks that need to be performed to achieve those goals. A needs analysis identifies the problems, challenges or needs that the project must resolve or address. The needs analysis determines a customer's requirements, desires, time frame and budget.

The project goals, or objectives, specify the steps you and your Web development team will perform to satisfy the customer's needs. Ideally, project goals should be measurable as well as specific. By determining the customer's needs during the initiating phase, you can create a Web site that meets those needs, and you can determine the tasks, technologies, time and cost required to complete the Web site.

A stakeholder is a person or group with an interest in a project and the power to exert influence (either positive or negative) over the project and affect results. All stakeholders must understand and agree to the project objectives. These objectives are the yardstick by which the success or failure of your Web development project will be measured. In addition to agreeing on the goals of the project, stakeholders must also discuss and agree on assumptions and constraints.

The end product or ultimate goal of a project (in this case a Web site), along with the tasks required to achieve that goal, constitute the project scope. A common problem associated with project management is

the tendency for the project's scope to increase over time. Any changes in the schedule, cost or performance required to complete the project can affect its scope. Issues often arise during the project that were not initially considered. Changes in project scope tend to occur in small increments, and therefore might seem negligible. Small increases in scope will add up, and may require additional resources or time to execute. These gradual increases are called scope creep. If you do not adequately manage scope creep, the success of your Web development project may be compromised.

For more information about Web site project implementation factors, see *Web Design Specialist*, Lesson 3: Web Project Management Fundamentals.

1.1.12: Web project development and rollout plan

The Web project planning phase addresses project details — the specific ways that you are going to achieve the goals and objectives of your Web development project. Project planning involves identifying all the tasks required to achieve the project goals, the people who will perform those tasks, and the resources that will be required. Planning also involves estimating the time each task will take to complete and determining project tasks that are dependent upon one another.

During the planning phase, you must develop a project schedule, or development schedule. A project schedule lists the planned dates for performing tasks and meeting goals identified in the project plan. To develop a project schedule, you should meet with the entire Web development project team to determine the time and money required to complete each task. You should then outline the required tasks and assign resources (money, time and people) to each task.

The development schedule should also include a site rollout plan. A site rollout plan details the implementation steps for launching the site after it is built. In the site rollout plan, it is important to include the time, money and skills that will be required to test and deploy the finished Web site.

The final goal of the planning phase is to have a development plan that outlines tasks, responsibilities, budgets and deadlines. However, remember that planning is a constantly evolving process that lasts throughout most of the project life cycle.

For more information about Web project development and rollout plans, see *Web Design Specialist*, Lesson 3: Web Project Management Fundamentals.

1.1.13: Site characteristics and resources

Consider the following key Web site characteristics that you may include in a site.

Web Site Characteristic	Description
Navigation	Navigation controls the user's movement through the site. Clear and easy navigation is critical to a site's success, and therefore must be well planned and carefully implemented to enable visitors to effectively use your site.
Interactivity	Interactivity is a feature of Web sites that distinguishes it from other media types. Some Web sites offer little interactivity beyond providing several pages for users to browse among and read. Other sites offer a high level of interactivity, with forms for users to complete and submit, multimedia to watch or listen to, games to play, catalogs to search, and products to buy. Depending on the type of interactivity the project requires, the project team may need programmers or graphic designers with specialized skills.
Database integration	Database integration allows Web pages to use dynamic data, often in conjunction with interactivity. Databases provide the ability to store and sort vital information, such as customer data and product information. The site designer's job role does not necessarily include the skill of database integration, but it is an important and common site characteristic that you may need to incorporate into your design and page layout features.

The personnel and resources required for a project depends on the characteristics of the Web site that will be built. If the project requires database integration, for example, then a database administrator must be part of the team. If the project has complex information organization needs, then an information architect should be involved. Depending on the type of interactivity the project requires, the project team may need programmers or graphic designers with specialized skills.

For more information about Web site characteristics and resources, see *Web Design Specialist*, Lesson 2: Web Development Teams.

1.1.14: Web site branding

Branding is the practice by which a company tries to create an image of itself in the minds of the public with the hopes that consumers will purchase its products or services. The most important element of a company's brand is its logo. Think of several well-known companies, such as Pepsi, Coca-Cola, Apple Computers or Nike. Everyone knows what these companies' logos look like, and we instantly associate certain shapes or colors with these companies.

It is important for an established company to ensure that its Web site enhances its brand. A poorly designed site that does not reflect the company's other marketing or branding efforts can negatively impact the company's image. By contrast, a high-quality Web site that is designed to enhance the company's image and support its other branding efforts can be an important component of any company's marketing strategy.

New companies that do not already have well-known or established brands can use the Web to build recognizable brands. To be successful, a branding effort should be unique, attractive and memorable. If a company's Web site is sloppy and does not incorporate a visual theme or coherent look, users are unlikely to remember the site or the company.

Some of the ways that a company can apply and strengthen its brand on its Web sites include:

- Consistently using colors that it wants associated with its brand.
- Placing the company logo on every page, generally in the upper-left corner.
- Creating a company style guide that standardizes the ways to which its products or services are referred.
- Using page templates to ensure a consistent page appearance throughout the site.
- Requiring the use of the trademark symbol whenever the company's name, logo or products are mentioned on the site.

For more information about Web site branding, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements.

1.1.15: Audience usability tests

The only way to achieve maximum site usability is to conduct a usability test. The developer cannot validly indicate that his or her site design will be functional for the audience because the developer has an intimate knowledge and understanding of the intentions of the site from his or her production efforts. This closeness gives the developer a biased perspective of the site's actual usefulness. A site's effectiveness should be tested with users who have had little or no exposure to the site. Conducting a usability test gives the developer an objective view of the site. The following table describes common steps and considerations in audience usability testing.

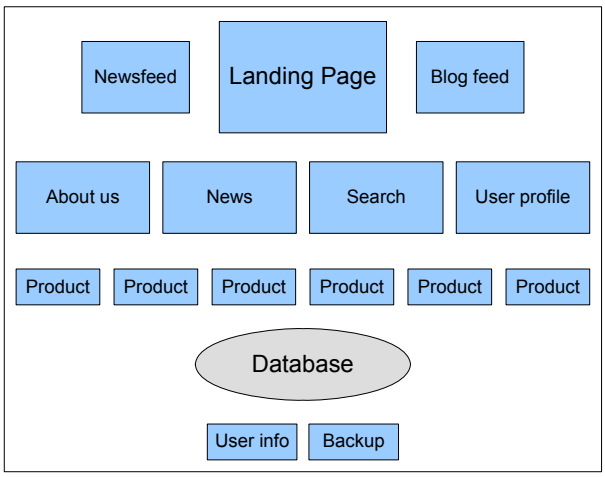
Usability Testing Step	Description
Before the test	The first step in testing usability is to develop the site to a point very close to the finished product. If the project is not yet at this stage, a usability test cannot provide an accurate evaluation.
Identifying testers	The test pool can range from as few as five or six users, to as many as you can accommodate. Testing the site with other design team members is inconclusive. Testers should include actual target audience members. A good cross-section should include some professionals familiar with the Web and your site's subject, some users who are moderately familiar with the subject, and some users who know little or nothing about it. Whoever the testers are, be sure you clearly understand their backgrounds so that their evaluations can be put into perspective. An exception to the diverse-background rule exists if the site is for internal corporate use, such as an intranet. In this case, the project team is very likely a part of the audience. However, you should still include others members of the organization, particularly those who are removed from the development process. This more objective group is the most accurate representation of successful site usability.
Usability tasks	During the test, participants should be asked to perform actual tasks. Provide a list of tasks and operations to each participant, with no indication of how to perform them. The site itself should tell users all they need to know to navigate and complete tasks. Participants should also be asked to note elements they like as well as those they do not. This feedback will make the site more effective. As a developer, you may find it difficult to watch test participants stumble through tasks or become frustrated, but you must not interfere in the process. In fact, if you feel observation is necessary, conduct it through video or a one-way mirror. This method precludes any contamination of the testing.
Results	After testing, interview the participants upon exit. You can include questions such as the following: - What was your first impression when you saw the site? - What type of company image did the site portray? - Do you understand the site structure? - Can you recall the site's major elements? Written data should be compiled and analyzed from a distance. How many users had the same experience? Were any problems consistently noted? These trends are the real indicators of usability. Be sure to take a closer look at these problems individually, to see how you as the developer can heighten user awareness and usability.
Applying the results	Some insufficiencies will probably surface during the testing. You must consider feedback constructively and address the issues. The sting of critique from a few participants will be short-lived compared with a Web site catastrophe that could occur if you go online with a faulty site and receive widespread criticism for preventable or foolish mistakes.

For more information about audience usability tests, see *Web Design Specialist*, Lesson 6: Web Site Usability and Accessibility.

1.1.16: Flowcharts and Web wireframes

A Web site wireframe is a sketch of the skeletal view of a site's architecture. Creating a wireframe is the next step you would usually take after creating a mindmap, and the Web site wireframe is a finalized representation of the site.

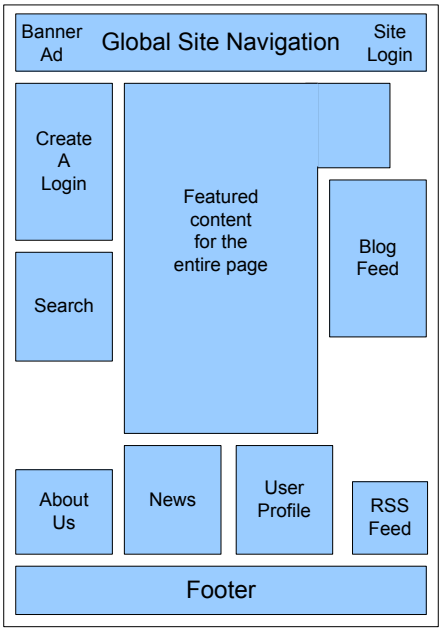
The following figure provides an example of a Web site wireframe. Notice how the hierarchical format helps describe the relationships between the pages.



Elements of a Web site wireframe include:

- A description of each page on the site.
- A listing of all elements necessary for the site to fulfill its business purpose. This includes discussion of database connections, scripts, Web applications and other programming required to make the site fully functional.

A Web page wireframe allows you to focus on the flow of content on an individual Web page, as opposed to an entire site. See the following figure.



Creating wireframes for your site and pages can help you think through each element to ensure you are making the proper high-level decisions.

For more information about using flowcharts and Web wireframes, see *Web Design Specialist*, Lesson 4: Web Site Development Process.

Practice exam questions

Following are some practice exam questions intended for student review.

1. In a collaborative Web development environment, which team member is responsible for assessing and creatively managing the technological and human resources available?

a. Project manager
b. Graphic designer
c. Marketing manager
d. Information technology manager

Correct response: a. Project manager

Explanation: The project manager is in charge of determining and documenting the requirements of a project, creating specifications for the project, assembling a team, and ensuring that the project stays on schedule and on budget. Marketing professionals are responsible for developing the strategy for communicating a company's message, image and brand identity. Information technology professionals work with technologies such as databases, programming languages, servers and networks.

2. Which fundamental framework in the Web development process defines the scope and intent of a Web site by stating a value and a measurable goal?

a. Site tactic
b. Vision statement
c. Design specification
d. Architecture specification

Correct response: b. Vision statement

Explanation: The vision statement is the fundamental framework that defines the scope and intent of a Web site. This statement should be concise enough that everyone involved in the development process can focus on fulfilling the intended vision. The vision statement should include a value and a measurable goal.

1.2: Customer expectations

This subdomain includes skills and knowledge required to meet customer expectations with Web site project and design.

1.2.1: Needs and usability vs. design and aesthetics

It is important for Web designers to learn the customer's needs and the site's audience before forming ideas about the design and objectives for the site. These factors determine all the components needed for the site to be successful, from its look and feel to its functionality.

The needs analysis determines a customer's requirements, desires, time frame and budget. The project goals, or objectives, specify the steps you and your Web development team will perform to satisfy the customer's needs. All stakeholders must understand and agree to the project objectives. These objectives are the yardstick by which the success or failure of your Web development project will be measured.

Knowing your user audience is also crucial to the success of your site. You can establish a set of desired user behaviors for a Web site. Each of these desired behaviors introduces possible scenarios to be considered. Scenarios greatly facilitate the process of developing a Web site. They first provide you with a clear image of the project scope, and then they serve as a tool during site development to keep the project on track, budget and schedule. The look, feel and functionality of a Web site emerge as the various user scenarios are developed from the user's point of view.

Ultimately, if you do not satisfy your Web users' needs or desires, they will find other sites that will. The Web designer who thinks only from his or her own perspective, and not from the users' perspective, will certainly find dissatisfied Web visitors, clients and customers.

For more information about needs and usability vs. design and aesthetics, see *Web Design Specialist*, Lesson 1: Overview of Web Design Concepts, and Lesson 2: Web Development Teams.

1.2.2: Customer expectations and feedback

A customer can be another company, an individual, another department within the same company, or even the Web team itself (in the case of internal Web site projects such as intranets or repositories).

Once the need for a Web site is determined, the next job is to document the scope of the project. In other words, you need to determine the goals of the Web site and the tasks that need to be performed to achieve those goals. In order to conduct a thorough needs analysis for a Web site development project, you must understand the customer's desires as well as the customer's business. After an initial meeting, the project manager should work with the customer to create the needs analysis document.

Once a Web project manager understands the customer's needs, he or she can create a plan of action, which will be stated as a list of goals and objectives. All stakeholders must understand and agree to the project objectives.

Finally, the project manager and all stakeholders should document the ways that project success will be measured. At the end of the initiating phase, the project manager should complete a Statement Of Work (SOW) before proceeding to the next phase. The SOW outlines the requirements for each project task to ensure that the objectives are met. The SOW should contain clearly defined goals and an agreed-upon plan to achieve them. The SOW can be created between individuals, between departments, between individuals and departments, and so forth. If the SOW is completed between your organization and an outside organization or individual, the payment structure can be included in the SOW.

For more information about customer expectations and feedback, see *Web Design Specialist*, Lesson 3: Web Project Management Fundamentals.

1.2.3: Communicating progress to stakeholders

As a Web development project progresses, the project manager is responsible for keeping project participants informed in order to keep the project running smoothly. Regular communication with the customer and all stakeholders helps to keep the project on track and avoid unnecessary work. Understanding the status of a project can also help participants focus on areas that need immediate attention. Documentation of progress is an important part of communication both during and after a project.

For more information about communicating progress to stakeholders, see *Web Design Specialist*, Lesson 3: Web Project Management Fundamentals.

1.2.4: Changes in project scope

The end product or ultimate goal of a project (in this case a Web site), along with the tasks required to achieve that goal, constitute the project scope. A common problem associated with project management is the tendency for the project's scope to increase over time. Any changes in the schedule, cost or performance required to complete the project can affect its scope. Issues often arise during the project that were not initially considered. You may be able to contain the scope of the project, but only by introducing subprojects, which also must be managed.

Changes in project scope tend to occur in small increments, and therefore might seem negligible. Small increases in scope will add up. These gradual increases are called scope creep.

Increases in scope should be documented in the same way that the initial goals and objectives were documented. Although it is tempting to accept every additional request made during the project, the project manager needs to know when to accommodate requests, and must advise the customer that increases in scope will require increases in resources and time, or that additional features could be implemented in a separate project as part of a "phase 2." If you do not adequately manage scope creep, the success of your Web development project may be compromised.

For more information about changes in project scope, see *Web Design Specialist*, Lesson 3: Web Project Management Fundamentals.

1.2.5: Changes in development plan

Certain features are critical in every Web site. In an e-commerce site, the basic e-commerce functionality must be functional before you can launch the site. In order to meet a rollout deadline or stay within budget, however, it may be necessary to delay or scale down some secondary features, such as customer review functionality or recommendation functionality.

After the planning stage, you should be able to identify the tasks that are most important and the tasks that can be delayed without affecting the completion of other tasks or the overall project. Being aware of critical tasks will help you make adjustments to ensure that the project is completed on time.

For more information about changes in the development plan, see *Web Design Specialist*, Lesson 3: Web Project Management Fundamentals.

1.2.6: Project tracking reports

During the executing and controlling phases of a Web development project, you should document project tasks to provide a paper trail, or record, of the team members who worked on tasks, and the dates they started and completed them. The project tracking report is the document that records this information and provides your paper trail. The paper trail does not need to be documented on paper, however; many complex project management software applications exist that can automate tracking and manage every part of the project, including documents, contacts, tasks, schedules, issues logs and more.

By documenting who did what and when on the Web site as the project advances, you can track its progress by determining whether tasks are starting and finishing on time, and whether tasks are being

completed within the budget. By tracking these factors while the project is still in progress, you can solve problems as they occur and make necessary adjustments. Documenting a project also promotes team member accountability and enables stakeholders to monitor various stages of the project. Tracking the progress of a Web site development project from beginning to end is essential for keeping stakeholders informed and for keeping the project on track.

You should also keep an issues log in which you document problems that need to be escalated to managers or executives outside the Web development team for resolution. Issues often arise during the course of a project that require authoritative decisions in order for the team to complete tasks and keep the project on track. Examples of issues that may arise during the Web development process include design or programming bugs, requested changes to the design or content of the site, and server or browser incompatibilities that need to be addressed. You can use the issues log as backup documentation to support any time, resource or cost changes that may accrue due to circumstances beyond the project team's control.

For more information and a hands-on lab about project tracking reports, see *Web Design Specialist*, Lesson 3: Web Project Management Fundamentals.

1.2.7: Project evaluation

The final phase of a Web development project is the closing phase. During the closing phase, your team should evaluate the project schedule, budget, scope, resources and assignments to determine the aspects of the project that worked well and the changes that should be implemented in the future. A project is deemed a success when it is completed within the budget and time frame specified, and the finished product meets quality standards.

At the end of the closing phase, you should receive a formal acceptance of the Web site from the customer, a documented history of the development project, and recommendations for revising the project plan for future Web development projects.

You must also establish how or at what point the Web site will be turned over to the customer. This step includes defining the responsibilities and duties of the project team and the customer. In some cases, the Web development team may stay involved with support and maintenance of the site; in other cases, the customer takes on these responsibilities after receiving the finished Web site. You should document the technologies used on the site (such as programming languages and server components) and project style guidelines (such as fonts and colors used in the design) that will be needed by the people who will maintain the site, whether it is your team members or someone else's. The transition may be difficult if you assume that the customer will be ready and able at any time to accept the Web project and associated maintenance.

For more information about project evaluation, see *Web Design Specialist*, Lesson 3: Web Project Management Fundamentals.

1.2.8: Web page and site templates

Design professionals prefer to use templates when creating their Web pages. Using a Web page template will help you apply your design consistently and quickly to each page in your site, ensuring that the site has a unified look and feel that does not vary from page to page.

A Web page template is an X/HTML page structure (and sometimes an associated style sheet) that acts as the foundation for each page you create. A template specifies default settings or attributes. For example, a template can specify the layout structure, images, typeface, font colors and so forth used throughout a page or site. By using templates, you can ensure that all your pages consistently adhere to the site's design specifications without having to manually format each page. Your page template must always fulfill design specifications, so it is critical to ensure that all stakeholders review and approve the template design (i.e., mock-up).

Because many of the pages in a site will use the same elements, it makes sense to create a template to reduce development time. After a template is created and saved, it can be used anytime you want to create a new page with the specified properties.

Most major Web-development applications, such as Expression Web and Dreamweaver, allow you to create your own Web page design templates. You can also find examples of ready-made page templates at various Web sites, some of which you can use free of charge. Ready-made templates can provide you with some good ideas for your site's look and feel. However, these pre-designed templates are unlikely to fulfill all your design specifications, so be sure to modify any ready-made templates you use according to your site's needs.

For more information and hands-on labs about Web page and site templates, see *Web Design Specialist*, Lesson 4: Web Site Development Process, Lesson 18: Site Development with Expression Web 3 — Basic Features, and Lesson 21: Site Development with Dreamweaver CS5 — Basic Features. For examples of ready-made page templates, visit Free Website Templates from Virtual Promote (*http://freesitetemplates.com*), TemplateMonster.com (*www.templatemonster.com*) and Hoover Web Templates (*www.hooverwebdesign.com/templates*).

1.2.9: Designing for PDA-based vs. traditional browsers

A growing number of people surf the Web on personal digital assistants (PDAs), and many of these people have disposable income they can spend on your site. Therefore, it is often well worth the time to create separate pages (or at least separate style sheets) for PDA-based browsers.

When designing for PDA-based browsers, keep the following points in mind:

- PDA browsers are not as powerful as traditional browsers.
- PDA browsers do not handle client-side scripting very well, so it is best to avoid interactive elements, using only XHTML and style sheets.
- PDA screens are small. Avoid using static page widths.
- Users are paying for their time online, so use few words and small images to ensure that your pages load quickly.
- Validate your code (*http://validator.w3.org/mobile/)*.
- Test your pages on various mobile devices.

For more information about designing for PDA-based vs. traditional browsers, see *Web Design Specialist*, Lesson 4: Web Site Development Process, and Lesson 7: Browsers.

Practice exam questions

Following are some practice exam questions intended for student review.

1. In a Web development project, what is scope creep?

a. The goals and tasks of a project, and the work required to complete them
b. A document that lists the planned dates for performing tasks and meeting goals
c. Incremental increases in a project's requirements after the project has started, without an increase in resources
d. The process of determining a customer's needs by acquiring information then creating a plan of action to address the identified needs

Correct response: c. Incremental increases in a project's requirements after the project has started, without an increase in resources

Explanation: Changes in project scope tend to occur in small increments, and therefore might seem negligible. Small increases in scope will add up. These gradual increases are called scope creep. The project scope is the goals and tasks of a project, and the work required to complete them. The project schedule is a document that lists the planned dates for performing tasks and meeting goals defined in a project plan. A needs analysis is the process of determining a customer's needs by acquiring information then creating a plan of action to address the identified needs.

2. Why is it important to use a Web page template when designing your site?

a. Because it documents and provides reference to the design conventions you choose to use
b. Because it defines the scope and intent of a Web site, and helps all developers focus on achieving the goals of the site
c. Because it helps ensure that you fulfill your site's design specifications by applying a consistent look and feel to your site
d. Because it allows you to structure ideas on paper in the order your brain follows, rather than the linear process normally used when documenting ideas

Correct response: c. Because it helps ensure that you fulfill your site's design specifications by applying a consistent look and feel to your site

Explanation: Using a Web page template will help you apply your design consistently and quickly to each page in your site, ensuring that the site has a unified look and feel that does not vary from page to page. Your page template must always fulfill design specifications, so it is critical to ensure that all stakeholders reviewed and approved the template design. A style guide can be used to document or provide reference to the design conventions you choose to use, but does not apply or ensure these specifications. A vision statement defines the scope and intent of a Web site, and helps all developers focus on achieving the goals of the site. A mindmap allows you to structure ideas on paper in the order your brain follows, rather than the linear process normally used when documenting ideas.

1.3: Ethical and legal issues

This subdomain includes skills and knowledge required to identify ethical and legal issues relevant to Web development and design.

1.3.1: Ethics, legal vs. ethical issues

Currently, many laws related to Web businesses and practices are not completely clear; others are evolving each day as court cases and legislation are addressed and settled. Laws specify illegal behavior that can be punishable by fine or imprisonment. However, the law is only a baseline for the conduct you should observe — some practices may currently be legal, but the lack of a forbidding law does not necessarily make them acceptable. Some practices that are still legal are considered unethical by professional standards.

Ethics are a set of accepted standards of conduct governing members of a profession, such as Web developers. Ethics establish basic values for responsible actions and practices within a professional community. Ethics deal with good versus bad: values, moral duty and obligation. Although there may be no punishment for violating ethical standards, some practices (such as spamming and sending viruses) are being written or will be written into law.

The generally agreed-upon ethical standards for Web professionals have developed from years of experience, and many have been inherited from other professions. Examples of ethical behavior for Web professionals include the following:

- Do not send unsolicited bulk e-mail, or spam.
- Do not buy domain names that you do not intend to use (a practice also known as domain squatting or cyber-squatting).
- Do not knowingly spread malicious program code such as viruses or worms.
- Do not pass along chain e-mail messages, especially those that imply threats.
- Be honest with your customers, and do not overcharge for technical services that they may not understand.

For more information about legal vs. ethical issues, see *Web Design Specialist*, Lesson 11: Ethical and Legal Issues in Web Development.

1.3.2: Proper content usage

Intellectual property laws govern ideas and products of the mind. The content of a Web site — including text, graphics, multimedia, music and code — is considered intellectual property. Understanding the major laws governing intellectual property (described in the following table) will help you protect the intellectual property on your Web site, as well as prevent you from infringing upon the property of others. Infringement upon legally protected intellectual property is punishable by law.

Intellectual Property Law or Issue	Description
Copyright	In the United States, the 1976 Copyright Act prohibits unauthorized duplication of "original works of authorship that are fixed in a tangible form of expression" — literary, artistic, photographic, musical and audiovisual works. According to copyright law, the two components that distinguish a work of authorship are originality and expression, so copyright applies only to the work itself, not the objects or ideas described in the work. Note that the content owner is not required to complete a formal copyright registration process to be protected by copyright law. Copyright is secured automatically from the time that the work is created. A work of authorship immediately becomes the property of the author who created the work, and only the author of that work can share or transfer the ownership rights. The result is that you cannot copy text, images, music, program code, etc., from any source, on or off the Web, and use it in your Web site or any other publication.
Copyright infringement and fair use	Copyright infringement is the use of or profit from someone else's work without their permission. However, copyright does not restrict you from quoting or making reference to someone's work. The Fair Use doctrine included in the 1976 Copyright Act allows limited use of copyrighted works in socially important ways, such as research, teaching, journalism and criticism. Essentially, you can quote from or refer to other authors' works, but you cannot copy them wholesale or present them as your own. If your Web site allows users to post information, it is important to include a Terms Of Use agreement that users are required to submit before posting any content. This agreement should include statements specifying that users are not allowed to post copyrighted material, that you are not responsible to validate their content as original, and that users are responsible for any actions resulting from content they post. If a user violates a copyright on your Web site, then the fact that he or she agreed to your Terms Of Use may protect you.
Trademark	A trademark is any word, name, phrase, sound or image that a company uses to distinguish itself from the competition — most frequently, a company's logo and slogan or catchphrase. You cannot borrow another company's logo or catchphrase and use it for your company. A parody of a company's trademarked name, logo or catchphrase may be permitted on a humorous Web site under the Fair Use doctrine. Trademark law is also the basis for domain name disputes, which currently are primarily resolved by the World Intellectual Property Organization (WIPO) Domain Name Dispute Resolution Service.
Trade secret	Although laws differ by state in the United States, a trade secret is generally a formula, pattern, idea, process or compilation of information that provides the owner with an advantage in the marketplace and that is treated by the owner in a way that can be expected to keep the public from learning about it. For example, a secret recipe, a customer information list or a series of books in progress could be trade secrets. If the owner of trade-secret information publishes the information on the Web, then it is no longer a trade secret. However, publication of trade-secret information by anyone who is not authorized to do so is illegal.
Non-disclosure agreement (NDA)	A non-disclosure agreement (NDA) is designed to protect trade secrets (as well as copyrighted material) from improper use by people who are allowed to access them, such as company employees, contractors and business partners. An NDA is commonly signed at the beginning and end of employment and contract relationships.
Licensing content for your site	If you want to use another author's copyrighted content (text, images, music, code, etc.) for your Web site or other purpose, you may be able to license the content from its owner for your use. Licensing means that you establish an agreement with the content owner, which might be a writer, artist, programmer, magazine, corporation or Web site. Licenses are typically limited in some way. When creating a license agreement, the content owner establishes the terms of the agreement, dictating how and where the content can and cannot be used, and what the owner will receive in exchange for use of the content (such as fees, credit attribution, etc.). Some content owners will allow free use of their content for personal, non-commercial purposes. Some Web sites will allow other sites to provide links to their content. Again, any use still has terms and limitations. Some sites specify these policies to the public on their legal, permissions or copyright pages.

Table cont'd

Intellectual Property Law or Issue	Description
Public domain	Content that is not copyrighted, or for which the copyright has expired, is said to be in the public domain. Any work that is in the public domain may be used freely by anyone for any purpose. Laws differ by country. In the United States, the following types of works are considered to be in the public domain: - Works that were first created and published prior to January 1, 1923. - Works whose term of copyright has expired. (Generally, works created on or after January 1, 1978, are protected by U.S. copyright for the life of the last surviving author plus 70 years.) - Works for which no copyright extension has been filed.
Inaccurate content / libel	Depending on the nature and purpose of your site, content that is inaccurate, misleading or false can harm your business. You can have an editor perform fact checking on your content as a precaution to avoid unintended false statements. Editors can review content to ensure that it is consistent throughout the site, and that statements made on one page do not contradict information on another page. Publishing false statements about a person can be considered libel (which is defined as unjust inflammatory statements). If you are posting any potentially libelous or infringing information on your site, you should protect yourself by first consulting a lawyer.

Because Web site content is protected by laws such as copyright and trademark, you must ensure that any content you use on your Web site (text, images, multimedia, etc.) is either your own original content — and thus protected from unauthorized use by others — or borrowed content that you have proper permission to use.

Although users can view and retrieve source code for just about anything on the Web, inappropriate use of other people's work is illegal. As the Web develops and programs are invented to track site visitors, anonymity will decrease. To avoid legal penalties, all Web designers should opt for a conservative interpretation of intellectual property issues, which is as follows: If you did not create it, you probably should not use it on your site. However, remember that many services — some free — offer Web content that you can use on a Web site with some restrictions, often a required attribution or limitation to only personal (non-commercial) sites.

For more information about proper content usage, see *Web Design Specialist*, Lesson 2: Web Development Teams, Lesson 10: Multimedia and the Web, and Lesson 11: Ethical and Legal Issues in Web Development. For more information about U.S. copyright law and associated topics such as fair use, licensing and trademarks, visit the home page for the United States Copyright Office at *www.copyright.gov*.

1.3.3: Site strategies and technologies to avoid

Some technologies and some design techniques limit your site's usefulness to certain audiences. Others commonly annoy users. You should avoid such technologies and techniques in your site designs, unless you have compelling reasons to use them and are not concerned about alienating some sectors of the Web audience. Following are some examples of technologies and site strategies that you should avoid when designing a Web site.

Technology / Site Strategy to Avoid	Description	Examples
Single-browser technologies	Single-browser technologies are generally those created by a vendor and thus functional only when that vendor's browser is used to view the page in which they reside. Such technologies limit your Web site's audience to users who are using the one specific browser. So unless you know that all of your site's users will be using a specific browser (such as with a corporate intranet at a company in which all users have a standard browser), you should avoid these types of browser-specific functionality in your site designs.	**The <blink> tag** — Created by Netscape in the early days of the Web, it was never supported by browsers other than Netscape. Its function was to cause text to appear and disappear (or blink) in the browser window. This effect proved to be of limited usefulness, but because it provided some action it was often overused and thus soon considered annoying.
		The <layer> tag — In its version-4 browser, Netscape introduced a new technique, called layering, for more precisely positioning elements on a Web page. This technique required a new tag, <layer>. Since then, the W3C has promoted universal standards for all browsers that use Cascading Style Sheets (CSS) for element positioning on Web pages. With its version-6 browser, Netscape stopped supporting the <layer> tag.
		ActiveX controls — The Internet Explorer browser supports the use of Microsoft's ActiveX components, or controls, which can be used to enable interactive content on the Web. However, non-Microsoft browsers and non-Windows operating systems usually do not support it. In addition, ActiveX components may pose some security risk, although Microsoft has been making progress in addressing this problem.
Reliance on browser navigation	Although many users prefer to use the browser's Back button to return to the previously viewed page on a site, they should not have to rely on this browser button. Browser elements are designed to assist the user, but they should be a subordinate form of navigation.	In general, if the user must use the browser's Back button to navigate your site, then the site's navigation elements are insufficient.
Pop-up and pop-under windows	Every Web browser has the ability to display multiple windows and to create new windows, and this functionality can be controlled programmatically. These new browser windows are called pop-ups or pop-unders (depending on whether they appear above or below the browser window that launched them). This ability is necessary (for example, so that you can access more than one Web site at a time). However, the ability to launch additional windows is often abused by programmers to force users to view things (usually advertisements) that they did not request.	Pop-ups are fine if the user requests the content and if they complement the content of the main browser window. For example, a link to a movie or audio clip may be labeled, "Click here to launch this clip in a separate window." But pop-ups that appear unexpectedly and present advertising or other unwanted content to the user tend to be annoying. Another reason to avoid pop-ups is that many browsers now feature pop-up blockers, which prevent pop-ups from appearing as the user browses sites. As a result, pop-ups and pop-unders are much less effective as advertising today than they were when they were new, and using them for legitimate purposes may exclude many users from some valid functionality of your site.
Spam	Spam is unsolicited, bulk commercial e-mail that the recipients did not ask to receive. Spam is the equivalent of junk mail, and the vast majority of Web users consider it highly annoying. Sending unsolicited spam is a site strategy to avoid because it reflects poorly upon your organization and because in some cases it may be illegal.	Many businesses use bulk e-mailing lists legitimately for advertisement and promotion by sending e-mail messages only to recipients who have requested or approved the messages. This type of advertising strategy is not considered spam because it is based on user request, and thus can provide a very useful tool for your business.

Advertising on the Web is fine, but any content that is unsolicited and intrusive (such as pop-ups and spam) is very likely to make users unhappy with your site and perhaps your business. Such content may even result in legal consequences or blacklisting (i.e., a published list of IP addresses known to be sources of spam).

For more information about site strategies and technologies to avoid, see *Web Design Specialist*, Lesson 7: Browsers.

1.3.4: Strategies for end-user privacy and trust

Your responsibility as an ethical Web professional is to inform your site users as to how information you collect about them — including name and address, past purchases, reviews of products, and frequently visited areas of your site — will be used by your (or the site owner's) company, and whether this information will be sold to other companies. For this reason, it is standard to post a privacy disclaimer statement on every Web site, and it is essential on sites that collect information beyond normal log files. Privacy policies detail the ways that information is gathered and whether it is transferred to third parties. If information is transferred to third parties, the privacy statement should also describe how the information is transferred.

If you want to use bulk e-mailing to promote your Web site, you should avoid being labeled a spammer at all costs. To create a legitimate e-mail list, you must give users the option to subscribe to your e-mail list, which they can do by submitting a form on your Web site or by checking a box when they complete a transaction. This type of e-mail list creates an opt-in e-mailing service, which consists exclusively of users who specifically requested or authorized their addition to the mailing list.

Opt-out e-mail messages are bulk e-mail that offer recipients the option of unsubscribing from the list, generally by clicking a link in the message or visiting a Web page. Recipients of opt-out e-mail are expected to unsubscribe themselves from the list if they do not want to continue receiving the e-mail; otherwise, they are assumed to have authorized the messages. Opt-out e-mail is often used by legitimate businesses to subscribe users who have not requested the e-mail service but whose e-mail addresses were collected by the Web site for other purposes (such as completing a purchase or obtaining a site password). The opt-out method for creating a mailing list in this way is not recommended if you want your customers to feel confident about using your Web site. Some may find receiving your unsolicited e-mail messages so annoying that they will not return to your site. However, remember that you can provide good customer service by adding a convenient opt-out function to the e-mail messages you send to your opt-in e-mail list recipients.

For more information about strategies for end-user privacy and trust, see *Web Design Specialist*, Lesson 11: Ethical and Legal Issues in Web Development.

1.3.5: Privacy disclaimers

Web server software collects information about Web site visitors through the creation of log files. These automatically generated log files contain information such as the pages a user viewed, the type of browser used, and the duration of a user's visit. By using cookies, the Web server can also record when a user returns to the site. This information combined with information that an e-commerce program can collect — such as name and address, past purchases, reviews of products, and frequently visited areas of the site — is data that the marketing and sales departments of many companies would love to acquire.

Your responsibility as an ethical Web professional is to inform your site users as to how this information will be used by your (or the site owner's) company, and whether this information will be sold to other companies. For this reason, it is standard to post a privacy disclaimer statement on every Web site, and it is essential on sites that collect information beyond normal log files. Privacy policies detail the ways that information is gathered and whether it is transferred to third parties. If information is transferred to third parties, the privacy statement should also describe how the information is transferred.

Your privacy statement should be honest and open, rather than vague. All privacy disclaimer statements should include the sections described in the following table.

Privacy Disclaimer Section	Description
Information collection	A technical description of how user data is collected on the Web site. This description should specify the types of data that are required to perform a transaction or create an account, and the types of data that are optional. The description should also specify the types of data that are gathered in aggregate rather than for a specific user.
Use of information	A detailed description of how user data is used on the Web site. This description should specify whether user data will be used to send occasional e-mail messages to the users or to create personalized recommendations for them as they browse the site, for example.
Information transfer	A description detailing the situations in which user data will or might be transferred to a third party. If you do not currently plan to transfer user data but you can foresee a circumstance in which you might, describe that in this section. Specify the data that is or would be transferred, and the way it would be transferred.

For more information about privacy disclaimers, see *Web Design Specialist*, Lesson 11: Ethical and Legal Issues in Web Development. For an example of a thorough privacy statement, see the TRUSTe Privacy Statement at *www.truste.com/privacy-policy-full.html*.

1.3.6: International legal issues

Web users from all over the world can visit your Web site. Considering the breadth of your potential audience, you must be aware of some international issues if you develop Web sites that will conduct business with a worldwide audience. The following table describes some of these issues.

International Legal Issue	Description
Products	If the products your Web site sells are illegal in some countries, you may be responsible to provide ways to regulate sales and customers. If you are engaged in e-commerce, you should consult with a lawyer who specializes in international trade to determine whether your products are illegal anywhere in the world — and if they are, do not allow shipments to those regions. You should also post a statement on your site notifying residents in such regions that they cannot purchase the products in question.
Contracts	As with any business arrangement, it is important to establish some ground rules in a contract or other binding agreement. Contracts between parties that reside in different regions should include a clause stating the country or state whose laws apply to the agreement. You should consult with a lawyer before signing any agreement, particularly one with a company in a different country.
Intellectual property	The laws in the country where your business is located, including intellectual property laws such as copyright and trademark, may not apply in other countries. For example, U.S. copyright laws are very specific, but no international copyright law exists. The World Intellectual Property Organization (WIPO) is an international body that fosters cooperation among countries on intellectual property issues. However, WIPO cannot guarantee that all member countries will observe or enforce each others' intellectual property laws.
Fair use	Fair use, which allows the reproduction or quoting of copyrighted works for certain purposes such as criticism or education, is unique to the United States. However, other countries have similar concepts. Be aware that practices considered to be fair use in the United States might be considered copyright infringement in other countries. To avoid violating international intellectual property agreements such as those established by WIPO, you should closely comply with the concept of "fair use" established in your country of residence when developing Web sites.

For more information about international legal issues, see *Web Design Specialist*, Lesson 11: Ethical and Legal Issues in Web Development. For more information about WIPO, visit the Web site at *www.wipo.int*.

1.3.7: Nature and purpose of site content

When writing or overseeing the development of content for your Web site, keep in mind the purpose of the site, site section or page you are developing. Is it instructional or promotional? Is it intended for professionals only or for any readers? Most sites offer general information for all readers at the top level of the site, with more detailed information one or two clicks deep in the site. However, a site devoted exclusively to database administrators, for example, may have a technical tone from the top level all the way down.

You must always consider your audience when developing your Web site. The following table discusses issues related to the nature and purpose of Web site content, site audience and site message.

Site Content Issue	Description
Tone (or voice)	The style and tone of your site's text content must be appropriate for your intended audience in order to convey your site's message properly. Tone, or voice, is the manner in which the writer speaks to the reader — technical, professional, formal or informal, casual, conversational, and so forth. A site intended for children should have a very different tone than one intended for engineers, for example, which in turn would differ from sites intended for art enthusiasts, online shoppers, teenagers, political pundits or those seeking light-hearted entertainment.
Jargon	Jargon is the technical language specific to an industry or profession. Avoid jargon unless the site — or the area of the site for which you are developing content — is intended exclusively for members of a particular industry. You can define any jargon you use to avoid alienating users who are not familiar with the jargon terms.
Unintended audience / offensive content	If your content may be offensive to people for whom it is not intended, you may consider taking steps to avoid or explain this, such as posting a disclaimer or modifying the content in some way. Cultural perspectives can contribute to this concern. Be sure to get feedback about your site from people of different cultural backgrounds and national origins. You can accomplish this by holding a focus group, hiring a consultant, or even informally asking colleagues.
Offensive vs. illegal content	While it is generally legal to post content that may be considered offensive to some audiences, it may be illegal to post certain kinds of content. For example, content that can be defined as pornographic is illegal in some, but not all, regions. Publishing false statements about a person can be considered libel (which is defined as unjust inflammatory statements). Parodies (which imitate a person's or work's characteristics or style, and are sometimes used for intentional mockery) are typically protected by the First Amendment. However, if you are posting any potentially libelous or infringing information on your site, you should protect yourself by first consulting a lawyer.

For more information about the nature and purpose of site content, see *Web Design Specialist*, Lesson 16: Site Content and Metadata.

Practice exam questions

Following are some practice exam questions intended for student review.

1. In a Web site's privacy disclaimer statement, what does the Information Transfer section describe?

a. Ways in which the site's developers obtained or created the site's content
b. Purposes for which information gathered on a site will be used by that site
c. Technologies that are used by the site to gather information from site users
d. Situations in which information gathered on the site will or might be shared with third parties

Correct response: d. Situations in which information gathered on the site will or might be shared with third parties

Explanation: In a privacy disclaimer, the Information Transfer section details the situations in which user data will or might be transferred to a third party. It should specify the data that is or would be transferred, and the way it would be transferred. The Information Collection section provides a technical description of how user data is collected on the Web site. The Use Of Information section details the ways that collected user data is used on the Web site.

2. Which strategy or technology should you avoid using in your Web sites mainly for the reason that it limits your site's usefulness to certain audiences?

a. Spam
b. Pop-up windows
c. Pop-under windows
d. Single-browser technologies

Correct response: d. Single-browser technologies

Explanation: Some technologies and some design techniques limit your site's usefulness to certain audiences. Others commonly annoy users. You should avoid such technologies and techniques in your site designs, unless you have compelling reasons to use them and are not concerned about alienating some sectors of the Web audience. Single-browser technologies are generally those created by a vendor and thus functional only when that vendor's browser is used to view the page in which they reside. Spam and pop-up and pop-under windows are not recommended mainly because they annoy users and thus many users block them.

Domain 2: Web Design Elements

This section will discuss CIW Web Design Specialist Domain 2 exam objectives covering fundamental Web site design skills and concepts.

2.1: Aesthetics and viewer experience

This subdomain includes skills and knowledge required to use Web design principles to evaluate and develop a site's aesthetic qualities as well as its ability to enhance viewer experience.

2.1.1: Web page design and layout elements

Before you begin the Web production process, you must understand the elements that compose the completed Web page, and how they interact with each other, from a visual standpoint as well as a technical standpoint. You should then carry the chosen elements throughout the Web site.

Page layout refers to the way in which the Web designer presents information to users. The format should be logical and easy to understand. Like documents or reports, structured formats help categorize, simplify and clarify information for distribution. As a Web designer, you must develop a structure and adhere to it so the user's experience will be meaningful and productive.

Consider the elements discussed in the following table when planning your Web page layout. Determining which of these elements your site will include helps focus the development process. Many times, a site's design and functionality can be greatly improved by eliminating design elements from the page rather than by adding new elements.

Web Page Layout Element	Function
Space (also called white space)	Separates elements on the page and reduces page element clutter
Color	Conveys a visual tone or message, and provides a consistent look and feel
Font	Conveys a visual tone or message, provides a consistent look and feel, and enhances readability
Rule (i.e., line)	Divides the page content into related sections
List	Organizes information into items
Paragraph	Groups text characters on a page
Heading level	Creates various sizes of text to designate and organize content
Image	Provides visual appeal, information and navigation (includes symbols and pictograms)
Logo	Provides branding and site navigation
Margin	Controls the proximity of content to the browser window edges
Border	Can be applied to XHTML tables and frames
Navigation elements	Control the user's movement through the site
Table	Formats data into rows and columns; also used to distribute elements into position on a page
Frameset	Allows multiple pages to be displayed simultaneously

Web sites use some common basic structures to perform certain important functions. For example, navigation elements are generally located on the left and top margins. Additional navigational features are frequently included at the bottom of the page. Black text on a white background is common, as are

company logos in the upper-left corner of the display. These and other common layout features actually help each other and the user, which is the reason they are used so often.

Site layout can be categorized by the way the navigational elements are placed. The type of layout that is chosen depends largely on the type and amount of content. Common layout types include:

- The traditional left-margin layout, with the navigational elements in the left margin.
- The top-margin layout, in which navigational elements are placed along the top of the page.
- The distributed left- and top-margin layout, which is the most commonly used layout.
- The less commonly used right-margin layout.
- The distributed layout, which has navigation elements in all margins and works well for sites with an extensive amount of content.

Some designers use a typographic grid when laying out a page because the grid helps them achieve coherency. A typographic grid is a two-dimensional structure consisting of intersecting vertical and horizontal lines. As the vertical and horizontal lines of the grid intersect, boxes called grid units are formed. Text and graphics to be placed on the page are placed within the grid units. Because of the regular and balanced occurrence of grid units on the page, Web page elements placed within the grid units are aligned and balanced with each other. The grid becomes an underlying framework for the page, providing spaces in which to "hang" or attach various elements. The grid helps the designer create a balanced, harmonious composition.

The theory behind using grid-based layout is that aesthetics can be constructed. That is, if you use a well-defined grid that fits the purpose of your content, and then you place elements within the grid units, you will most likely have the basis for an aesthetically pleasing page.

There is no strict rule regarding how many grid units to create, or how far apart they should be spaced, or whether they should be spaced evenly. The key is to determine a grid that is suitable for the content you want to display.

For more information about Web page design and layout elements, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements, and Lesson 9: Web Graphics.

2.1.2: Design and audience participation

When users arrive at your site, their first impressions are important. The Web site may be the only window to the world for your business. If it is unappealing to the customer, you may lose the customer to a competitor. Users do not want to see cluttered pages with irrelevant content and images. Content is essential; however, the layout and delivery are just as important.

Users usually visit your site because they want specific information, whether for research or purchases. The easier you make their tasks, the more likely you are to earn their business. Remember as you design your Web sites that the design choices you make can help your audience experience the site as you intend it or hinder your audience's participation.

Because colors and images convey information about the organization that created a Web site, you must take care to evaluate the effectiveness or appropriateness of the colors you choose based on your target audience and the image that the site's stakeholders want to convey.

The element of color plays a vital role in the perception and presentation of a Web site. A company's style, culture and mood can be conveyed by the colors presented in the site and how they blend, coordinate or contrast. Color is perceived as a representation of the type of culture and industry in which a company participates.

For more information about design and audience participation, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements.

2.1.3: Visually balanced page/site

Most users prefer Web pages with even distribution and some blank space, known in the development community as white space. Users do not appreciate useless clutter and masses of content on Web pages. Users quickly scan pages, and the more they must scan through, the more information they will miss. Each Web page should contain approximately 50 percent less text than a printed version of the same information would contain.

A Web page should be designed to offer information concisely. You should provide users with the basics then give them links to more depth and detail. Not every user wants every piece of information; let each user choose whether to go deeper with links. Also, do not divide a page just to make it shorter unless it is at a logical breaking point. Each page should be able to stand on its own.

After determining what users need to know, you can lay out the page to position content and provide content-free areas. The following table describes layout strategies you can use to design visually balanced Web pages.

Page Layout Strategy	Description
Transparent GIFs	A fully transparent image file can be inserted into the page as a blank-space placeholder. You can assign height and width attributes in the XHTML <img> tag to create the desired white space dimensions. Transparent GIFs can also be placed into table data cells to control column or row size if necessary. Keep in mind that the goal is to reduce clutter on the page, enabling users to easily scan the document and select links if they choose.
Tables	The W3C now recommends against using tables for page layout in favor of using CSS. However, tables are still widely used for this purpose, so the XHTML designer must understand the construction and use of tables in page-layout design. By default, everything in XHTML aligns to the left. Developers use XHTML tables to distribute content over the entire browser display. A table structure can be populated with content, and table borders can be set to zero so the user never sees the table.
Frames	Frames can be used for page layout, although they also play a vital role in navigation. Whether frames will be used in a site should be discussed early in the design and planning process because frames can dramatically affect layout and navigation. Frames affect other factors in the development and deployment of the site as well.
Layers and CSS positioning	Layered or stacked elements are another way of laying out pages, although they are supported differently for different browsers. The preferred way to lay out a page in XHTML is with CSS positioning. However, important differences exist in the ways that various browsers have implemented this standard, which sometimes make it less predictable than using tables.

Consider the three types of balance described in the following table. If you divide a Web page vertically down the middle, you can imagine the objects on either side of that line working like a see-saw: Balance can be related to the size or type (image or text) of objects on a page, as well as to the darkness or lightness of colors of objects.

Visual Balance Type	Description
Symmetrical balance	Symmetrical balance is created when all elements on one side of the page are mirrored on the other. Although they may not be exactly the same, the mirrored objects are similar in terms of color and weight. An example of symmetrical design is a site with two similar columns (of color, text or image content, etc.) on either side of a larger area.
Asymmetrical balance	Asymmetrical balance is created when a single object on one side of the page is balanced by a number of smaller (or more lightly colored or weighted) objects on the other side.
Radial balance	Radial balance is created when elements on the page radiate or spiral out from a central point. Radial balance is not used much in Web design. Examples of radial balance include the petals of a sunflower or a round stained-glass window.

Some designers use ratios when laying out a Web page. Ratios are often used for composition in art and print. For example, the golden ratio has been used in art and architecture for centuries. This ratio is evident in nature as the number Phi (1.618033988749895) and is found in many places, such as in the spirals of a nautilus shell, in the petal distribution of various flowers, and in the seed heads of numerous plants.

If you use the golden ratio to create a design grid for laying out a page, the result will be a page with a wider (main) column and a narrower (secondary) column. To create a design grid based on the golden ratio, do the following:

- Determine the width of your design area
- Divide the design width by Phi (1.62 is sufficient) to arrive at the width of the main column.
- Subtract the width of the main column from the design width to determine the width of the secondary column.

You can also use percentages for a flexible-width layout: 100% / 1.62 = 62 percent for the main column, leaving 38 percent for the secondary column. A flexible-width layout adjusts to the size of the browser window.

For more information about visually balanced pages and sites, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements, and Lesson 9: Web Graphics.

2.1.4: Color and contrast

The element of color plays a vital role in the perception and presentation of a Web site. A company's style, culture and mood can be conveyed by the colors presented in the site and how they blend, coordinate or contrast.

Color is perceived as a representation of the type of culture and industry in which a company participates. For example, a Web site with primarily hard, bright colors such as red, pink, yellow and green may give the impression of a flashy, artistic type of industry and culture, such as a high-tech software design or graphic art firm. By contrast, a site with more subdued colors such as white and soft blue or gray might be perceived as a more conservative or traditional organization.

Web developers must address these perceptions when selecting a color scheme for a company Web site. Background design also plays a role, even after the color scheme has been chosen.

Because colors and images convey information about the organization that created a Web site, you must take care to evaluate the effectiveness or appropriateness of the colors you choose based on your intended audience. Certain color schemes may have gained negative connotations on the Web. For example, red text on a black background is often associated with hostility or hackers. Other color schemes — such as the colors used in a specific country's flag — may suggest geographic or cultural biases. Images portrayed

in one color scheme may convey a different message in another (consider again a country's flag colors, or a puddle of liquid shown in blue compared to the same puddle shown in red).

For more information about color and contrast, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements.

2.1.5: Design strategies for user focus

The location of Web site elements within the user's browser window has a large effect on how the user will perceive your site. Usability studies have shown that site visitors focus on the site's content, rather than the navigation or user interface elements. You should design your sites so that the content important to your audience occupies at least 50 percent of the screen space, but preferably 80 percent. This guideline is particularly vital in the page area that a typical user sees without scrolling.

Eye-tracking is a technique used to study the ways that people focus on Web pages. By tracking the motion of users' eyes when they visit Web sites, researchers can discover which parts of a Web page a user looks at and for how long. Eye-tracking studies have consistently found that most Web users first look at the center of the page (where they expect content to be), then to the left of the page, then to the right. Users rarely look at the bottom of the page unless they are seeking something they expect to find there.

By considering these facts as you design and lay out your Web pages, you can learn to emphasize elements that you consider important. Ways to ensure this include the following:

- Conduct your own reviews of Web pages using a representative group of users. Assembling your own group can help ensure that you address cultural concerns.
- Consult with marketing and others to determine common browsing habits and ways that you can adjust your Web pages to focus user attention where you want it.

Composition is the placement or arrangement of visual elements, whether on a canvas, a printed page or a Web page. Composition involves the element(s) that draws the viewer's eye into the page, or the place where the viewer "enters" the page, and it deals with ways to lead the eye through the page.

When considering composition, the designer must determine what the central point of interest on the page will be, and then compose other elements of the page accordingly. Each page should have a focal point.

When viewing any form of visual communication, the eye is attracted to the point of highest contrast. You create a visual hierarchy in a Web page by varying text size, weight and color. Titles, subtitles and body text are used to create this hierarchy because the human eye is trained to look for the points of highest contrast first. By emphasizing the titles, and then to a lesser degree the subtitles, etc., you make the outline of the page readily apparent. This helps the visitor ascertain the organization of the page quickly.

A basic principle of composition is that the eye follows lines, gesture and movement in the page (whether these are real or implied), and that it rests temporarily on masses before moving on. Masses include images and shapes (remember that words and paragraphs are shapes too). Therefore, the placement of elements on your page will affect or guide how the eye moves around the page.

Human faces and gestures in images play a big part in overall compositional flow. Be sure to place these types of elements so that they point into (and not off of) the page. Our eyes also tend to move around the edges of shapes, including the page or browser borders, so placing visual cues along these boundaries redirects the eye back into the page to discover other levels of your visual hierarchy.

For more information about design strategies for user focus, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements, and Lesson 9: Web Graphics.

2.1.6: Tools for site's visual consistency

Visual consistency within a site helps users to understand your site. Using layout techniques that are common within your site — and those common to all sites — helps Web site visitors to use your site more comfortably and effectively. The following techniques can help you establish visual consistency throughout your site:

- Using a limited and specific color scheme throughout the site makes each page look like it belongs to the whole.
- Using pre-designed page templates helps you to quickly and consistently style each page to your site's look and feel.
- Placing an image (such as the company logo) in the same place on every page increases visual consistency and aids navigation.
- Creating and following a style guide — a company's collection of guidelines and specifications for standardizing the appearance and tone of the Web site or other products — also helps designers focus on the details of presentation, such as color schemes, fonts, language usage, content tone and more.

A style guide is a standards document or manual that establishes a set of conventions or rules for performing common tasks. Applying the conventions specified in a style guide helps to ensure that the site has visual consistency, making it easier to use and more professional in appearance. Adopting or creating a style guide for your organization will standardize your company's look and feel, and define a common format for your Web site. The style guide is also useful for integrating documents from many departments or individuals. Some topics you may want to include in a style guide are:

- Default body text styles.
- Default background images or colors.
- Rules for capitalizing or styling heading levels.
- Preferred methods for emphasizing text (e.g., italic or bold).
- Guidelines for use of images and logos (e.g., size, format and scaling).
- Templates for standard pages (e.g., content pages, forms, press releases).
- Guidelines for using navigation links and image maps.

From a composition standpoint, keep in mind that the human mind groups objects that have similarities in shape, color, size, location and proximity. Apply the same typeface, color, weight, etc., to all the objects in a group so that a viewer can easily identify all the elements that comprise the group.

Grouping related elements on a page helps to define the page's structure. When we first glance at a page, we try to group the elements in our minds in order to simplify the visual field. We use grouping to ignore the details and grasp the main concepts. A good designer facilitates this process by grouping Web page elements on the page.

For more information about tools for a site's visual consistency, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements, Lesson 9: Web Graphics, and Lesson 15: Cascading Style Sheets.

2.1.7: Site message, culture and tone

The style and tone of your site's text content must be appropriate for your intended audience in order to convey your site's message properly. Tone, or voice, is the manner in which the writer speaks to the reader — technical, professional, formal or informal, casual, conversational, and so forth. A site intended for children should have a very different tone than one intended for engineers, for example, which in turn would differ from sites intended for art enthusiasts, online shoppers, teenagers, political pundits or those seeking light-hearted entertainment.

The element of color also plays a vital role in the perception and presentation of a Web site. A company's style, culture and mood can be conveyed by the colors presented in the site and how they blend, coordinate or contrast. Color is perceived as a representation of the type of culture and industry in which a company participates. For example, a Web site with primarily hard, bright colors such as red, pink, yellow and green may give the impression of a flashy, artistic type of industry and culture, such as a high-tech software design or graphic art firm. By contrast, a site with more subdued colors such as white and soft blue or gray might be perceived as a more conservative or traditional organization.

Because colors and images convey information about the organization that created a Web site, you must take care to evaluate the effectiveness or appropriateness of the colors you choose based on your intended audience. Certain color schemes may have gained negative connotations on the Web. For example, red text on a black background is often associated with hostility or hackers. Other color schemes — such as the colors used in a specific country's flag — may suggest geographic or cultural biases. Images portrayed in one color scheme may convey a different message in another (consider again a country's flag colors, or a puddle of liquid shown in blue compared to the same puddle shown in red).

As you consider your audience, make sure that you identify the following:

- Color combinations that might be attractive or acceptable to specific audiences or cultures
- Color combinations that might be unattractive to specific audiences or cultures
- Symbols, objects or images that may attract or repel an audience

Remember that colors and images that attract one audience might offend another.

For more information about site message, culture and tone, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements.

2.1.8: Eliminating unnecessary elements

Many times, a site's design and functionality can be greatly improved by eliminating design elements from the page rather than by adding new elements. Knowing which layout elements to use and which to omit is at least as important as knowing the technical aspects of using different XHTML and multimedia elements.

Most users prefer Web pages with even distribution and some blank space, known in the development community as white space. Remember that users do not appreciate useless clutter and masses of content on Web pages. Users quickly scan pages, and the more they must scan through, the more information they will miss. This fact does not mean that you cannot provide the user with a lot of information, just not all on one page. Each Web page should contain approximately 50 percent less text than a printed version of the same information would contain.

A Web page should be designed to offer information concisely. You should provide users with the basics then give them links to more depth and detail. Not every user wants every piece of information; let each user choose whether to go deeper with links. Also, do not divide a page just to make it shorter unless it is at a logical breaking point. Each page should be able to stand on its own.

After determining what users need to know, you can lay out the page to position content and provide content-free areas.

For more information about eliminating unnecessary elements, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements.

2.1.9: Typographical issues in printable content

Content that appears properly in a browser window will not necessarily appear the same when printed on paper. If you expect that users will print content from your Web site, you must be aware of certain issues and differences between the Web and paper media, and then design your site with printing in mind.

When a user clicks the Print button on his or her browser, the browser will attempt to scale the width of the current Web page to paper size (about 650 pixels for 8.5x11-inch paper). If the page consists mostly of text, it can usually print well without a problem. However, if the design is rigid, includes large images or contains objects positioned outside the printable area, then the browser will generally crop the page, and the right side of the page will not print past the paper width. Also consider that many users will be printing your pages using a black-and-white printer. If your site makes extensive use of typographical effects such as various font colors and backgrounds, this information may be lost when users print the page.

To make your Web pages usable in a printed format, you can design printable versions of pages that you think users are likely to print. Printable versions of Web pages are typically simplified versions of the page that usually contain all of the content, but none of the navigation and other elements that are not useful when printed. These pages should also use standard fonts such as Times New Roman or Arial, and UTF-8 coding. Alternatively, you can offer printable content in a different, print-friendly format, such as a word-processing document, presentation slides or Portable Document Format (PDF).

For more information about typographical issues in printable content, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements, and Lesson 9: Web Graphics.

2.1.10: Screen resolution issues

You must consider screen resolution during development because the appearance of a Web page will vary on different monitors based on their resolution settings.

Historical screen resolutions include:

- 640x480 (VGA) — now considered legacy.
- 800x600 (Super VGA).
- 1024x768 (XGA).
- 1280x1024 (SXVGA).
- 2048x1536 (UXGA).

Several years ago, the most common setting was 640x480; this setting was considered the lowest common denominator. However, no one uses this resolution anymore. Most Web users are now using a resolution setting of 1024x768 or higher. It is acceptable today to consider either 800x600 or 1024x768 as the lowest common denominator. If you design for 1024x768 resolution, for example, you can still make your design accessible to those using lower resolution if you use percentage values instead of pixels to help adjust the screen display for the lower-resolution monitors. Additionally, JavaScript programs can be used to direct low-resolution users to alternative pages that are better suited for their display setups.

It remains popular among very conservative designers to assume an 800x600 screen size. However, this resolution is overly conservative in many situations, as the majority of people use resolutions of 1024x768. This size will likely change as technology improves.

If you know that much of your audience is still using 800x600, then you should design to that lowest common denominator.

When you are designing the layout of your site, it is also important to consider the growing number of people who surf the Web on mobile devices, also known as personal digital assistants (PDAs). It is often well worth the time to create separate pages or at least separate style sheets for PDA-based browsers, because the individuals who use them often have disposable income they can spend on your site.

For more information about screen resolution issues, see *Web Design Specialist*, Lesson 4: Web Development Teams, Lesson 5: Web Page Layout and Elements, and Lesson 9: Web Graphics. A good

resource for display issues can be found on the About.com site at the following URL: *webdesign.about.com/sitesearch.htm?terms=resolution&SUName=webdesign&TopNode=3042*

2.1.11: Applying a single W3C standard consistently

The key to developing an accessible Web site is not to write each new page to the latest standard, but to select one Web language standard and apply it consistently throughout your Web site.

Consistency throughout your Web site is essential to developing accessible pages; therefore, you should adopt a "pick it and stick to it" approach. Whichever X/HTML standard you choose, apply that single standard to all your pages, and validate all your pages to that standard.

For more information about applying a single standard consistently, see *Web Design Specialist*, Lesson 13: XML and XHTML.

Practice exam questions

Following are some practice exam questions intended for student review.

1. In a visually balanced Web page design, which of the following describes symmetrical balance?

a. A page has adequate white space to promote readability.
b. Elements on the page radiate or spiral out from a central point.
c. Each object on one side of the page is mirrored on the other side.
d. Multiple objects on one side of the page balance one larger object on the other side.

Correct response: c. Each object on one side of the page is mirrored on the other side.

Explanation: Symmetrical balance is created when all elements on one side of the page are mirrored on the other. Although they may not be exactly the same, the mirrored objects are similar in terms of color and weight. Asymmetrical balance is created when a single object on one side of the page is balanced by a number of smaller (or more lightly colored or weighted) objects on the other side. Radial balance is created when elements on the page radiate or spiral out from a central point.

2. Which of the following can be used to establish standards for visual consistency on a Web site?

a. Cookies
b. Style guide
c. White space
d. Radial balance

Correct response: b. Style guide

Explanation: Visual consistency within a site helps users to understand your site. Creating and following a style guide — a company's collection of guidelines and specifications for standardizing the appearance and tone of the Web site or other products — helps designers focus on the details of presentation, such as color schemes, fonts, language usage, content tone and more.

2.2: Navigation, usability and accessibility

This subdomain includes skills and knowledge required to use Web design principles to enable navigation, usability and accessibility.

2.2.1: Site characteristics and strategies

The Internet is a medium that enables the user to choose which information to access and when to access it. This fact makes the Internet a one-to-one medium as opposed to a broadcast medium. Thus, the concepts and applications of mass media are not necessarily valid for the Internet.

By its nature, the Internet is transactional. The entire Internet experience, from logging on to Web browsing, is predicated on user requests and server responses — in other words, transactions and interactivity. Furthermore, by its nature the Internet is non-linear. The user constantly makes transactional decisions, first leading to and arriving at the site, then navigating within the site, performing searches (often within the site's integrated databases), conducting e-commerce, and finally deciding to return to the site. However, users can switch to another site — and another business — any time they choose.

Characteristics such as interactivity, navigation and database integration set Web sites apart from other media that do not implement these strategies. The following table discusses these key Web site characteristics that you may include in a site.

Web Site Characteristic	Description
Navigation	Navigation controls the user's movement through the site. Clear and easy navigation is critical to a site's success, and therefore must be well planned and carefully implemented to enable visitors to effectively use your site. You can use many types of navigational elements in a site, including text and image links, labels, positional indicators, menus, searching features and more.
Interactivity	Interactivity is a feature of Web sites that distinguishes it from other media types. The Web is transactional in nature because Web use is based on the interactions between Web users and the sites they visit and explore. Some Web sites offer little interactivity beyond providing several pages for users to browse among and read. Other sites offer a high level of interactivity, with forms for users to complete and submit, multimedia to watch or listen to, games to play, catalogs to search, and products to buy. Interactivity that serves your site's purpose well can attract users to your site, persuade them to stay longer, and give them reasons to return.
Database integration	Database integration allows Web pages to use dynamic data, often in conjunction with interactivity. Databases provide the ability to store and sort vital information, such as customer data and product information. For example, your site might use a database to store and inventory information about the products your company sells. A visitor can thus search for a specific product, and your Web site can retrieve this data and display it in a Web page for the user to view, check inventory, make selections and so forth. The site designer's job role does not necessarily include the skill of database integration, but it is an important and common site characteristic that you may need to incorporate into your design and page layout features.

For more information about site characteristics and strategies, see *Web Design Specialist*, Lesson 1: Overview of Web Design Concepts, and Lesson 5: Web Page Layout and Elements.

2.2.2: Site hierarchy/architecture concepts

The following table discusses concepts relating to Web site hierarchy and architecture.

Site Hierarchy / Architecture Concept	Description
Information architecture	A hierarchy or organizational chart visually defines the dynamics of an organization or even a home, in which parents are at the top and children are layered below. In a Web site, the home page or site entrance is at the top; below are the other pages, which are referred to as child pages. These pages do not follow a linear model; they branch out and grow to various depths, depending on how much content supports each branch. This structure is known as the information architecture of the site.
Site structure	Site structure describes how a Web site is stored on the Web server. Web sites are structured in a hierarchical format. Each directory is subdivided into more directories. This arrangement creates a file hierarchy that helps you manage your system. The Web server stores files and images in directories that you designate. Structuring files on the server into directories and subdirectories is recommended to maintain order and manageability.
Site structure and directory / file names	File and directory names play an important part in Web page addressing. Together, file names and URLs help give Web users a better understanding of a Web site's structure. It is important to name directories and files on the Web server descriptively to aid in navigation and organization. File names can be as helpful as directory names. If the name of the HTML file confirms to the user which page he or she is viewing, it helps in the process of navigation and awareness.
Site structure and URLs	Web sites are structured in a hierarchical format. This format provides the information that a browser needs to find a given Web page using a Uniform Resource Locator (URL). URLs can be used to determine location and depth. For example, a user who is unsure where he or she is in the site hierarchy could look at the URL in the browser window to help visualize his or her location. This technique only works if the directories and files were given meaningful names.
Navigation depth and paths	Depending on your Web site's purpose, remember the "three-click rule" for managing user access to linked files: Users should not need to click more than three times during their navigation of a site to find the information they seek. This guideline is especially important for informational or research-oriented Web sites. Unclear navigational paths will confuse users and discourage them from returning to the site. Also consider whether any multimedia will provide the navigational framework for the site or merely supplement it. For example, using an Adobe Flash navigation bar on your site will present navigational problems for users who cannot support Flash. A clear sense of direction is very important to the user.
Navigation features in page design	Browsers provide limited navigational support. Therefore, you should design your Web site with extensive navigation to help users overcome the limitations of their software. These include: - Site identifiers (e.g., a corporate logo on every page). - Landmark page access (e.g., a link to the home page on every page). - Browser history. - Clear information architecture. - Default link colors for visited and unvisited pages. - Site map.
Navigation icons and controls	Navigation toolbars should be simple, easily understood and intuitive. Icons are very popular and most people are familiar with them. Icons include two types: labeled and unlabeled. If users must guess where the icon link will take them, the icon should be labeled. Other common navigation elements include (but are not limited to): - Buttons. - Image maps and hot spots. - Rollover mouse events. - Arrows. - Navigation bars and menus. - Drop-down select menus.

Table cont'd

Site Hierarchy / Architecture Concept	Description
Positional awareness	Positional awareness includes understanding any parent, peer or child pages relative to the current location. Users should never wonder where they are within the site. At all times, there should be indicators that users can reference for positional awareness. Many techniques can be used to provide the user with this important information (see the following table).

Positional awareness is particularly important for e-commerce sites. If users become lost while they are shopping, they are likely to abandon their purchases and visit competitors. The following table describes many techniques that can be used to provide the user with positional awareness.

Positional Awareness Technique	Description
Headings	Current page headings provide an effective means for users to determine where they are within the site. A limitation of using only headings for positional awareness is that they provide only the current location, not the relative location of parent, peer or child pages.
Images	Images provide useful indicators of positional awareness. For example, consider the tabs in a day-planner. All or most of the tabs are always visible, each tab has a name, and the tabs change appearance at each location. This scheme provides better positional awareness because many peer and parent pages are visible from any given location.
Path markers	Symbols such as arrows or lines indicate the navigational path that the user took to arrive at the current location. These are helpful as the user moves through multiple pages on many levels of the site. Path markers are also called "bread crumbs."
Site map	The site map exists in a separate HTML document, and each page of your site can be represented by a simple text or graphical description. Users can then follow links from the site map to the desired page. The disadvantage is that the user must leave his or her current location to visit the site map.
Colors	Some sites use colors to indicate position by applying different color values to different content. One limitation of using colors is that the user must first understand the color scheme and then associate it as he or she moves through the site. Another drawback is the ineffectiveness of this strategy for visually impaired or colorblind users.

For more information about site hierarchy and architecture concepts, see *Web Design Specialist*, Lesson 8: Navigation Concepts.

2.2.3: Common navigation conventions

Many common elements used throughout the World Wide Web have become familiar to users and therefore have predefined meanings. Some of these elements can be referred to as labels. For example, when users see a button or link labeled "Home Page," they know exactly where that link will take them. Do not attempt a clever label such as "Entry Point" for a link to the home page.

Following are some additional common links and labels:

- Corporate logo or branding image, which should be linked to the home page; users have come to expect this link.
- Search or Find, which links users to a page at which they can conduct a site search.
- FAQ (Frequently Asked Questions).
- Downloads.
- News.

- Site Map.
- About Us.
- Contact Us.
- Help.

Do not try to be unique; it is wise to use recognized labels because users expect them.

Another popular technique for helping users through a convoluted process is to guide them. Provide a link to the next step, and continue the process by establishing links that keep users on track. These links should supply the necessary information, as well as an alternative course clearly marked to allow the user to exit. For example, an online purchase should lead the user through shipping information, then on to payment information, then to receipt information. This technique is called guided navigation. If you use a graphic interface, it must be intuitive. The audience must be able to navigate easily through the site and recognize which elements are clickable.

The term "mystery meat navigation" was coined by Vincent Flanders to describe a Web page or graphical user interface (GUI) in which it is difficult for the user to determine the destination of hyperlinks, or in more extreme cases, to locate the hyperlinks on the page at all. As a result, the user has great difficulty determining the site's navigation structure. A site that suffers from mystery meat navigation is less accessible and less useful than sites that feature clear and simple navigation. Mystery meat navigation can also cause accessibility issues because screen readers may not be able to interpret the navigation.

For more information about common navigation conventions, see *Web Design Specialist*, Lesson 4: Web Site Development Process, and Lesson 8: Navigation Concepts. You can view a Flash video of Flanders' original example of mystery meat navigation at the following URL: *www.webpagesthatsuck.tv/saturn/saturn.html.*

2.2.4: Navigation action plan

Navigation is difficult partly because it is subjective. Every designer has different opinions about which strategies work best. This observation is extremely important from a usability perspective. If the site's navigation does not function as it should, the site may risk losing customers.

When navigation works well, users flow smoothly from action to action, from page to page. Navigation must be tested; this process takes careful planning. When designing your Web sites, remember the practices discussed in the following table.

Practice for Navigation Action Plan	Description
Determine users' goals and needs	Navigation design is about predicting your site users' actions and building a site to support them. To do this, you must understand your audience's goals and needs. Determining your users' goals and needs may require you to interview the people who will use the site. In the case of an intranet, spend time talking with employees about their daily activities. For a commercial site, talk to potential customers and discuss their preferences. Try to learn the goals behind their comments. Real solutions must be uncovered, not constructed.

Table cont'd

Practice for Navigation Action Plan	Description
Learn from navigation that works	Much can be learned from studying sites that are easy to navigate. Consider the elements common to successful navigation, regardless of the approaches used. Those sites that share certain qualities display good navigation planning. Generally, good navigation includes several characteristics: - Offers easy-to-learn elements - Remains consistent - Provides feedback - Appears in context - Offers alternatives - Provides clear visual messages - Offers clear and understandable labels - Remains appropriate to the site's purpose - Supports user goals and behaviors
Go deeper than the home page	Even on sites with truly poor navigational design, it is usually possible to move from the home page to an interior page without incident. Most navigation problems tend to involve being stranded inside a site. Planning your home page is only a small fraction of your job as a navigation designer. The deeper you can plan your site's structure and navigation, the more successful your site's navigation will be.
Provide quick links	After you consider the structure of the site as a whole, spend some time thinking about shortcuts through this structure. Shortcuts provide quick links and easy access to small pieces of content. Some common shortcuts include search features, site maps and tables of contents. The goal of shortcuts is to help users find information easily and rapidly. Design your shortcuts with this goal in mind.
Design for various user preferences	An example of how users' personal preferences affect your approach to navigation can be found in searching and browsing behaviors. Some people like to search for the exact information they need, and they refuse to browse unless forced. Others prefer to browse through information to find what they need. By providing multiple ways to find information on your site, you can accommodate different styles of Web use.

Navigation design is complex, but it is key to helping your users accomplish their goals. Remember that a navigation system that supports users is vital to the success of your Web site.

For more information about navigation action plans, see *Web Design Specialist*, Lesson 8: Navigation Concepts.

2.2.5: Multimedia purpose

Designing, implementing and maintaining a Web site that can deliver content with advanced multimedia capabilities and state-of-the-art interactivity is easier than it was in the past. New applications are constantly introduced to put bigger and better multimedia on the Web. In the current developmental stage, no multimedia standard is agreed upon to which every Internet application developer can work. The challenge in building multimedia-rich Web sites is in deciding which methods, resources and tools are necessary for successful deployment. With the growth in consumer interest and online buying activity, your question becomes this: How can my company best use its knowledge of Web interactive programs and multimedia content to successfully reach customers before the competition does? This business question differs from the traditional only in its emphasis on Web technology.

A Web page gives its author only a few seconds to convince the reader that the information inside is of interest. The page's "look and feel" (its overall impression of appearance and functionality) is an important tool in this effort. Graphics play an important part in your user's experience. For this reason, interesting

graphics are important at this level of Web development. The first page of your site must also display exact pathways for the user to gather information; this navigability should also help users in an intuitive way.

One of the most common misconceptions about Web design is that a good site must dazzle the user with a multimedia experience, and that the content of the site is of secondary importance. As a Web designer, you want your site users to have a satisfying experience, but dazzling them is not necessarily your goal. The primary goal in Web design is to give users what they want. This goal can be achieved with a complex balance of well-planned design, high-quality content, and proper use of available media. If multimedia makes sense and enhances the usability of a site, you should use it. If multimedia does not enhance the user experience, or if it degrades the user experience by creating an unnecessarily long download, then you should not use it.

The key difference between Web design and design for traditional media such as print or television is interactivity. Web designers must be aware of the way that information is presented on the screen, and also of the ease with which site visitors can use the site's navigation and other interactive elements.

For more information about multimedia purpose, see *Web Design Specialist*, Lesson 1: Overview of Web Design Concepts, and Lesson 10: Multimedia and the Web.

2.2.6: User-accessibility standards and laws

Web pages should be accessible to all people, including those with disabilities. To assist in this mission, the World Wide Web Consortium (W3C) has created the Web Accessibility Initiative (WAI). WAI aims to ensure that core technologies used on the Web, such as HTML, CSS, XML and DOM, are equally accessible to users with physical, visual, hearing and cognitive disabilities. The WAI works with various W3C Working Groups to ensure that the standards for various W3C technologies include accessibility options.

The WAI Web Content Accessibility Guidelines (WCAG) 1.0 specification divides conformance requirements into a hierarchy with three levels. Each level of conformance encompasses a specific set of checkpoints, each with an assigned priority level. All checkpoints are organized under 14 specific guidelines. The guidelines are developed with consideration for groups of users with specified disabilities or needs. Although different situations should be considered when designing Web documents, each accessible design choice generally benefits several disability groups, and the Web community as a whole.

In 2001, the U.S. government implemented Section 508 of the Rehabilitation Act: Electronic and Information Technology Accessibility Standards. Section 508 requires that all electronic and information technology developed, procured, maintained or used by federal agencies be comparably accessible to users with disabilities. Section 508 is based on the Priority 1 and 2 checkpoints of the W3C's WAI Web Content Accessibility Guidelines (WCAG) 1.0.

For more information and a hands-on lab about user-accessibility standards and laws, see *Web Design Specialist*, Lesson 6: Web Site Usability and Accessibility. For information about the W3C's Web Accessibility Initiative (WAI), visit *www.w3.org/WAI, www.w3.org/WAI/wcag-curric* and *www.w3.org/WAI/about.html*. To view the WAI Web Content Accessibility Guidelines (WCAG) 1.0 specification, visit *www.w3.org/TR/WAI-WEBCONTENT*. To learn more about Web page accessibility for disabled users, visit *www.ada.gov/stdspdf.htm*. You can learn more about Section 508 by visiting *www.access-board.gov/508.htm* and *www.section508.gov*.

2.2.7: User-accessibility challenges and solutions

The World Wide Web Consortium (W3C) has created the Web Accessibility Initiative (WAI) to help make Web pages accessible to all people, including those with disabilities. WAI aims to ensure that core technologies used on the Web, such as HTML, CSS, XML and DOM, are equally accessible to users with physical, visual, hearing and cognitive disabilities.

For example, a person with a visual disability may be unable to view a multimedia presentation on the Web. One way to solve this problem is to include text equivalents of the presentation in the code. The multimedia player, such as RealNetworks RealPlayer or Microsoft Windows Media Player, could then access the text equivalent and present it to the user in Braille or as speech.

The WAI works with various W3C Working Groups to ensure that the standards for various W3C technologies include accessibility options. For example, the HTML standard supports improved navigation, extended descriptions of complex graphics, and multimedia captions. It also supports device-independent user interface descriptions that allow users to interact with Web pages using a mouse, keyboard or voice input.

Another example is the HTML <noframes> tag, which allows Web page developers to specify alternative text that will appear to users whose browsers cannot support frames (older browsers, adaptive browsers used by those with disabilities, mobile and alternative browsers, and other non-standard browsers). The alternative text can inform these users that the Web page will not appear as intended, and can specify an alternative version of the site or an alternative resource for the information provided.

Consider the challenges inherent in making Web pages accessible to people with disabilities. The WAI offers many good solutions that every Web developer should be aware of. Although different situations should be considered when designing Web documents, each accessible design choice generally benefits several disability groups, and the Web community as a whole. Following are the guidelines defined by WAI:

- Provide equivalent alternatives to auditory and visual content.
- Do not rely on color alone.
- Use markup and style sheets properly.
- Clarify natural language usage.
- Create tables that transform gracefully.
- Ensure that pages featuring new technologies transform gracefully.
- Ensure user control of time-sensitive content changes.
- Ensure direct accessibility of embedded user interfaces.
- Design for device independence.
- Use interim solutions.
- Use W3C technologies and guidelines.
- Provide context and orientation information.
- Provide clear navigation mechanisms.
- Ensure that documents are clear and simple.

For more information about user-accessibility challenges and solutions, see *Web Design Specialist*, Lesson 6: Web Site Usability and Accessibility, and Lesson 14: Web Page Structure — Tables and Framesets. For information about the W3C's Web Accessibility Initiative (WAI), visit *www.w3.org/WAI, www.w3.org/WAI/wcag-curric* and *www.w3.org/WAI/about.html*. To view the WAI Web Content Accessibility Guidelines (WCAG) 1.0 specification, visit *www.w3.org/TR/WAI-WEBCONTENT*. To learn more about Web page accessibility for disabled users, visit *www.ada.gov/stdspdf.htm*.

2.2.8: Written site content

Your Web site's written content is generally most of the data your site provides to Web users. Writing content for the Web is different from writing for marketing materials or other printed media because people do not read as much on the Web as they do in print — rather, they tend to scan on the Web, looking for relevant data. Following are some tips to consider when developing Web content:

- Keep text concise, clear and to the point. Professional writers and editors can help you achieve this goal.
- Divide long paragraphs into multiple paragraphs.
- Divide long articles or information areas into multiple pages.
- Stay on subject, and keep text relevant to the purpose of the site or page on which it appears. Your site's message will be stronger.
- Use language style and tone consistently to help strengthen your site's message. Professional editors and style guides can help apply this type of consistency.
- Use industry-specific jargon sparingly, if at all. Jargon can be appropriate if it is familiar to your audience, but consider that it may alienate some users who do not know the meanings of jargon terms. Unless you know your audience will be exclusive and understand the terms, you should define any jargon you use.
- Consider how your site will perform in search engines. Use keywords and phrases by which you want your site to be indexed and found.
- Avoid misleading or inaccurate statements about products and services. There is no advantage to keeping users at your site longer only for them to eventually learn that they have wasted their time there. Deliberately false claims may even invite legal repercussions.

For more information about written site content, see *Web Design Specialist*, Lesson 2: Web Development Teams, and Lesson 16: Site Content and Metadata.

2.2.9: Audience and end-user capabilities

Generally, Web designers are likely to have the most recent hardware, software and plug-ins. However, the vast majority of the Web audience is not seeking high-end design or an exciting multimedia experience. As a designer, your job is usually to create a Web site that meets the stakeholders' needs and that can be used by the site's target audience. If you are designing a site that will be used by anyone on the Web who finds your site, then the range of your audience's potential browsers and bandwidth limits your technology choices.

A good guideline to follow when designing for the masses is to design for the lowest common denominator. Although there is no real agreement on what the lowest common denominator is today, the following guidelines will help you choose "safe" specifications:

- Many users still have 15-inch monitors.
- Some users set resolution to 800x600, but 1024x768 (or higher) is more common today.
- Many users still use modem connections, although modems are most likely 56 Kbps or faster.
- Most users have 4.x or newer versions of browsers.
- Although you should always be careful about using plug-ins, some plug-ins such as the Flash Player and MP3 players are very commonly used.

If you are designing for a technical audience, or for an audience you know to generally have higher-end computers and monitors (such as graphic designers), you can adjust your specifications upward.

Even though the lowest common denominator system specifications continue to move upward, it is still important to try to accommodate users with lower-end systems when possible. You can do this through the use of alternative, text-only versions of multimedia rich pages, or by checking for browser and plug-in versions and informing users if their settings do not meet the minimum requirements for your site.

For more information about audience and end-user capabilities, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements, and Lesson 10: Multimedia and the Web.

Practice exam questions

Following are some practice exam questions intended for student review.

1. Which site characteristic can be described as the element that keeps Web users who are just browsing engaged in your site?

a. Navigation
b. Advertising
c. Interactivity
d. Database integration

Correct response: c. Interactivity

Explanation: Interactivity is a characteristic of Web sites that distinguishes them from other media types. Interactivity that serves your site's purpose well can attract users to your site, persuade them to stay longer, and give them reasons to return. Navigation controls the user's movement through the site. Database integration provides the ability to store and sort vital information, allowing Web pages to use dynamic data.

2. Which concept refers to the hierarchical format in which your Web site's files and directories are stored on a Web server?

a. Site map
b. Site structure
c. Navigational depth
d. Positional awareness

Correct response: b. Site structure

Explanation: Site structure describes how a Web site is stored on the Web server. Web sites are structured in a hierarchical format. Together, file names and URLs help give Web users a better understanding of a Web site's structure.

Domain 3: Basic Web Technologies

This section will discuss CIW Web Design Specialist Domain 3 exam objectives covering fundamental Web technology skills and concepts.

3.1: Basic X/HTML

This subdomain includes skills and knowledge required to use basic HTML and XHTML (X/HTML) to develop a series of Web pages.

3.1.1: X/HTML origins, standards and versions

Note: This guide sometimes refers to X/HTML to signify an interchangeable reference to HTML and/or Extensible HTML (XHTML), the most current HTML standard.

Hypertext Markup Language (HTML) is based on SGML and is the traditional authoring language used to develop Web pages for many applications. Tim Berners-Lee of the Massachusetts Institute of Technology invented HTML with colleagues from CERN (the European Particle Physics Laboratory) as a means of distributing nonlinear text, called hypertext, to multiple points across the Internet. One document links to another through pointers called hyperlinks. A hyperlink is an embedded instruction within one text file that calls another file when the link is accessed, usually by the click of a mouse. The global set of linked documents across the existing Internet framework grew into what is now known as the World Wide Web.

A markup language is very different from a programming language. Program files and data files exist separately in traditional applications. In a markup language, the instructions and the data reside in the same file. In addition, HTML does not provide data structures or internal logic, as do programming languages such as C and Pascal.

HTML is a cross-platform language that works on Windows, Macintosh and UNIX platforms. HTML and the Web are client/server systems. Like SGML, HTML facilitates data exchange through a common document format across different types of computer systems and networks on the Web. Whereas SGML is used specifically to define context as opposed to appearance, however, HTML has evolved into both a contextual and a formatting language. By applying a heading style to text using HTML, for example, you are not only marking that text contextually as an important line that begins a new section; you are also applying the visual formatting elements of boldface and a larger font size.

The HTML standard defines the individual elements that make up the language. These elements are the instructions, or markup tags, that indicate how HTML documents should be displayed in a browser. This standard ensures that hypertext is displayed consistently in different browsers and on various computer platforms. Standardization limits the proprietary software specifications that do not operate with other vendors' products.

The World Wide Web Consortium (W3C) is the standards organization that controls the evolution of HTML. When the W3C has fully endorsed a technology, it publishes a "recommendation" to the Internet community. A recommendation published by the W3C is the document that specifies all standardized elements of the given technology. The following table describes the various versions of the HTML standard.

HTML Standard Version	Notes
HTML 1.0	Essentially the first version of HTML used for graphical browsers. It is equivalent to the first HTML specification (version 0), with added support for images. Version 0 was used as a prototype to test the Web with the first generation of character-based browsers, such as Lynx from CERN.
HTML 2.0	Includes all the capabilities of version 1.0 plus support for user input fields, which is necessary for forms. In practice, browser developers such as Microsoft and Netscape have moved well beyond the HTML 2.0 standard.
HTML 3.0	HTML 3.0 was never ratified; instead, it evolved directly into HTML 3.2. As more browser-specific tags were introduced, it became obvious that a new standard was needed. For this reason, version 3.0 was abandoned in favor of version 3.2.
HTML 3.2	Added widely deployed features such as tables, applets and text flow around images, while providing full backward-compatibility with the existing standard HTML 2.0. The major enhancements included improved tags, and support for tables, figures, frames and mathematical equations.
HTML 4.0	The W3C standard for the last major revision of SGML-based HTML. It provides support for: -Style sheets. -Internationalization features (e.g., reading from right to left). -Accessibility features (issues involving Web users with disabilities). -Enhanced tables and forms. -Scripting and multimedia. HTML version 4.0 is specified in three variants or "flavors" (see next table).
HTML 4.01	HTML 4.01 includes a few minor modifications to the standard, and calls for support of Extensible HTML (XHTML). The 4.01 designation means that some minor errata have been corrected and the XHTML specifications were added. But the major revisions from version 3 to version 4 persist.

HTML version 4.0 is specified in three variants or "flavors," which are described in the following table.

HTML Flavor	Description
HTML 4.0 Transitional	This version takes advantage of HTML 4.0 features including style sheets, but does not rely on them for the benefit of those viewing pages with older browsers. HTML 4.0 dropped some common attributes and tags that style sheets now handle; HTML Transitional allows the use of some of these deprecated elements.
HTML 4.0 Strict	The HTML 4.0 Strict recommendation was the first version of HTML to disallow the use of layout and style elements. You can use this version with Cascading Style Sheets (CSS) to produce the font, color and layout effects you want. HTML Strict also does not allow the use of deprecated elements.
HTML 4.0 Frameset	The HTML 4.0 Frameset recommendation supports the use of frames to partition the browser window into two or more adjacent windows.

Extensible Hypertext Markup Language (XHTML) is the latest version of HTML. Essentially a rewrite of HTML 4.01 as an XML application (rather than SGML), XHTML was designed by the W3C to ease the transition from HTML to XML. XHTML requires stricter code syntax than HTML. XHTML-compliant documents can be used, viewed and validated by both HTML and XML processors. Therefore, developers can begin using some of the features of XML with the confidence that backward-compatibility issues with HTML will be addressed, as well as future issues with XML. XHTML is also designed to make Web documents accessible and interoperable across platforms, partly due to the strictness of XML.

Because XHTML is simply a reformulation of HTML 4.x, the specification includes the same three flavors as HTML 4.

XHTML Flavor	Description
XHTML Transitional	The Transitional flavor of XHTML is designed to enable Web designers to start taking advantage of the benefits of XML today while still being compatible with older Web browsers. XHTML Transitional includes all the HTML 4.0 features, such as style sheets, but does not rely on them. This variation of the standard allows you to use some deprecated elements.
XHTML Strict	The XHTML Strict recommendation does not allow the use of layout and style elements. This version requires the use of Cascading Style Sheets (CSS) to produce font, color and layout effects. XHTML Strict also does not allow the use of deprecated elements.
XHTML Frameset	The XHTML Frameset recommendation supports the use of frames to partition the browser window into two or more adjacent windows.

In XHTML, the document type declaration (<!DOCTYPE> tag) is required to specify the markup version used by the page, as well as the document's primary language. (In HTML, it is recommended for good coding practice but not required.) The DOCTYPE declaration is placed at the very top of your XHTML document. If you declare an XHTML DTD (e.g., XHTML Transitional), then the interpreter will read the document according to XHTML rules for syntax.

Note: Be careful not to confuse the document type declaration (<!DOCTYPE> tag) with the Document Type Definition (DTD). The <!DOCTYPE> tag is a statement that identifies code versions in a document. The DTD is a separate document containing a set of rules for structure, syntax and vocabulary, used commonly with XHTML and XML. A <!DOCTYPE> tag may contain a URL referencing a DTD document that applies the rules of the specified language version.

Following are descriptions and proper DOCTYPE declarations for the three flavors of XHTML 1.0.

XHTML Flavor	DOCTYPE Declaration
XHTML Transitional	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
XHTML Strict	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml1-strict.dtd">
XHTML Frameset	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "DTD/xhtml1-frameset.dtd">

For more information about X/HTML origins, standards and versions, see *Web Design Specialist*, Lesson 12: HTML and the Evolution of Markup, and Lesson 13: XML and XHTML. Read the W3C's HTML and XHTML summaries and specifications at *www.w3.org/MarkUp*, *www.w3.org/TR/html4* and *www.w3.org/TR/xhtml1*.

3.1.2: Basic X/HTML code

Knowledge of X/HTML is a prerequisite for CIW Web Design Specialist. You should be able to create a basic Web page using a text editor such as Notepad. CIW Site Development Associate courseware provides detailed instruction in XHTML code.

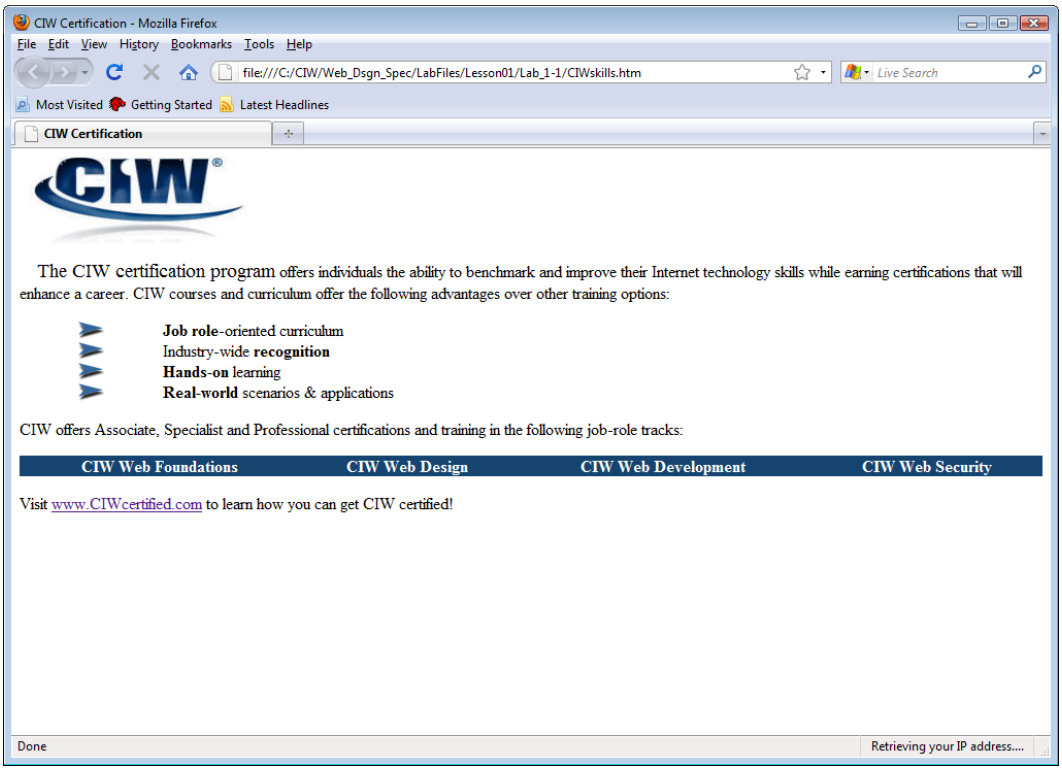
Following are the basic structure tags used to create an XHTML document:

- <!DOCTYPE>
- <html>
- <head>
- <title>
- <body>

Additional XHTML tags can be used to create or modify the following elements on a basic Web page:

- Table
- Font
- Color
- Hyperlink (using HTTP as the protocol)
- Image

Study the basic structure of the Web page shown in the following figure.



The following XHTML code was used to create the page shown in the preceding figure.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<title> CIW Certification </title>
</head>

<body bgcolor="#ffffff">

<a href="http://www.ciwcertified.com"><img src="ciw-logo.gif" alt="CIW logo" width="231"
height="84" border="0"/></a>
<p>
   <span style=”font-size:14pt”>
The CIW certification program</span>
offers individuals the ability to benchmark and improve their Internet technology skills
while earning certifications that will enhance a career. CIW courses and curriculum offer the
following advantages over other training options:
</p>
```



```
<table width="70%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td width="20%" align="center"><img src="arrow.gif" alt="arrow"/></td>
    <td width="80%"><b>Job role</b>-oriented curriculum</td>
  </tr>
  <tr>
    <td width="20%" align="center"><img src="arrow.gif" alt="arrow"/></td>
    <td width="80%">Industry-wide <b>recognition</b></td>
  </tr>
  <tr>
    <td width="20%" align="center"><img src="arrow.gif" alt="arrow"/></td>
    <td width="80%"><b>Hands-on</b> learning </td>
  </tr>
  <tr>
    <td width="20%" align="center"><img src="arrow.gif" alt="arrow"/></td>
    <td width="80%"><b>Real-world</b> scenarios &amp;amp; applications</td>
  </tr>
</table>
<p>
CIW offers Associate, Specialist and Professional certifications and training in the
following job-role tracks:
</p>
<table width="100%"
border="0" cellspacing="0" cellpadding="0" align="center">
  <tr bgcolor="#164470" align="center">
    <td><b><span style="color:#ffffff">
    CIW Web Foundations
    </span></b></td>
    <td><b><span style="color:#ffffff">
    CIW Web Design
    </span></b></td>
    <td><b><span style="color:#ffffff">
    CIW Web Development
    </span></b></td>
    <td><b><span style="color:#ffffff">
    CIW Web Security
   </span></b></td>
  </tr>
</table>
<p>
Visit <a href="http://www.ciwcertified.com">www.CIWcertified.com</a> to learn how you can get
CIW certified!
</p>
</body>
</html>
```

For more information and hands-on labs about basic X/HTML code, see *Web Design Specialist*, Lesson 1: Overview of Web Design Concepts, Lesson 2: Web Development Teams, and Lesson 23: Creating Web Pages Using Open-Source Tools. For information and tutorials about X/HTML, visit *www.w3.org* or *www.w3schools.com*.

3.1.3: Hexadecimal color values

Hexadecimal code values range from 00 to FF (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Hex code correlates to the base-10 values but is stated in hex (base-16) values. The value 00 represents no presence of a color, and the value FF represents maximum presence of a color.

The color white is represented in hexadecimal code as follows:

Red=FF

Green=FF

Blue=FF

Thus, the hexadecimal value for white is written as FFFFFF, which represents maximum presence of red, green and blue.

The color green is represented in hexadecimal code as follows:

Red=00

Green=FF

Blue=00

Thus, the hexadecimal value for green is written as 00FF00, which represents no presence of red, maximum presence of green, and no presence of blue.

As you can see, hex code assigns each color a two-character code for each red, green and blue value, whereas the RGB scheme uses digits between 0 and 255 for each red, green and blue value. RGB values are also separated by commas, whereas hex values are not. The following table shows some examples.

Color	RGB Value	Hex Code
RED	255,0,0	FF0000
GREEN	0,255,0	00FF00
BLUE	0,0,255	0000FF
WHITE	255,255,255	FFFFFF
BLACK	0,0,0	000000

When hex-code values are used in HTML, they are preceded by the # (number or hash) symbol, which is not required but is part of the HTML 4.0 specification. In the <body> tag, for example, the background color green is specified as follows:

```
<body bgcolor="#00FF00">
```

For more information about hexadecimal color values, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements. For a complete listing of RGB colors and their corresponding hexadecimal values, visit the Browser-Safe Web Palette page at *www.lynda.com/resources/webpalette.aspx* or the VisiBone Webmaster's Color Laboratory at *www.visibone.com/colorlab/*.

3.1.4: X/HTML layout elements

Raised in the WYSIWYG environment of the current word-processor generation, users wanted the ability to make type bold, change the font size or face, and add color when creating Web pages. Netscape, producer of the first widely successful browser, responded to users who were frustrated with HTML's limitations by creating extensions to the language. These extensions were tags that specified appearance formatting (presentational or procedural markup) rather than content structure (logical or structural markup). As these extensions became widely used, they were ultimately folded into later versions of HTML.

The following code demonstrates a page that contains both structural markup and procedural markup indicating formatting characteristics. The tags that define appearance but not structure appear in bold.

```
<title>Body Surfing</title>

<center>
<font face="Arial" color="#0000FF" size="6">
Body Surfing
</font>
```



```
</center>
<hr width="80%"/>
This document is designed to teach one how to enjoy <b>the ride of your life.</b><p>

You will find many good tips and tricks here. Read, and then go out and <i>try this on your
own!</i>

<img src="perfectwave.gif" border="0"/>

<font face="Arial" color="#0000FF" size="5">
Finding the Wave
</font>
The following are the key elements to look for in a wave:

<ul>
<li/>Height
<li/>Speed
<li/>Shape
</ul>
```

HTML offers the ability to choose from a wide variety of font options, colors, and other layout and formatting elements. However, some non-standard HTML formatting elements may not be recognized in all browsers. The XHTML Strict recommendation does not allow the use of layout and style elements. This version requires the use of Cascading Style Sheets (CSS) to produce font, color and layout effects. XHTML Transitional includes all the HTML 4.0 features, such as style sheets, but does not rely on them; this variation of the standard allows you to use layout and style elements, as well as some deprecated elements.

For more information about X/HTML layout elements, see *Web Design Specialist*, Lesson 12: HTML and the Evolution of Markup.

Practice exam questions

Following are some practice exam questions intended for student review.

1. XHTML 1.0 is a reformulation of which standard?

a. XML 1.0
b. SGML 2.0
c. HTML 3.2
d. HTML 4.01

Correct response: d. HTML 4.01

Explanation: Extensible Hypertext Markup Language (XHTML) is the latest version of HTML. Essentially a reformulation of HTML 4.01 as an XML application (rather than SGML), XHTML was designed by the W3C to ease the transition from HTML to XML.

2. Which of the following is the correct hexadecimal value for the color green?

a. 0,255,0
b. 00FF00
c. FF00FF
d. 255,255,0

Correct response: b. 00FF00

Explanation: Hexadecimal code values range from 00 to FF (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Hex code correlates to the base-10 values but is stated in hex (base-16) values. The value 00 represents no presence of a color, and the value FF represents maximum presence of a color. Each value (red, green or blue) is assigned a number specifying the degree of that color. The hexadecimal value for the color green is stated as Red=00, Green=FF, Blue=00. Thus, the hexadecimal value for green is written as 00FF00, which represents no presence of red, maximum presence of green, and no presence of blue. RGB (Red, Green, Blue) values are formatted in base-10 numbers ranging from 0 to 255. The corresponding RGB value for the color green is written as 0,255,0.

3.2: X/HTML and extended technologies

This subdomain includes skills and knowledge required to use X/HTML and extended technologies to enhance Web page structure, format and usability.

3.2.1: X/HTML metadata tags and content

Statistics show that only about one of every four Web sites uses keyword and description <meta> tags. However, using these tags can give your site an advantage over sites that do not. Using <meta> tags enables more visitors to find your site initially. Unfortunately, not all search engines use metadata, and the ones that do often have different criteria for how it is used.

Consider the most commonly used metadata tags.

Metadata Tag Type	Description	Example
Keywords	Some search engines will search for values specified by the *content* attribute on pages that use the <meta> tag *name* attribute value of "keywords." Specified keywords are not displayed anywhere on the site pages; they are simply target keywords that a search engine can match to keywords entered by users. Keyword values are usually separated by commas. Do not use keywords excessively; several occurrences of the same word can cause the tag to be ignored.	<meta name="keywords" content="*keyword1, keyword2, keyword3*"/>
Description	When the <meta> tag *name* attribute specifies the "description" value, the *content* attribute value will appear as a short site description in some search engine results. A specified site description is not displayed anywhere on the site pages, but it is displayed in the results of search engines that recognize this tag. Keep this description brief: no more than 25 words (maximum allowance is 150 characters).	<meta name="description" content="*This site provides vegetarian recipes.*"/>
Robots	Certain pages should not be indexed by search engines, such as interior pages that should not be accessed without first visiting their introductory pages, or hidden pages that are posted only for exclusive users who have their URLs. When this is the case, the <meta> tag *robots* attribute value of "noindex" can be used to tell search engines not to index the page. Other *content* attribute values for "robots" include the following: -"nofollow" — prevents the crawler from following the links on the page and indexing the linked pages. -"noimageclick" — prevents links from pointing directly to images. Instead, only links to the page will be allowed. -"noimageindex" — prevents the images on the page from being indexed, but the text on the page can still be indexed.	<meta name="robots" content="noindex"/> <meta name="robots" content="nofollow"/> <meta name="robots" content="noimageclick"/> <meta name="robots" content="noimageindex"/>

Search results are ranked according to relevance to the given search criteria. Among searches and search engines, relevance varies. The following table lists a few common characteristics.

Web Page Element Used to Determine Relevance	Description
Titles	A search engine will first scan the X/HTML <title> tag to look for keywords that match the query.
Beginning content	A search engine will then look for the query keywords(s) in page text near the top of the document, on the assumption that the most relevant content is mentioned immediately.
Frequency	A search engine will look for frequency of the query keywords(s) in the entire document, on the assumption that the more times a word appears, the more likely the document is a good match to the search query.

Different search engines yield different results for the same search. Consider a few variables. Each search engine indexes sites to different levels, at different frequencies, then uses some proprietary steps to create relevance. For example, some search engines (such as Google) weigh factors such as the number of other sites that link to a given site, which may indicate the site's popularity. Some search engines do not use <meta> tags at all. And some search engines penalize the repetition of words by refusing to index a site. Some search engines do not catalog common words such as "the," "and," "Web" or "of" in an effort to conserve storage space.

Keywords contribute significantly to your site's ranking by search engines and directories. Regardless of the search engines or directories to which you submit your site, consider the following guidelines.

- Choose keywords that are relevant to your site, and do not use misleading words.
- If possible, choose a domain name that includes your keywords.
- Do not use X/HTML framesets. Most search engines have difficulty ranking sites that use them.
- When possible, use keywords when naming your X/HTML files (for example, cellular1.htm, cellular2.htm and so forth).
- Include combination keywords, such as "mobile phones" in addition to "mobile" and "phones." Studies indicate that as many as 60 percent of all searches use two or more words.
- Test your chosen keywords on some popular search engines to see the results that are returned. Then visit those sites, and look at the text and source code to see other keywords that were used.
- Check to see if the keywords you select are among commonly misspelled words. You can include commonly misspelled words in your <meta> tags to help users find your site, but do not misspell words on your pages.
- Avoid using keywords in your <meta> tags that search engines do not index, such as "to," "the," "and," "or," "of" and so forth.
- Use both singular and plural spellings of keywords, as well as other variations (for example, "phones," "phone," "telephone" and so forth).
- Use abbreviations (if appropriate) as well as both uppercase and lowercase spellings.
- Use the X/HTML <title> tag to include a brief 7-to-12-word description of your site, using as many as three of your keywords. Many search engines use the title as the site description, so make it appealing. The length should not exceed 70 characters nor use all-capital letters. Avoid using your company name (unless it is highly recognizable) because doing so wastes space for keywords.
- Create a logical descriptive summary paragraph using as many keywords as possible. Place it in the first section of the <body> of the document, within the first 200 characters. Remember that users will read this information, so it must make sense. Also, include the keywords in other areas of the document, including the last sentence.
- Use the *alt* attribute of the <img> tag to include keywords.
- Avoid using graphics exclusively on your home page. Search engines read text, not graphics.
- Do not try to mask a field of repeated keywords with background color.
- Do not place a banner ad at the top of your home page. Using the *alt* attribute and hyperlink this way describes keywords other than your own.

For more information about X/HTML metadata tags and content for search engine placement, see *Web Design Specialist*, Lesson 16: Site Content and Metadata.

3.2.2: Non-standard/proprietary X/HTML code

Raised in the WYSIWYG environment of the current word-processor generation, users wanted the ability to make type bold, change the font size or face, and add color when creating Web pages. Netscape, producer of the first widely successful browser, responded to users who were frustrated with HTML's limitations by creating extensions to the language. These extensions were tags that specified appearance formatting (presentational or procedural markup) rather than content structure (logical or structural markup). Microsoft followed, adding its own propriety language extensions.

Soon, page developers found themselves frustrated. The tantalizing extensions were not part of any standard, and were usually recognized only in the browser made by the company that created the extension. It takes a long time for a new markup element to be proposed, discussed and accepted into a new version of HTML. Netscape and Microsoft opted to present first and propose later.

If you create Web pages using standard HTML code, each page will appear essentially the same viewed in any browser. Most problems with display differences between browsers occur as a result of browsers incorrectly or incompletely implementing standards, or as a result of proprietary browser additions to the HTML specification. Remember that extensions might not be included in the latest specification, and some browsers might never support them. If you develop your Web site with proprietary extensions, you might be forced to revise your site to comply with the proposed standard and appeal to a wider audience.

Some browsers simply ignore proprietary tags, which has a negligible effect on how your HTML documents are displayed. Proprietary extensions can also be misinterpreted by a browser, which can have a destructive effect on your documents. HTML authors should be aware of these effects and test Web documents in several browsers.

For more information about non-standard/proprietary X/HTML code, see *Web Design Specialist*, Lesson 12: HTML and the Evolution of Markup.

3.2.3: Web page formatting with CSS1 and CSS2

Cascading Style Sheets (CSS) refers to the use of multiple style definitions in a single document. CSS is the standard way to format a Web page. A style sheet file can link to every document in a Web site, thus controlling the overall look and feel of the site. However, within any of the linked documents, a style header block can override the linked style sheet. Within the same file, a "spanned" style can also override the style information embedded in the header block, along with any style information from the linked style sheet. The term "cascading" refers to inheritance, or the hierarchical relationship between linked, imported, embedded and inline styles (i.e., the ability for styles that you apply to an element to be passed down to child elements).

X/HTML specifies the following four ways to apply CSS styles or style sheets to Web pages.

Method for Applying CSS to X/HTML Page(s)	Internal or External	Description / Example
Linking	Applies an external style sheet across multiple Web pages.	A single style sheet controls multiple Web pages. However, each page must be linked to the style sheet, which is a plain text file with the .css file name extension. To link a Web site file to the style sheet, use the following syntax in the X/HTML file: *<link rel="stylesheet" type="text/css" href="http://www.domain.com/styles.css"/>* Style sheet files should include only style information.
Importing	Applies external style sheets across multiple Web pages.	The import method allows you to apply multiple style sheets using one style element. The syntax for using the import method in an X/HTML file is as follows: *<style type="text/css">* *@import url("http://www.domain.com/styles.css");* *@import url("http://www.domain.com/morestyle.css");* *</style>* Note that in this method, a semicolon is needed at the end of the @import statement to separate it from any following style information.
Embedding	Applies internal styles to a single Web page.	The syntax for embedding a block of style information in a single X/HTML document is as follows: *<head>* *<style type="text/css">* *<!--* *h1 {color: blue;font-family: "Arial";font-size: 20pt}* *-->* *</style>* *</head>* *<body> </body>*
Inline	Applies internal styles to a single page; makes quick, temporary style changes to existing X/HTML code.	Inline styles are added inline to existing X/HTML tags or used in conjunction with the approved HTML 4.0 tag. The tag is a container tag that affects all content on the page between it and the closing tag. For example: * Habitat for Humanity * The advantage of using inline styles is that you can designate a set of attributes with a single tag. This method should be used only for occasional style changes. If you plan to use the same inline style more than once in a page, you should consider defining it as a style element in an embedded style header block so you do not waste time repeating inline instructions.

In a style sheet, you specify the elements to which styles will be applied by using the CSS selector types listed in the following table.

Selector Type	Description	Example
Tag	Defines the style effect that an element (such as <h1>) has on the text to which it is applied. Tag selectors are easy to use because once you redefine an element for a certain document, you simply use that element as you normally would and the new style is applied to it.	*p {font-family: Arial;}* This tag selector defines the style for all paragraph elements.
Class	Can be applied to any element. To define a class selector, you simply add a period (.) before the name of the selector, which you can specify to describe the purpose of the formatting. To apply a class selector to an element, you add the *class* attribute to the given element tag	*.important {color: red;}* This class is applied in the following statement: * Very Important Text *
ID	Can be applied to only one element in a document. To create an ID selector, you use the number symbol, or hash (#), before the name of the selector.	*#banner { padding: .5em;}* To apply the ID selector to an element, you add the *id* attribute to the element tag. For example: * This is the page banner *
Descendant	Selects an element only when it is the descendant of another specified element. The syntax for a descendant selector is simply a list of the selectors separated by white space.	*ul em {color: red;}* This style would be applied only to elements that are contained within an unordered list. Any other elements in the page would be unaffected.

All major browsers support the Cascading Style Sheets (CSS) 1.0 specification. CSS2 styles are applied in the same ways as CSS1 styles, and most browsers support CSS2, although some bugs still exist in certain implementations of this standard. The CSS2 Recommendation provides some additional features for:

- Media types.
- Paged media.
- Aural style sheets.
- Bi-directional text.
- Font support.
- Relative and absolute positioning.

The CSS3 specifications are currently being developed. They will be released as separate modules that will include new functions and features for borders, backgrounds, color, text effects, box resizing, generated content, multi-column layout and media queries.

For more information and hands-on labs about Web page formatting with CSS1 and CSS2, see *Web Design Specialist*, Lesson 15: Cascading Style Sheets. To read the CSS1 and CSS2 recommendations, visit *www.w3.org/TR/CSS1* and *www.w3.org/TR/CSS2/*. The W3C also offers a CSS code validation service at *http://jigsaw.w3.org/css-validator/*.

3.2.4: Linking style sheet to X/HTML page

The linking method uses external style sheets (text files that use the .css file name extension and contain nothing but style definitions) to apply styles across multiple Web pages). A linked style sheet allows you to make one change that affects multiple X/HTML elements. A style sheet file can link to every document in a Web site, thus controlling the overall look and feel of the site. Note that linked styles will be overridden by any embedded or inline style changes added to a single page.

With the linking method, a single style sheet controls multiple Web pages. However, each page must be linked to the style sheet. To link a Web page file to the style sheet, use the following XHTML syntax in the Web page:

```
<link href="mystyles.css" rel="stylesheet" type="text/css"/>
```

The style sheet file (mystyles.css) may include syntax such as the following to declare styles on any Web pages linked to mystyles.css:

```
body {
    font-family: Arial, Helvetica, sans-serif;
    background-color: #336699;
}
h1 {
    text-align: center;
    border: dotted #FFFFFF;
    font-style: italic;
    padding: 4px;
    background-color: #000000;
}
h2 {
    color: #006633;
}
p   {
    color: #FF9966;
}
```

The following table lists a few examples of style attributes that can be changed with style variation methods. A colon and the appropriate value follow each of these attributes. Multiple attribute/value pairs can be separated with a semicolon. All should be enclosed within curly braces {}.

Attribute	Sample Value	Description	Example
color	Any X/HTML color name or value.	Sets font color.	*color: red*
background	Any X/HTML color name or value; also accepts images.	Displays a background color or background image behind this HTML element. You can use any image types your browser recognizes. Both Firefox and Internet Explorer recognize GIF and JPG file types; Internet Explorer also recognizes the BMP format.	*background: yellow* *background: url(filename.gif)* *Note: When using a file for background, the url attribute must be followed with the file name in parentheses.*
font	Bold, 14 point, Verdana.	Sets the three important attributes of a font: the weight (bold, normal or light), the size (in points), and the font name. Not all attributes must be specified.	*font: 38pt Times New Roman* *font: light 18pt Arial*
font-family	Times, Palatino and so forth.	Specifies type. If multiple fonts are given, the browser uses the first in the list found on the end user's site.	*font-family: Garamond* *font-family: Arial, Sans Serif*
text-align	Center, left, right.	Specifies horizontal alignment of any given text element.	*text-align: center*

For more information and a hands-on lab about linking style sheets to X/HTML pages, see *Web Design Specialist*, Lesson 15: Cascading Style Sheets. To read the CSS1 recommendation, visit *www.w3.org/TR/REC-CSS1/*. The W3C also offers a CSS code validation service at *http://jigsaw.w3.org/css-validator/*.

3.2.5: X/HTML tables

The W3C now discourages the use of tables for page layout in favor of positioning with Cascading Style Sheets (CSS). Professionals use CSS and layering. Until recently, designing page layout using tables has been the only way to create complex layouts. With nearly universal browser support for CSS, however, it is now possible to use CSS positioning for most page layout. The reason for the W3C's recommendation against using tables for page structure is that using table layouts reduces a site's usability for people with disabilities and others browsing the site on mobile or other non-standard browsers.

When you want your Web page to display a group of data that is best suited to a tabular format, the set of X/HTML table tags create useful and attractive grids for Web page content. Tables provide formatting superior to the HTML <pre> tag, which allows you to display preformatted text that includes structure such as tabs and spacing. However, the <pre> tag does not allow you to format data exactly as you want it, whereas the <table> tag gives you many formatting options.

When content works well in a tabular format, a table improves readability and adds visual interest. You can use tables to organize existing content on your Web pages simply by adding the appropriate XHTML tag structure. Then you can supply a couple sentences to introduce the table, as well as a table caption with a title for the table.

The easiest method for creating a table in X/HTML is to begin by diagramming it, specifying the basic structure of the page, columns and rows, and any spanned columns or rows. Tables can be presented with invisible borders to hide the table formatting, and allow the content in the table cells to stand out and visually occupy the space on the page independently.

Knowledge of X/HTML is a prerequisite for CIW Web Design Specialist. You should be able to create a basic X/HTML table using a text editor such as Notepad. CIW Site Development Associate courseware provides detailed instruction in XHTML code. The following table briefly describes the main X/HTML tags used for creating basic tables.

Basic Table Creation Tag	X/HTML Element	Description
Table tag	<table>	The <table> tag must be modified to allow your table to occupy the entire space of a Web page. Adjusting the *border*, *height*, *width* and *cellspacing* attributes will accomplish this goal.
Table row tag	<tr>	Modification of the <tr> tag is not required to create a borderless page structure. However, it is still useful when desired to provide uniform attributes, such as *align*, *bgcolor* or *background*, to all cells in a row.
Table data tag	<td>	The <td> tag can to be modified by changing the *background*, *bgcolor*, *height*, *width* and *align* attributes. Empty narrow cells can be used to create visual separations between rows and cells.

For more information and hands-on labs about X/HTML tables, see *Web Design Specialist*, Lesson 14: Web Page Structure – Tables and Framesets.

3.2.6: X/HTML framesets

Most designers no longer use frames. Nevertheless, when designing your Web site, you may want to create a page structure in which certain information (such as navigation links, copyright notices and title graphics) is visible and persistent while other information changes. As users navigate the site, the static

frame's content will remain fixed, while the contents of the adjoining frames may change or can scroll. You can produce this functionality by creating separate panes in the browser window called frames and combining them in a structure called a frameset. The advantage of using frames is that static and dynamic information can be combined on a page.

Frames are an extension of the HTML 3.2 standard introduced by the Netscape release of the Navigator 2.0 browser, and expanded by the Microsoft Internet Explorer 3.0 browser. Frames were submitted to the Internet Engineering Task Force (IETF) and the W3C for consideration as an HTML standard, and are part of the HTML 4.0 / XHTML 1.0 Recommendations.

Knowledge of X/HTML is a prerequisite for CIW Web Design Specialist. You should be able to create and link basic X/HTML frames using a text editor such as Notepad. CIW Site Development Associate courseware provides detailed instruction in XHTML code. The following table describes the basic tags used when creating X/HTML framesets.

Frameset Tag	Description	Notes / Examples
<frameset>	A container tag that allows you to define regions in your browser window and assign separate files to each region. Typically replaces the <body> tag in an X/HTML frameset document because no text other than the frame pages' content generally appears on the frameset page.	Requires an attribute of either *cols* or *rows* that designates the number and size of columns or rows in a browser window. You can specify the *cols* and *rows* attributes in two ways: by percentages or by pixels. For example, the tag <frameset cols="35%,65%"> indicates two frames: One column occupies 35 percent of the available screen width, and the other column occupies 65 percent. Frame sizes expressed in percentages are referred to as relative sizing. Expressing the size of frames in pixels is called absolute sizing because the size of each frame remains constant regardless of browser window size. In addition, you can use a wildcard character for the second frame. For example, by specifying <frameset rows="150,*"> you use the first 150 pixels of available screen space for the top frame and the remaining space for the bottom frame.
<frame>	Defines the content that will appear in each frame of a frameset. It is enclosed within the <frameset> tags.	The *src* attribute in <frame> specifies the file that will appear in the frame. In the following example, the content source for the top frame is the file named top.htm, and the content source for the lower frame is named bottom.htm. The frameset defines two rows and opens the frames with the files specified in the <frame> tag: *<frameset rows="100,*">* *<frame src="top.htm"/>* *<frame src="bottom.htm"/>* *</frameset>* The frame's content source can be a local document or a URL pointing to a Web page.
<a>	Used to create hyperlinks for targeting frames.	You can designate internal links without specific file addresses by using the <a name="anyName"> tag. After you specify a name ("anyName") to which you can anchor your link, you can create the <a href="anyName"> tag and attribute combination. The *name* attribute can also be used to target hyperlinks from the navigation division of your frame to the main area of your browser window. When you name a frame, you can request that frame by name to load files into it. Target names are case-sensitive.

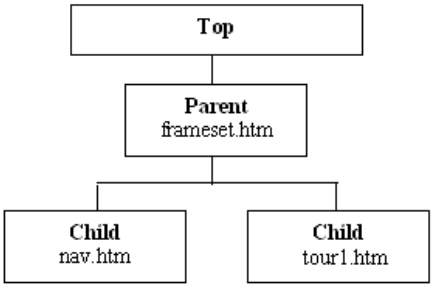
The <frameset> tag will create frames only if it is placed correctly into the X/HTML document. The following example demonstrates proper structure for creating frames.

```
    <?xml version="1.0" encoding="UTF-8"?>
<DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
        "DTD/xhtml1-frameset.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml" XML:LANG="en" lang="en">
    <head>
    <title>Simple Frameset</title>
    </head>

    <frameset cols="30%,70%">
        <frame src="nav.htm"/>
        <frame src="tour1.htm"/>
    </frameset>
    </html>
```

Following are three key points to observe:

- The document type declaration names the DTD for the Frameset variant of XHTML. You only need to use the Frameset DOCTYPE declaration for the Web page that creates the frameset. The documents called by the <frame> tags should use either the Transitional or Strict variant of XHTML.
- The opening <frameset> tag must follow the closing </head> tag and must precede the opening <body> tag. If you do not plan to use the <noframes> tag to display alternate text for browsers incapable of rendering frames, you do not need the <body> tag at all.
- The <frameset> tag must contain the *rows* attribute and/or the *cols* attribute. Although both can be specified in the same <frameset> tag, it is advisable to declare them in separate <frameset> tags because output results may vary among browsers.

You can target frames in two ways: by name or by relationship. Recall that the target name is defined in the <frame> tag. The frame relationship, however, is less tangible. The following figure shows an example of relationships among files and frame names.



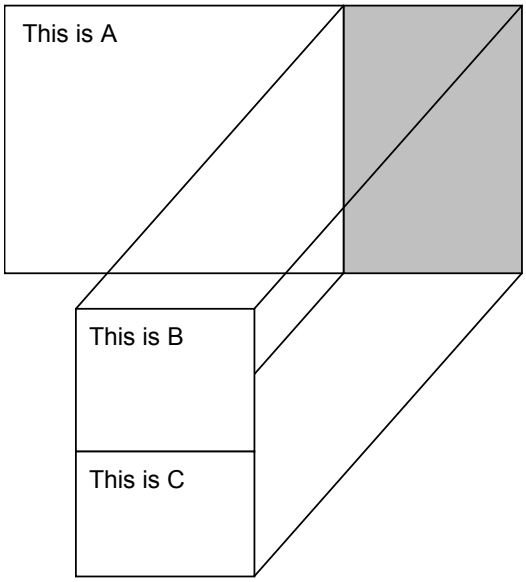
The following code shows an example of targeting a link to the top frame:

```
    <a href="http://www.habitat.org" target="_top">HOME</a>
```

You can create a nested frameset by combining columns and rows structuring. For example, you can add to your frameset page an additional pane that can display a banner ad consisting of static content or alternative linked content. You can combine rows and columns in your frameset to achieve this effect. The following code shows an example of nesting another frameset within an existing frameset:

```
    <frameset cols="150,*">
       <frame src="nav.htm" name="nav"/>
    <frameset rows="30%,70%">
       <frame src="header.htm" />
       <frame src="tour1.htm" name="tour"/>
    </frameset>
    </frameset>
```


Be sure that you understand the difference between a nested frameset and combined framesets. The nested frameset places a frameset within one frame in an existing frameset. The combined framesets allow you to place multiple framesets together, so each page functions as an individual frameset. Consider the combined frameset shown in the following figure. This frameset actually combines two framesets: one divided into two columns, and the other into two rows. The figure illustrates this frameset as a three-dimensional model. This combined structure allows one link in Pane A in the diagram to change the content in both Panes B and C.



The following code shows an example of a combined frameset:

```
        <frame src="header.htm" name="top"/>
        <frame src="tour1.htm" name="bottom"/>
    </frameset>
```

For more information and hands-on labs about X/HTML framesets, see *Web Design Specialist*, Lesson 14: Web Page Structure – Tables and Framesets.

Practice exam questions

Following are some practice exam questions intended for student review.

1. When you apply CSS style variations to Web pages using the linking method, where do you place the style information?

a. In the <style> tag
b. In the tag
c. In a separate text file
d. In a separate XHTML file

Correct response: c. In a separate text file

Explanation: With the linking method, a single style sheet controls multiple Web pages. However, each page must be linked to the style sheet, which is a plain text file with the .css file name extension. You use the <link> tag in your XHTML file to refer to the style sheet.

2. You want to create a dynamic structure for your Web page that changes with the browser window size (known as relative sizing). When specifying the *height* and *width* attributes in the XHTML table or frameset structure tags or in a style sheet, which unit of measure should you use for the attribute values?

a. Pixels
b. Inches
c. Kilobytes
d. Percentages

Correct response: d. Percentages

Explanation: When specifying the height and width attributes in table and frameset structure tags, the values can be expressed in pixels or percentages. If you use percentage values (%) for the height and width attributes, the structure becomes dynamic, and will change as browser window size changes. This technique is called relative sizing. In absolute sizing, you specify these attributes in pixels, and the structure size is the same regardless of browser window size.

3.3: Image files and X/HTML

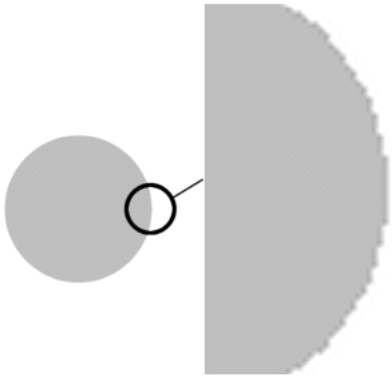
This subdomain includes skills and knowledge required to create image files, and use images in X/HTML pages and site design.

3.3.1: Vector vs. raster graphics

Web images can be categorized into two graphic formats: raster (or bitmap) and vector. Each format has different file subtypes.

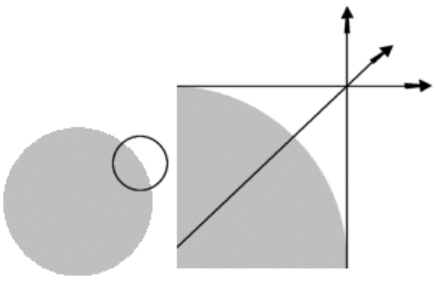
The most common graphic format is raster, also known as bitmap. Raster graphics use small dots to create images and colors. Raster graphics include the JPEG, GIF and PNG formats. Raster graphics are best suited for photographs and realistic graphics.

The larger the dimensions of the image, the larger the associated file size. When viewed with magnification, a raster graphic resembles a series of little squares, each of which has a color value that contributes to the overall shape. The following figure shows a raster image as it looks when magnified.



For a raster graphic to render its shape, the browser and the available RAM work in unison to load the graphic. The information for each pixel is stored, and then rendered in the browser to create the image. The reason that larger raster images result in larger file sizes is that more pixels must be instructed on the values to display. Because raster graphics are created using actual dots of color, making a raster graphic larger decreases the image resolution. Removing pixels and compressing files will decrease file size, but will also reduce image quality.

Vector graphics are best suited for line art, shapes and illustrations. Vector graphics are quite different from raster graphics in their rendering process. Vector graphics store the information about the image in mathematical instructions that are then interpreted and displayed. For example, to create a circle, the vector graphic need only know the center coordinates, radius and color values. With this information, mathematical instructions are used to render the graphic. To increase the size of the circle, only the information values need to change, therefore increasing the size of a vector does not increase the file size of the image. The following figure shows a sample graphic and illustrates how the graphic uses lines to interpret and render the shape.



Another advantage of vector graphics is their ability to scale to the output device being used. Vector has this ability because it is not resolution-dependent.

For more information about vector vs. raster graphics, see *Web Design Specialist*, Lesson 9: Web Graphics.

3.3.2: Image file formats

Graphic images come in numerous file formats. Only GIF and JPEG are natively supported by all browsers; other file formats can be viewed with the aid of a plug-in. Native support means that the browser does not require any special software or plug-in to display the image. When creating Web graphics, designers must choose between GIF and JPEG based on the type of images they want.

The following table lists various image file formats and their file name extensions.

File Name Extension	File Format	Origin
.jpg, jpe, .jpeg	Joint Photographic Experts Group (JPEG)	Joint Photographic Experts Group
.gif	Graphics Interchange Format (GIF)	CompuServe Inc.
.tif	Tagged Image File Format (TIFF)	Aldus Corporation
.bmp	Bitmap (BMP)	Microsoft Corporation
.wpg	WordPerfect Graphic	WordPerfect Corporation (Novell)
.png	Portable Network Graphics	Codelab Inc.
.pcx	Bitmap (BMP)	ZSoft Corporation

The following table describes the major Web-ready image file formats.

Image File Format	Description
Graphics Interchange Format (GIF)	A platform-independent file format that is limited to a display of 256 colors. Adopted by most developers because of its small file size. Graphic interlacing (the progressive rendering of images) is unique to GIFs and is a preferred method for display of large graphic files. GIF is considered a "lossless" format, meaning that as the image is compressed, no information is lost. Consequently, a GIF file may not compress as much as a JPEG file of the same size. GIF files exist in two versions: 87a and 89a.
GIF 87a	The original 87a format supports most of the GIF's advantageous features, such as small file size, lossless compression and transparency.
GIF 89a	The 89a version of GIF also supports animation and interlacing. Animation is the storage and playback of a sequence of still images to create the illusion of animation. GIF files can provide many full animation and video effects without the need for plug-ins. Interlacing is the progressive rendering of images as they are downloading, which makes download time seem shorter.

Table cont'd

Image File Format	Description
Joint Photographic Experts Group (JPEG)	Capable of much greater color depth than GIFs, but usually require more time to download. JPEG files can contain up to 24 bits of color information (16.7 million colors) and work particularly well for photographs. JPEG files are compressed automatically, and decompressed when they arrive at the Web page. Compression effects vary: The greater the compression, the greater the level of degradation in the final image, thus JPEG is considered a "lossy" format type. One drawback to this format is that the designer has no control over how the 24 bits are mapped into the 256-color palette used by a client's display. Also, considerable differences exist between the way Netscape and Internet Explorer (and PCs and Macintoshes) display images. Designers should test their images in multiple browsers and multiple operating systems before deciding which format best serves their purposes.
JPEG 2000	A newer JPEG format enhances the compression feature of standard JPEGs and improves scalability and editability. With JPEG 2000, the compression includes a Wavelet technology that stores the information differently, reducing the amount of speckling that occurs when the JPEG is decompressed. The file name extension for JPEG 2000 is .jp2.
Portable Network Graphics (PNG)	The PNG file format is emerging as the new format for Web graphics. PNG files are lossless and support transparency like GIFs, yet also support compression and high bit depth like JPEGs. In addition, PNG bit depth can be adjusted, unlike GIFs or JPEGs, which must be 8-bit and 24-bit depth, respectively. Compression is enhanced in PNGs by using compression filters that can support up to 48-bit color. However, browser support for the PNG format is currently not complete. As a result, it is not advisable to rely on PNG as a fully supported format for the Web.

For more information about image file formats, see *Web Design Specialist*, Lesson 9: Web Graphics.

3.3.3: Image-editing software

You can create, retouch and export Web-ready digital images with a variety of graphics programs. These programs can be divided into two general groups: vector-based drawing programs and paint-type programs. The following table describes these two types of image-editing software.

Image-Editing Application Type	Description	Examples
Vector-based drawing programs	These programs map shapes that you create onto an invisible grid. Image information is stored as a set of mathematical instructions. Vector graphics are resolution-independent, meaning that computer monitor settings do not affect their size or appearance. Vector files are also smaller than similar paint-type image files. Currently, vector graphics cannot deliver truly photo-realistic detail, and they display somewhat more slowly on screens. They also require a browser plug-in or helper application for viewing on the Web. However, vector graphics work very well for industrial, manufacturing, scientific and educational purposes.	If you want to develop your own graphics, vector-based drawing applications commonly used in the industry include the following: - Adobe Illustrator - Adobe Freehand - CorelDRAW - Inkscape - Adobe Fireworks
Paint programs	These programs create raster images from scanned photos and video frame captures, or can be used to create original artwork. Raster files can represent highly detailed images. A raster (also called bitmap) graphic is an arrangement of small dots, and each dot corresponds to a pixel on the screen. Raster images can be edited pixel-by-pixel if necessary. One drawback of rasters, however, is that the original files are large. Both JPEG and GIF formats are raster graphic types.	If you want to develop your own graphics, paint-type applications commonly used in the industry include the following: - Adobe PhotoShop - Microsoft Image Composer - Corel PaintShop Photo Pro - Adobe Fireworks *Note that Fireworks supports both vector and raster graphics.*

For more information about image-editing software, see *Web Design Specialist*, Lesson 9: Web Graphics, and Lesson 24: Image Editing with Adobe Fireworks CS5.

3.3.4: Image manipulation functions

Image-editing applications are generally either vector-based or raster-based, although some (such as Adobe Fireworks) may support both capabilities. Image-editing applications generally offer a user interface similar to other application interfaces by the same vendor, which can reduce learning curves for some tools. Image-editing applications generally offer the same types of functionality and features. Many create native files in JPEG, GIF or PNG format, although they can generally export to most other image file formats as necessary. The following table describes some of the common image-manipulation functions of image-editing applications.

Image-Manipulation Function	Description
Creating new image document	Creates a new image document. Generally offers image properties that can be modified, such as canvas size; image width and height in pixels, inches or centimeters; color of the image canvas and background color; file format; and so forth.
Adding text to image	Adds any text you type to an image document, and allows you to format the text and apply additional effects to it.
Cropping	Trims excess background space from around the image. You generally draw an outline around the image, then double-click the selection to crop the image to the new specified dimensions. Most tools are limited to rectangular and square outline shapes for cropping. Cropping images is an important part of image creation because removing the excess white space (image canvas) from an image decreases its dimensions and file size.
Selecting image areas	Many programs offer additional tools to select images that support circular, oval and freehand outline shapes. These tools allow you to choose from multiple shapes to select areas of an image.
Image slicing	Divides large images into smaller images for use on Web pages. When the sliced image loads in a browser, it is a series of individual images that are constructed within an X/HTML table to make them appear as one image. Slicing large images decreases the perceived download time, because smaller images download faster than large ones. Although the combined size of the smaller, divided files is not any less than the size of the single undivided file, users will usually see image parts on the screen faster if the image is divided than if it is kept as a single image.
Modifying image size, resolution and bit depth	Image-editing programs allow you to modify properties of existing images such as their size, resolution and bit depth. This capability allows you to adjust the appearance of images relative to their functions and adjacent content, and optimize their quality for best viewing in all browsers. When you reduce resolution, print size increases. This is because the number of pixels per inch decreases while the number of pixels stays the same, thus necessitating more inches to print the image. Changing the bit depth of an image is one way to reduce its file size, but it also reduces image quality. Because photos typically have a large number of colors, reducing the bit depth is not a good way to reduce the file size of photo images.
Rasterizing	An image-editing program can convert the vector graphics that you create into bitmap, or raster, graphics. This process is called rasterizing.

For more information and hands-on labs about image manipulation functions, see *Web Design Specialist*, Lesson 24: Image Editing with Adobe Fireworks CS5.

3.3.5: Transparent and animated images

The 89a version of GIF allows storage and playback of a sequence of still images to create the illusion of animation. GIF files can provide many full animation and video effects without the need for plug-ins.

Another advantage that a GIF image has over a JPEG image is that the designer can designate a color of the GIF image to be transparent. PNG files support transparency like GIFs, yet also support compression and high bit depth like JPEGs. However, browser support for the PNG format is currently not complete. Older versions of the major browsers offer little support for PNG. Internet Explorer version 6 had incomplete support for transparent PNGs, and Internet Explorer 7 and 8 are unable to correctly display PNG images with color correction.

The following table describes animated and transparent images in more detail.

Image Type	Description	Creation
Animation (image states in Fireworks)	The animated GIF functions much like a cartoon flipbook. Animated GIF files consist of sequential frames, or states, that reload from a browser's cache and replay in an infinite or predetermined loop to simulate motion. One advantage that animated GIFs have over other image animation is that they do not depend on client-pull or server-push. Both server-push and client-pull require a file to be downloaded sequentially with numerous exchanges between the user's computer and the server.	Most image-editing programs allow you to compile a series of GIF images into a single animated GIF. The image states feature is unique to Fireworks and differs from the way other graphics applications handle animation. Using Fireworks, you can create an animated image from a single file using states, then export the states to an animated GIF. Each state acts as an independent image of the entire animation, thus allowing the same effect as using individual images compiled into one animated GIF. Think of states as individual snapshots, like those that appear on a strip of film to create a movie. When you combine a series of still shots in rapid succession, animation occurs.
Transparent images	Transparent images are useful when the background of a page needs to be displayed behind the image. Image transparency is also used to give the appearance that an image is floating, or is not constrained to a rectangular or square shape. For example, the designer can create a circular logo in a square image by making the background color transparent. Thus the image appears circular when, in fact, it is square with information for the square background to appear transparent.	Most image-editing programs allow you to create transparent GIF image files. To create a transparent image, you must essentially remove the background color from the image file. When you create transparency, you can remove only one color. Therefore, the image with which you are working must have a consistent background color that is not an image color you want to keep.

For more information and hands-on labs about transparent and animated images, see *Web Design Specialist*, Lesson 9: Web Graphics, and Lesson 24: Image Editing with Adobe Fireworks CS5.

3.3.6: Image layers

Some image-editing applications offer the ability to create image layers. When you design a graphic, each individual component of the image can be created on its own layer, thus allowing that component to be manipulated independently of the entire image. A series of layers will compose an entire image, and an image can have as many layers as necessary.

Although layers are supported in the PNG file format, they are not supported by GIF or JPG formats. This means that when the image is finalized, it must be flattened into a single layer to be exported to those formats, and then the layers can no longer be manipulated. If the designer maintains a copy of the layered PNG version of the image, he or she can make changes easily, and other versions of the image can be exported when necessary.

For more information and hands-on labs about image layers, see *Web Design Specialist*, Lesson 23: Creating Web Pages Using Open-Source Tools, and Lesson 24: Image Editing with Adobe Fireworks CS5.

3.3.7: Image colors and audience cultures

Because colors and images convey information about the organization that created a Web site, you must take care to evaluate the effectiveness or appropriateness of the colors you choose based on your intended audience.

Certain color schemes — such as the colors used in a specific country's flag — may suggest geographic or cultural biases. Images portrayed in one color scheme may convey a different message in another (consider again a country's flag colors, or a puddle of liquid shown in blue compared to the same puddle shown in red).

As you consider your audience, make sure that you identify the following:

- Color combinations that might be attractive or acceptable to specific audiences or cultures
- Color combinations that might be unattractive to specific audiences or cultures
- Symbols, objects or images that may attract or repel an audience

Remember that colors and images that attract one audience might offend another. Also, be ready to consider and adopt different perspectives. Otherwise, your design efforts will be less successful.

For more information about image colors and audience cultures, see *Web Design Specialist*, Lesson 5: Web Page Layout and Elements.

3.3.8: Image files in X/HTML

Knowledge of X/HTML is a prerequisite for CIW Web Design Specialist. You should be able to insert an image into a Web page with basic X/HTML using a text editor such as Notepad. CIW Site Development Associate courseware provides detailed instruction in XHTML code.

The <img> tag is used to include an image in the Web page, and the name of the image file is specified as a value of the required *src* attribute within the <img> tag. The <img> tag is an empty tag, so in XHTML you must include the forward slash character (/) at the end of the tag.

XHTML also requires the addition of the *alt* attribute, which specifies a text description to appear in place of the image in some browsers, to make your page accessible to users whose browsers cannot render images. The correct XHTML syntax to insert an image into a Web page is as follows:

**

A GUI HTML-editing application can insert images into your Web pages for you by writing this code automatically. Generally, you place your cursor on the page in the location you want the image, or between the <body> tags in the code, and select the Image tool from the toolbar.

Understand also that you will want to create a logical repository structure for your images. Otherwise, you may not be able to find the images in the future, and you also may discover that the links you created to them will no longer be valid. For example, you may choose to create a subdirectory named "images." You can then create an organizational structure for all images off of this subdirectory. Organization is central to successful Web design.

For more information and a hands-on lab about image files in X/HTML, see *Web Design Specialist*, Lesson 23: Creating Web Pages Using Open-Source Tools.

3.3.9: Metadata in images

Metadata is data about other data. Metadata for photographs typically includes the date and time a picture was taken, and the details of the camera settings. Many digital cameras record metadata in exchangeable image file format (EXIF).

A key feature of photosharing sites is the ability to tag items in order to categorize them. A tag is a non-hierarchical keyword or term that you can assign to an item such as a Web page or an image file. Tagging or "image tagging" an item allows it to be found again by browsing or searching; that is, a tag is metadata.

Many photosharing sites allow users to tag their images, making the pictures highly searchable. Photosharing Web sites allow you to upload, organize, view, share and download photos and other image files. Using metadata in your images (along with properly naming your Web pages and using keywords in your pages) can help you achieve higher rankings in searches.

For more information and a hands-on lab about inserting metadata into images, see *Web Design Specialist*, Lesson 9: Web Graphics.

3.3.10: Stock photography in site development

Stock photography refers to images for which you can purchase the rights to use in printed material or on a Web site. Photographers file their images with an agency that negotiates licensing fees on the photographer's behalf. Stock images are then licensed from the agency.

Many modern stock photography distributors offer still photos, videos and illustrations. Millions of photos are available for licensing, and all images on stock photography Web sites include embedded metadata that makes it easy to search for images based on keywords.

To use a stock image, you must purchase a license for the image. The license is a one-time fee that allows you to use the image multiple times for multiple purposes. The following table describes the two types of licensing that are available.

License Type	Description
Royalty-free license	Allows the buyer to use an image without having to pay a royalty each time the image is used. Royalty-free does not imply that the image is free to use without purchasing a license or that the image is in the public domain. The buyer also does not have the right to resell or transfer the image. The fee for the image is usually based on the size of the digital file.
Rights-managed license	Allows the buyer to "rent" an image through negotiation of a specific price for a specific use. Some rights-managed licenses stipulate exclusivity; that is, the buyer may restrict similar use of the image by others for the duration of license. Rights-managed licenses are usually more expensive than royalty-free licenses, but they allow for much larger print runs

Advantages to using stock photography when developing a Web site include saving time and money:

- Image databases are quick and easy to search.
- Licenses can be purchased online.
- Images are downloaded immediately for use.
- Licenses (especially royalty-free licenses) are much less expensive than hiring a photographer and models for a photo shoot, in most cases.
- Using stock photos increases project speed.

Another advantage to using stock photography is that you know exactly what the finished image looks like, while an assignment photo shoot may deliver photos that require editing.

Disadvantages to using stock photography are (in some cases):

- The cost associated with the licensing fees.
- A reduction in creative control. You have to select from the images that are available, and you may not be able to find exactly what you want.

For more information about using stock photography when developing a site, see *Web Design Specialist*, Lesson 9: Web Graphics.

Practice exam questions

Following are some practice exam questions intended for student review.

1. Which commonly used digital image file format is natively supported on the Web and offers small file size, supports transparency, and is considered a "lossless" format?

a. Scalable Vector Graphics (SVG)
b. Portable Network Graphics (PNG)
c. Graphics Interchange Format (GIF)
d. Joint Photographic Experts Group (JPEG)

Correct response: c. Graphics Interchange Format (GIF)

Explanation: Graphics Interchange Format (GIF) is considered a "lossless" format; this term means that as the image is compressed, no information is lost. Consequently, a GIF file may not compress as much as a JPEG file of the same size. The amount of compression cannot be determined by the designer for a GIF file, though it can for a JPEG file. GIF files exist in two versions: 87a and 89a. The original 87a format supports most of the GIF's advantageous features, such as small file size, lossless compression and transparency. The 89a version of GIF also supports animation and interlacing. PNG is not yet widely supported on the Web.

2. Which type of graphic stores the information about the image in mathematical instructions that are then interpreted and displayed?

a. JPEG
b. Vector
c. Raster
d. Bitmap

Correct response: b. Vector

Explanation: Vector graphics store the information about the image in mathematical instructions that are then interpreted and displayed. By contrast, raster (or bitmap) graphics use small dots to create images and colors. Vector graphics are best suited for line art, shapes and illustrations.

3.4: GUI site development applications

This subdomain includes skills and knowledge required to create Web sites using GUI site development applications.

3.4.1: X/HTML text editors vs. GUI site management applications

A text editor, such as Notepad or jEdit, requires you to know and type all X/HTML code for your Web pages. Although this task can be time-consuming, it gives you full control over your code and allows you to code to the most recent standards.

Graphical user interface (GUI) site-development tools use a display format in which the file being edited appears on the screen just as it will appear to the end user. These applications are sometimes also called WYSIWYG (pronounced "whiz-ee-wig") tools, which is an acronym for What You See Is What You Get. The developer can simply point and click editing functions to create Web pages, and the application writes the HTML code automatically. These applications enable advanced Web layout and design without the need to know HTML code. Examples of GUI site-development applications include Microsoft Expression Web and Adobe Dreamweaver, and open-source tools such as KompoZer and OpenWebSuite.

With today's sophisticated Web projects, Web management has become as important as design. Many GUI site-development applications incorporate the following functions within the application to assist in the overall development process:

- **Page layout and design** — The developer can choose among tables, framesets or CSS positioning for page layout, as well as HTML code preservation, advanced DHTML, multimedia and scripting technologies.
- **Site management** — Management of large Web applications is integrated through the use of the various views, preferences, file sharing and reporting capabilities.
- **Data connection** — Generating content with dynamic data has become an integral part of large Web applications. Creating, adding, retrieving and deleting data are possible for the non-programmer using the data connection features.

GUI site-development applications can dramatically reduce the time required to manage and develop a Web site. Developers can save time by using the WYSIWYG interface for HTML design, and the similarities to other vendor application products reduce product learning curves. However, most GUI applications have not kept pace with current XHTML standards, and may include deprecated or proprietary tags in the code they write. Also, some GUI applications may require you to host your site on a Web server that offers support extensions for the application.

Today’s GUI Web page-editing tools are good enough that the majority of Web designers see them as useful, and even necessary, for Web development. When using tools to automate development tasks, Web developers still need to understand the underlying technologies (including HTML), but developers should still use the tools available to help them do their jobs efficiently.

For more information about X/HTML text editors vs. GUI site management applications, see *Web Design Specialist*, Lesson 17: Site Development with Microsoft Expression Web 3 – Introduction, and Lesson 23: Creating Web Pages Using Open-Source Tools.

3.4.2: W3C-compliant code with GUI site applications

You can develop entire Web sites in GUI site applications without ever looking at the HTML code in the pages. However, many long-time Web developers are skeptical of generated code — and rightly so. Older versions of these products were notorious for creating convoluted and difficult-to-understand code with many proprietary tags.

Microsoft Expression Web and Adobe Dreamweaver can be configured for compliance with current W3C code standards. For Web designers who are not particularly adept at writing X/HTML code, using a GUI development tool such as Expression Web or Dreamweaver is a good way to create Web pages that comply with W3C standards and thus promote accessibility, even without a detailed knowledge of the standards' specifications.

Expression Web generates code that is compliant with the XHTML 1.0 Transitional standard by default, and Web designers may also specify various standards and CSS schemas on the Authoring tab of the Page Editor Options dialog box. Expression Web also includes tools such as the Compatibility Checker and the Accessibility Checker that you can use to validate your code to current W3C standards.

Dreamweaver allows you to select XHTML 1.0 Transitional from the DocType drop-down list when you create a new page. This selection makes all code generated for the page comply with the XHTML 1.0 Transitional specification. Dreamweaver also includes a markup validation feature that examines page code and indicates errors in a results pane for your review.

In KompoZer, an open-source GUI editor, you can modify your default template to make it an XHTML 1.0 Transitional document by adjusting the markup selections in the Advanced Settings window.

Because you do not have full control over the code that a GUI site application generates, it is not possible to make code created with it completely XHTML-compliant. However, you can validate the code using resources on the Web such as the W3C Markup Validation Service (*http://validator.w3.org/)* and review the errors to learn about the types of non-standard code that the application includes in the files it creates, then manually correct the code in the document.

For more information and hands-on labs about W3C-compliant code with GUI site applications, see *Web Design Specialist*, Lesson 17: Site Development with Microsoft Expression Web 3 – Introduction, Lesson 20: Site Development with Adobe Dreamweaver CS5 — Introduction, Lesson 22: Site Development with Dreamweaver CS5 — Advanced Features, and Lesson 23: Creating Web Pages Using Open-Source Tools.

3.4.3: Images with GUI site applications

GUI Web site applications such as Microsoft Expression Web and Adobe Dreamweaver make image insertion easy by using point-and-click for placement, offering simple menus to locate image files, and copying image files to the correct folder for your site structure. Generally, you can insert images that are located on the Internet or on your computer, or you can insert images already used in the Web site.

To insert an image in a Web page, place your cursor in the position on the page where you want the image to appear, then click the appropriate tool (e.g., Insert Picture button) on the toolbar. A dialog box usually appears so you can locate and choose the image to insert. Both Expression Web and Dreamweaver offer dialog boxes that allow you to specify alternate text for each image easily. These applications also allow you to use layers or divisions to arrange images on a page. Unlike a table structure that organizes the images for you, layers allow you to move the images manually and place them where you want them — you simply need to determine the most logical and aesthetic placement for each image, based on its function and purpose in the Web page.

An image map allows you to specify one or more portions of an image to act as hyperlinks. GUI site applications offer point-and-click functionality for creating hotspots, so you need not determine coordinates and dimensions manually. Consider the amount of code you must write to target hotspot links manually in X/HTML. Easy creation of image maps means you can target precise areas of images as hyperlinks when appropriate, instead of linking an entire image to save coding time. GUI site applications generally allow you to choose from three shapes (rectangle, circle or polygon) to define image map areas on the selected image. You create the hotspot areas by selecting tools and drawing on the image with your cursor.

Remember that any images used on a Web page must be accessible by the Web server to appear on the page. GUI site applications generally copy image files to the correct folder for your site structure.

For more information and hands-on labs about images with GUI site applications, see *Web Design Specialist*, Lesson 18: Site Development with Expression Web 3 – Basic Features, and Lesson 21: Site Development with Dreamweaver CS5 — Basic Features.

3.4.4: Page text, tables and hyperlinks with GUI applications

Most GUI site applications (such as Microsoft Expression Web and Adobe Dreamweaver) allow you to add content to Web pages in several ways, including manually typing it in. You can also import or paste content such as X/HTML, Rich Text Format (RTF) and Microsoft Word documents, with or without formatting, directly into a Web page in the page view. When creating or updating Web content, you will often use content that is already in electronic format, perhaps formatted to look as the customer wants it. Rather than take the time to manually enter content, correct any errors you make in entry, and format the content to match your customer's preferences, you can simply import or paste the electronic content exactly as it was provided. This saves you development time, enables more efficient team collaboration, and gives your customers exactly what they want.

To develop effective tables in a GUI site application, you should know the approximate table configuration you want prior to construction. Some determinations must be made with respect to the size and number of tables to be used. You will probably use multiple tables for advanced page layout or formatting of complex data. Multiple tables can be independent of each other, or they can be nested within one another. You do not need to know the exact table dimensions prior to construction because GUI editors allow you to visually modify tables to the desired sizes.

Most GUI site applications allow you to either draw a table or insert one of several predefined tables. The draw table option allows you to specify the table structure you want by drawing it manually with your cursor. An eraser tool is generally provided to erase parts of the table you draw. You can insert a table wherever on the page the cursor is placed by selecting a menu, clicking and dragging, or entering dimensions to specify the table size. The table is inserted when you release the mouse button or click OK in the dialog.

GUI site applications make hyperlink creation quick and easy, saving you development time. Generally, all that is required is content from which to link, and a destination to which the link will point. A hyperlink tool is generally provided in the toolbar or properties tab. You need only select or enter the file name to which you want to link, and the GUI site application creates the hyperlink for you.

For more information and hands-on labs about page text, tables and hyperlinks with GUI applications, see *Web Design Specialist*, Lesson 18: Site Development with Expression Web 3 – Basic Features, Lesson 20: Site Development with Adobe Dreamweaver CS5 — Introduction, and Lesson 21: Site Development with Dreamweaver CS5 — Basic Features.

3.4.5: Web forms with GUI site applications

A Web form is one way to obtain information from your Web site users. Creating a user-input form in X/HTML is not difficult, but simply time-consuming to write the code. However, processing the form can be challenging. GUI site applications (such as Microsoft Expression Web and Adobe Dreamweaver) make it easy to create forms quickly without knowledge of X/HTML by providing point-and-click functionality for form-field creation. However, not all applications offer form-processing options.

Expression Web makes it easy to create and process forms. When you create a form with Expression Web, you have the option to set up several different types of form processing if you elect to use FrontPage Server Extensions (FPSE). For example, the gathered form data can be saved in a tab-delimited, space-delimited or comma-delimited file, or you can save data to a Microsoft Access database. By specifying options that utilize FPSE, you allow Expression Web to automatically set up the processing and storage of collected information.

If you elect not to use FPSE, then form processing must be handled through client-side and server-side scripts, such as those created in Perl, Active Server Pages (ASP) or PHP.

Dreamweaver allows you to create and insert form fields using a simple click-and-drag action. However, Dreamweaver does not provide built-in server-side form processing, as does Expression Web. Therefore, you must implement a server-side script (using a server-side scripting language such as ColdFusion, Perl, Active Server Pages [ASP], JavaServer Pages [JSP] or PHP; or using an online tool such as Google's Custom Search Engine) to process the form data online. However, Dreamweaver offers the ability to generate client-side validation using JavaScript.

For more information and hands-on labs about Web forms with GUI site applications, see *Web Design Specialist*, Lesson 19: Site Development with Expression Web 3 – Advanced Features, and Lesson 22: Site Development with Dreamweaver CS5 — Advanced Features.

3.4.6: Creating templates with GUI site applications

A template is a document that specifies default settings or attributes. For example, a template can specify the layout structure, images, typeface, font colors and so forth used throughout a page or site. Because many of the pages in a site will use the same elements, it makes sense to create a template to reduce development time and to maintain consistency throughout a site. By using templates, you can also ensure that all your pages consistently adhere to the site's design specifications without having to manually format each page.

Most GUI site applications (such as Microsoft Expression Web and Adobe Dreamweaver) allow you to create and use page design templates. You simply designate and name a created page as a template. After a template is created and saved, it can be used anytime you want to create a new page with the specified properties. This saves you development time because you will not have to re-create identical page elements repeatedly as you develop the rest of the site. You can simply add or substitute any page-specific content into the template structure.

Expression Web and Dreamweaver enable you to ensure that specific page components remain consistent. When you create a template in Expression Web or Dreamweaver, all template areas are locked and cannot be edited unless specified. For example, if the template uses a common banner for all pages, the banner is locked on the template so it cannot be changed or moved in any page created with that template. The banner is considered a non-editable region, unless you specify to make it editable. When creating a template in these applications, therefore, you must establish certain editable regions that can be modified. Otherwise, when a new page is created with the template, nothing can be added or changed on the page, rendering the template useless. Specifying certain regions of the template as editable enables you to add content to these areas as you develop.

For more information and hands-on labs about creating templates with GUI site applications, see *Web Design Specialist*, Lesson 18: Site Development with Expression Web 3 – Basic Features, and Lesson 21: Site Development with Dreamweaver CS5 — Basic Features.

3.4.7: CSS in GUI page and site templates

Styles provide an effective way to maintain consistency throughout a site. Creating style templates for a site allows you to experiment with various styles on one page, then style multiple pages by attaching just one style sheet. This approach gives you flexibility while you design, saves you development time, and helps you unify the site pages. Further, using a style sheet helps you comply with W3C recommendations for Web page formatting.

How do styles differ from templates? Both are used to apply consistent properties to multiple pages. A template is a basic page structure upon which you can build and add varying content. A style is a style sheet that can be attached to the varying content on some or all pages to unify content appearance.

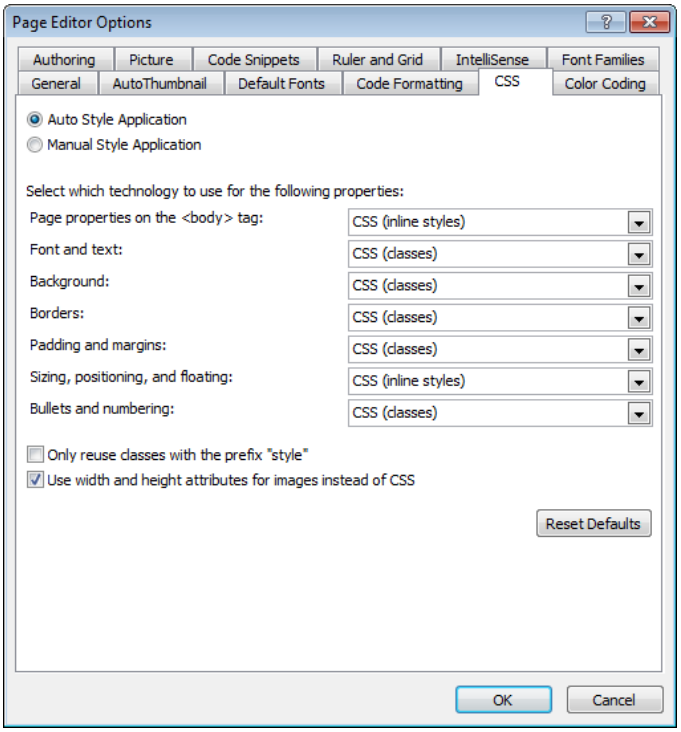
To implement styles in a GUI site application (such as Microsoft Expression Web or Adobe Dreamweaver), you typically select a style definition, and all aspects of the site then adhere to the rules of that definition. You can use an existing style sheet, or you can define your own styles and link your site pages to them.

You do not need to know how to code the styles because the application can write the code for you. All you need to know is how you want the style to look, then you can make the appropriate selections.

Styles can be applied three ways in Expression Web: They can be linked, embedded or inline. Expression Web allows you to choose which style you want to apply.

In general, local style rules take precedence over general style rules. That is, styles defined inline take precedence over those defined in an internal or external style sheet, and styles in an internal style sheet take precedence over styles in an external style sheet.

When you format items in a Web page, Expression Web automatically generates new styles (or modifies existing styles) in an internal style sheet or inline, as appropriate, unless you specify to create and/or modify styles in a linked external style sheet. Settings on the CSS tab of the Page Editor Options dialog box (shown in the following figure) determine whether Expression Web will generate inline or embedded styles.



Dreamweaver offers very good support for CSS positioning, and these positioning capabilities are collectively known as "layers" to distinguish them from table layout tools. Note that the layers created by Dreamweaver use absolute positioning. In absolute positioning, CSS is used to specify the exact pixel position of elements. You should use relative positioning (in which the positions of elements are specified relative to other elements) whenever possible in order to accommodate users with different screen resolutions. However, if you use absolute positioning, it is typical to lay out your Web site with a width of approximately 800 pixels. This guideline will allow a wide variety of users to browse your site without problems.

Browser support for CSS positioning is now good enough that any pages formerly laid out with tables can be converted to CSS yet retain their look and functionality. In addition, CSS positioning makes possible a much wider range of layouts, such as layouts with overlapping elements. Dreamweaver offers a feature for converting tables to layers, as well as converting layers to tables. In the past, the convert-layers-to-tables feature was useful for creating rough table layouts using the much more flexible layer layout tool. However, this practice is no longer recommended for use with any sites other than prototypes because it tends to generate less-than-optimal page layouts. Further, the W3C now recommends using CSS positioning for page layout instead of tables.

For more information and hands-on labs about CSS in GUI page and site templates, see *Web Design Specialist*, Lesson 19: Site Development with Expression Web 3 – Advanced Features, and Lesson 20: Site Development with Adobe Dreamweaver CS5 — Introduction.

3.4.8: Applying templates with GUI site applications

A Web page template is an X/HTML page structure (and sometimes an associated style sheet) that acts as the foundation for each page you create. A template specifies default settings or attributes. For example, a template can specify the layout structure, images, typeface, font colors and so forth used throughout a page or site. By using templates, you can ensure that all your pages consistently adhere to the site's design specifications without having to manually format each page.

Most GUI site applications (such as Microsoft Expression Web and Adobe Dreamweaver) allow you to create your own Web page design templates. Because many of the pages in a site will use the same elements, it makes sense to create a template to reduce development time. After a template is created and saved, it can be used anytime you want to create a new page with the same specified properties. This saves you development time because you will not have to re-create identical page elements repeatedly as you develop the rest of the site. To apply a template to a new page using a GUI site application, you simply begin with the saved template, then add or substitute any page-specific content into the template structure.

For more information and hands-on labs about applying templates with GUI site applications, see *Web Design Specialist*, Lesson 18: Site Development with Expression Web 3 – Basic Features, and Lesson 21: Site Development with Dreamweaver CS5 — Basic Features.

3.4.9: Validating source code with GUI site applications

The biggest benefit to validating your XHTML code is that pages written with valid XHTML will load faster than pages that use invalid code. This is because valid XHTML is predictable. The browser has to do less work to render validated XHTML code because every element has an opening and closing tag, attributes are enclosed in quotation marks, elements and attributes use lowercase letters, and so forth. With standard HTML, the browser often spends time inferring sloppily coded instructions or deciphering code errors before it can display the page. With valid XHTML, this extra work is not an issue.

Several tools and sites are available to validate the code in your XHTML documents, most notably the W3C Markup Validation Service (*http://validator.w3.org*). Many GUI site applications have built-in validation tools. Expression Web includes tools such as the Compatibility Checker and the Accessibility Checker that you can use to validate your code to current W3C standards.

Code validators use the information you provide in the <!DOCTYPE> tag to determine the standard by which they judge your code, so be sure your files include the correct DOCTYPE declaration at the top.

For more information and hands-on labs about validating source code with GUI site applications, see *Web Design Specialist*, Lesson 17: Site Development with Microsoft Expression Web 3 – Introduction, and Lesson 22: Site Development with Dreamweaver CS5 — Advanced Features. Visit the W3C Markup Validation Service at *http://validator.w3.org.*

3.4.10: Enforcing accessibility standards with GUI site applications

Microsoft Expression Web and Adobe Dreamweaver can be configured for compliance with current W3C code standards. For Web designers who are not particularly adept at writing X/HTML code, using a GUI development tool such as Expression Web or Dreamweaver is a good way to create Web pages that comply with W3C standards and thus promote accessibility, even without a detailed knowledge of the standards' specifications.

Both Expression Web and Dreamweaver offer dialog boxes that allow you to add alternate text for images, which provides accessibility for Web users who cannot view image files. Expression Web also includes tools such as the Compatibility Checker and the Accessibility Checker that you can use to validate your

code to current W3C standards. In Adobe Flash, you can use the Accessibility panel to provide a text equivalent for an object or group of objects, which optimizes content accessibility for screen-reader users.

Because you do not have full control over the code that a GUI site application generates, it is not possible to make code created with it completely XHTML-compliant or fully accessible to all users. However, you can validate the code and review the errors to learn about the types of non-standard code that the application includes in the files it creates, then manually correct the code in the document. You can also add or modify code manually to include accessibility features in your Web pages.

For more information and hands-on labs about enforcing accessibility standards with GUI site applications, see *Web Design Specialist*, Lesson 17: Site Development with Microsoft Expression Web 3 – Introduction.

Practice exam questions

Following are some practice exam questions intended for student review.

1. Which of the following accurately describes the use of CSS in GUI site development applications such as Microsoft Expression Web or Adobe Dreamweaver?

a. You cannot use CSS in a GUI site development application.
b. GUI site development applications support the use of CSS, but you must create the style sheets manually.
c. GUI site development applications include predefined style sheets that you may elect to use but cannot alter.
d. GUI site development applications support the use of CSS and can create the styles for you.

Correct response: d. GUI site development applications support the use of CSS and can create the styles for you.

Explanation: GUI site-development applications such as Expression Web and Dreamweaver offer good support for CSS. When you format items in a Web page, these applications automatically generate new styles (or modify existing styles) in an internal style sheet or inline, as appropriate, unless you specify to create and/or modify styles in a linked external style sheet.

2. Which statement accurately describes the user-input form creation process when using a GUI site-development application such as Expression Web or Dreamweaver?

a. GUI site applications do not currently offer any support for the creation of user-input forms.
b. GUI site applications currently provide utilities for creating server-side scripts, but you must create the forms manually using X/HTML.
c. GUI site applications allow you to drag form components onto a page to create user-input forms, even if you have no X/HTML knowledge; but not all applications offer form processing options.
d. GUI site applications currently offer full support for creating user-input forms, including form layout and processing options.

Correct response: c. GUI site applications allow you to drag form components onto a page to create user-input forms, even if you have no X/HTML knowledge; but not all applications offer form processing options.

Explanation: GUI site applications (such as Microsoft Expression Web and Adobe Dreamweaver) make it easy to create forms quickly without knowledge of X/HTML by providing point-and-click functionality for form-field creation. However, not all applications offer form-processing options. When you create a form with Expression Web, you have the option to set up several different types of form processing if you elect to use FrontPage Server Extensions (FPSE). If you elect not to use FPSE, then form processing must be handled through client-side and server-side scripts (such as those created in Perl, Active Server Pages [ASP] or PHP).

Dreamweaver allows you to create and insert form fields using a simple click-and-drag action. However, Dreamweaver does not provide built-in server-side form processing. Therefore, you must implement a server-side script (using a server-side scripting language such as ColdFusion, Perl, Active Server Pages [ASP], JavaServer Pages [JSP] or PHP; or using an online tool such as Google's Custom Search Engine) to process the form data online.

3.5: Site publishing and maintenance

This subdomain includes skills and knowledge required to publish and maintain a production Web site.

3.5.1: Staging/mock-up server

Web sites are often developed and tested on a Web server separate from the one that will host the published site on the Internet. This separate server is called a staging server. Other terms for a staging server are mock-up server or development server. The server on which the site is published and hosted is called the live server or the production server.

Staging servers are generally accessible only to the development team members who are working on the site and to authorized members of the organization for which the site is being developed. If you are developing a new Web site, the staging server can be the same server that will eventually become the live server hosting the site on the Web. If you are redesigning a Web site, however, it is likely that the customer will want the old site to be live while the new site is being developed. A staging server is necessary in these situations.

If you need to use a staging server, it should be as similar as possible to the server on which the site will be published. Most importantly, the following elements must be identical between the staging server and the production server:

- The software
- The operating system configuration

If these elements differ in any way, then your live production site will not function the same as it did when developed or tested on the staging server. Some differences may be minor but others could be fatal to your site.

The other important element to consider when configuring a staging server is hardware. If necessitated by budget constraints, staging servers can have slower processors and less RAM than production servers. The reason for this leeway is that the staging server will be accessed by only a few users at a time. However, if you expect a site to receive heavy traffic when it is published, you should have the same hardware on the staging server so that you can accurately test the Web site's performance under realistic production circumstances.

For more information about staging/mock-up servers, see *Web Design Specialist*, Lesson 34: Web Site Publishing and Maintenance.

3.5.2: Site testing

Before you publish your site to the World Wide Web, you need to perform testing tasks to ensure that the site appears and functions as intended. Publishing an untested site (or waiting until after the site is "live" to test it) will make any functional or usability problems in your site available for all Web users to see.

Testing is a critical part of every phase of Web development. From the earliest phases of a Web development project, your team should be conducting tests on the pages and site elements that you are creating:

- Programmers must thoroughly test any interactive elements they create for the site as they are developing them to ensure that they function as intended.
- Developers should test all pages on multiple browsers throughout the development process so they can modify code and functions to provide broad accessibility.
- The Web project manager should arrange to conduct audience usability tests on the site when it is as close to completed as possible. Usability testing assesses a site's effectiveness with objective users who have had little or no exposure to the site.

When all of the components of your Web site have come together and the site is finished and approved, everything on the site should be tested one last time. The entire Web project team, all stakeholders in the project and perhaps some objective people who have never seen the site should thoroughly test the site's functionality, including the navigation and any user-submission forms or programs. Be sure to look at every page, review site content one last time for spelling or grammatical errors, and test all the links. Your site is now ready to publish.

For more information and a hands-on lab about site testing, see *Web Design Specialist*, Lesson 6: Web Site Usability and Accessibility, Lesson 7: Browsers, and Lesson 34: Web Site Publishing and Maintenance.

3.5.3: In-house hosting vs. ISP or ASP hosting

To make your Web site available to users on the World Wide Web, you need a Web server to "host" your site. The server must have a dedicated connection to the Internet, so that any time your users want to obtain information from your site, they can. You can host your own Web server in-house, or you can contract an Internet Service Provider (ISP) to host your site. For hosting of specialized applications, such as contact management software or e-commerce sites, you may want to contract with an Application Service Provider (ASP). Typically, an ASP provides certain application functionality along with Web hosting, whereas an ISP simply provides access to the Internet and a place for your Web site to reside.

When it is time to publish your Web site, you generally need the following two pieces of information.

Information Required for Web Publishing	Description
The IP address or URL of the server to which you will copy your site	The IP address or URL usually refers to a Web server that will make your content available to users. The address can also refer to a backup server in case your Web server crashes.
A user name and password to access and publish your content to that server	Your Web server's user name and password are of critical importance and should be kept secure. The only people able to transfer content to your servers via FTP are those who have the passwords. User names and passwords ensure that only authorized persons, rather than any skilled Internet user, can publish or change content on your site.

Deciding how to host your Web site for public access can be difficult. You must choose whether to host the site in-house or use an ISP/ASP by analyzing the factors discussed in the following table.

Site Hosting Factor	Description	In-House Advantages/ Disadvantages	ISP/ASP Advantages/ Disadvantages
Cost	The cost of hosting your Web site can be the largest expense associated with site development. You will incur charges from an ISP or ASP that hosts your site, or you will pay costs associated with hosting and maintaining your own server in-house.	One disadvantage of hosting your Web site internally is the startup cost of purchasing a server and a dedicated connection to the Internet, which can be very costly. If you host your site internally, you might also need to hire personnel to continuously maintain the servers and network connectivity for your site.	If you contract an ISP or ASP to host your site, it will provide all equipment, service and maintenance for a fee. However, you will lose some control and options that you would have if you maintained your own servers. Typically, the cost savings of a hosting service are worth the loss of control over the server for smaller organizations.
Speed	Internet connection is usually referred to as bandwidth. The more bandwidth you have available, the faster users can access your Web site. However, as the bandwidth increases, so does cost.	A disadvantage of hosting internally is that you must pay all costs associated with establishing the best Internet connection possible and keeping that connection in service.	If you host your Web site externally, you generally have higher bandwidth available to you, and without the setup costs.

Table cont'd

Site Hosting Factor	Description	In-House Advantages/ Disadvantages	ISP/ASP Advantages/ Disadvantages
Reliability	Hosting internally or externally does not make your Web server more reliable; you must rely on the resources available to you to keep the server functioning properly.	If you host internally, your site may be the only site on your server, which will increase site performance over shared hosting. However, if the server fails in the middle of the night, who is going to restore it?	If you host your Web site externally, you may share a server with many different companies or sites, which can affect the reliability of your site. However, your ISP will probably have someone monitoring the server around the clock.

For more information about in-house hosting vs. ISP or ASP hosting, see *Web Design Specialist*, Lesson 34: Web Site Publishing and Maintenance. For a comprehensive list of ISP and hosting services, visit *www.thelist.com.*

3.5.4: Site publishing with FTP

The generic method of publishing a Web site uses a stand-alone File Transfer Protocol (FTP) application. The software used to transfer files via FTP is called an FTP client. The FTP client must communicate with an FTP-capable server. After communication has been established and any security requirements have been met, the file transfer process can proceed.

All FTP client applications allow for file transfers to and from the FTP server. FTP uses one of two modes to transfer data: binary and ASCII. The format used for any transaction depends on the type of data being transferred. Text-based files, such as X/HTML, CSS and XML documents, are transferred in ASCII mode. Non-text files, such as music and images, are transferred in binary mode.

Many FTP clients are available for purchase or as shareware downloads. Most have features that make transferring files over the Internet simple. Some offer a graphical interface designed for the beginning FTP user and also include more complex features for advanced users as well. Examples of such FTP clients include open-source FileZilla, the FireFTP Firefox plug-in and WS_FTP Professional by Ipswitch, Inc.

Generally, the FTP client interface has a left window and a right window. The left window lists files and directories on the local machine; the right window lists all the files and directories on the remote machine to which you are connecting. A graphical interface provides buttons for the most common functions, such as refreshing the file and folder listing, changing directories, and renaming files. The buttons in the left pane control the files on your local computer, and the buttons in the right pane control the files on the remote server. Left and right arrow buttons in the middle of the interface (between the two windows) allow you to transfer files from the local machine to the remote machine, or vice versa.

Many clients also allow you to customize settings such as your default e-mail address, local network settings or security settings. Advanced support may include handling frequently visited FTP sites, timeouts, port numbers and passive file transfers; enabling remote-to-remote transfers; command-line support for automating file maintenance operations; specifying ASCII, binary or auto-detect as your default transfer mode; and automatic conversion of file name extensions on the local or remote machine to ensure that all file names are uniform (e.g., every .htm file name extension becomes .html).

For more information and hands-on labs about site publishing with FTP, see *Web Design Specialist*, Lesson 34: Web Site Publishing and Maintenance. For a list of stand-alone FTP clients, visit *http://download.cnet.com/windows/* and search for FTP.

3.5.5: Configuring DNS entries

Although DNS configuration is generally performed by a server administrator, it is useful to know a little about the structure of entries in a domain name server, described in the following table.

DNS Configuration Component	Description	Example
Address (A) record	Defines the location of your Web site. The basic structure of an A record is as follows, where xxx.xxx.xxx.xxx is the IP address: *domain.com. IN A xxx.xxx.xxx.xxx*	The A record for the habitat.org domain would be: *habitat.org IN A 63.246.14.100* The A record simply says that the IP address for the habitat.org domain name is 63.246.14.100.
Subdomain	Subdomains are used to subdivide a domain, and they do not have their own IP addresses. The subdomain is specified before the domain name in a Web address. Internet traffic going to a subdomain will first access the domain's IP address, then go to the subdomain location.	For Web sites, the most common subdomain is www, which is specified as follows: *www.habitat.org* Other common subdomain names include mail, ftp and news. For example: *news.habitat.org* For Web sites, subdomains are usually configured as shortcuts for subdirectories. For example, *news.yourdomain.com* will usually also be accessible by going to *www.yourdomain.com/news*.
Shared domain	A domain owned by someone who hosts other people's Web sites as either subdirectories of the domain or as subdomains. By using a shared domain, you eliminate the need to register your own domain name. The disadvantage is that you do not have control over the domain, and shared domains may not look as impressive to your audience as your own domain name would.	For example, if you use Yahoo! for Web hosting, your Web site address might be *www.yahoo.com/yoursite* or *http://yoursite.yahoo.com*.

For more information about configuring DNS entries, see *Web Design Specialist*, Lesson 32: HTTP Servers and Web Applications. For more information about valid domain names, visit *www.dns.net/dnsrd/rfc/*.

3.5.6: Site security issues

Because Web servers and Web sites are connected to the Internet, they are particularly vulnerable to certain types of security issues and attacks, as described in the following table.

Site Security Threat	Description	Examples	Countermeasures
Social engineering	A practice hackers use to try to trick authorized persons into providing information that can be used to access restricted computer resources. Sometime it involves the hacker posing as someone authorized to have the requested information. Another technique is finding seemingly benign reasons to request information that has no apparent relationship to system security. Social engineers typically exploit the fact that most people want to be helpful when they can.	A common technique is to call the reception desk of a large company and pose as a remote contractor who forgot the password to the company's network. Another example involves a study in which students gave teachers and administrators at their school a survey that included questions about birthdates, spouses' and children's names, pets' names, and so forth. The students then tested words provided in the survey answers as passwords using password-cracking software, and managed to access private files in the school's systems.	Network security — like the security of your own personal information — demands that you verify identity before providing any sensitive data to anyone. Organizations should train their employees about the types of information that could be used to access systems, as well as ways they can identify social engineering attempts, identity verification processes they can use, people to whom they can provide data and the safest ways to provide it.
Denial-of-service (DOS) attacks	An attack on a server that consumes all of the server's available network bandwidth and thus causes a loss of service to users.	High-profile Web sites are frequently the targets of DOS attacks. However, many small Web sites have also been victims of DOS attacks launched against their hosting providers. In other words, if someone perpetrates a DOS attack against the company that provides you with bandwidth for your Web site, your site will also be deprived of bandwidth as a result of being "downstream" from the main target of the attack.	DOS attacks can sometimes be stopped by filtering out network traffic from the attacker. However, this defense is not as effective when the attack is distributed, or launched from multiple sources. Owners of smaller Web sites cannot do much to fight a DOS attack except to contact the hosting and bandwidth provider, which can try to thwart the attack and prevent future attacks with a combination of techniques. Prevention techniques include performing regular (even constant) server log file analysis to look for suspicious activity; setting network rules to look for and block abnormally high amounts of traffic coming from certain hosts; and installing faster hardware that can handle larger amounts of traffic without failing.
Brute-force attacks	The least sophisticated but most frequent types of attacks on servers. A brute-force attack occurs when someone tries any of various techniques to hack straight into a server, usually without any specific knowledge or information about the server, and often with the help of scripts written for this purpose.	One type of brute-force attack is the dictionary attack, in which the attacker runs a program that repeatedly enters passwords from a long list of common passwords, hoping that one will work and admit access.	Dictionary attacks can be thwarted by enforcing a policy of all users choosing strong passwords for your system resources and periodically changing those passwords. Other types of brute-force attacks can most often be prevented by keeping the server and its operating system up-to-date with security patches and configurations, and by using a firewall.

For more information about site security issues, see *Web Design Specialist*, Lesson 34: Web Site Publishing and Maintenance.

3.5.7: Server security

Security is always an important consideration for server administrators. Hackers or other unauthorized users who find ways to gain administrator access privileges to an organization's servers can make unauthorized and potentially damaging changes to files and data. They may also gain access to confidential information. You should understand several general security principles to help keep your servers safe, as discussed in the following table.

Server Security Technique	Description
Use strong passwords	Some good practices when choosing a password are to avoid using common words or names, to include a mix of uppercase and lowercase letters, and to include numbers or symbols.
Disable any service you do not need	If you do not plan to use FTP on your server, for example, you can use a firewall to block access to FTP port 21. The fewer points at which your server can be accessed and the fewer server services you expose to the Internet, the more secure your server will be.
Apply security patches as soon as they are released	Server and operating system vendors regularly update their software by offering security patches when new vulnerabilities are discovered. Usually, the vendor has a mailing list to which you can subscribe to be automatically notified of new versions or security patches.
Restrict access	Give each user access on the server only to what he or she needs to perform job duties. Although it is easier to simply give all users the same level of access to your server, this creates unnecessary risk. If some users will not need to modify documents on the server, then they should have only read access. If some users do not need an entire area of the server available to them for any reason, then give them no access to it at all.

For more information about server security, see *Web Design Specialist*, Lesson 34: Web Site Publishing and Maintenance.

3.5.8: Web site maintenance

After you have published a Web site, it is essential that the site be properly maintained. With experience, you will improve your skills in maintaining and revising your Web site. Web publishing is quick and easy, so experimenting with design and element positioning is often the best way to improve a Web site that is already live.

The following table discusses common methods for maintaining a Web site.

Site Maintenance Technique	Description	Methods
User feedback	One of the best ways to learn how your site can be improved. User feedback can be: - Direct, such as when a user sends an e-mail saying that a link is broken or that he or she cannot determine how to use a particular site feature. - Indirect, such as that provided by server logs. Many revisions to a site are made in response to user feedback, both direct and indirect.	You can obtain direct user feedback by providing an e-mail address for users to contact the Web team with comments or problems; by inviting visitors to submit a survey form in which you ask specific questions about the site; or by sending a survey to visitors who agree to receive and respond to it. Server logs provide excellent user feedback about your site because they reveal every user's actions on your site without taking the user's time or effort to answer questions. For example, server logs can identify: - Most and least popular pages - Browsing time - Navigation sequence - Demographics - Repeat visits
Link checking	Hyperlinks on your Web site may become invalid as a result of changes you make to your site or changes in sites to which your site links. Bad links make your site look outdated and neglected, and are a sure way to discourage visitors from returning. For this reason, an important step in site maintenance is checking all your site links regularly. There are two methods for checking links: manually and automatically.	Manual link checking is the process of reviewing every page on your site and clicking each link to verify that it still points to the correct location. If you perform link checking manually, it is very important to be methodical about the process so you do not miss any links. Automatic link checking involves using a software application to test links on the site. An automatic link checker is useful for learning whether pages to which your site links have changed address or no longer exist. However, an automatic link checker cannot tell you whether the content on a page to which your site links is still relevant — the link checker only verifies that a page exists at the URL to which you linked. For this reason, it is important to use a combination of automatic link checking and manual link checking.
Revising site features	Revising a Web site may entail adding new content, fine-tuning the structure or, in a worst-case scenario, redesigning the entire site. If your site was planned well, it should be fairly easy to add content. More radical revision efforts might be required for sites that were poorly designed and poorly implemented, or sites that simply no longer work for the site owner's purpose and goals.	If your Web site is in need of extensive revision or redesign, you should carefully evaluate the site and fully document its failures before planning the revised site. Without understanding the problems, you may repeat the same design errors. Similarly, if your site is receiving many hits and you want to expand its scope, consider the entire site before simply adding a page here and there. Although the design may have been adequate for the original site, it may not accommodate growth well. As with your original site design process, you should always have other users test and evaluate your revised site. Sometimes changes do not improve the site, and it may be worthwhile to revert to a previous version. Always keep copies of files from the previous generation of your site in case you need to return to them.

For more information about Web site maintenance, see *Web Design Specialist*, Lesson 34: Web Site Publishing and Maintenance.

3.5.9: Documenting site changes

After a Web site has been published, it is important to begin keeping a record of any and all changes made to the site. It is also wise to keep backup copies of previous file versions when changes are made to them, in case you need to revert to previous designs or content. The process for documenting changes can be as simple as creating a spreadsheet with columns for dates and descriptions of the changes.

Larger Web sites may use a version-control software system, which stores a history of each file along with notes about each change that was made to the site. Version-control systems can make the process of publishing Web site files more complicated. However, they also provide you with a complete record of each change, and allow you to easily restore any file or the entire site to its exact version from any previous date.

For more information about documenting site changes, see *Web Design Specialist*, Lesson 34: Web Site Publishing and Maintenance.

Practice exam questions

Following are some practice exam questions intended for student review.

1. What is a staging server?

a. A server used for development and testing of your Web site
b. A server used as backup for the server hosting your Web site
c. A server used for hosting the database(s) integrated with your Web site
d. A server used to store source files for the pages, images and other elements of your Web site

Correct response: a. A server used for development and testing of your Web site

Explanation: Web sites are often developed and tested on a Web server separate from the one that will host the published site on the Internet. This separate server is called a staging server. Other terms for a staging server are mock-up server or development server. The server on which the site is published and hosted is called the live server or the production server. Staging servers are generally accessible only to the development team members who are working on the site and to authorized members of the organization for which the site is being developed.

2. Web site maintenance tasks include maintaining site functionality and finding ways to continuously update and improve your site. Which of the following tools provides indirect user feedback that you can use to determine whether parts of your site need to be revised?

a. Server logs
b. User surveys
c. Automatic link checking
d. E-mail form for users to report site problems

Correct response: a. Server logs

Explanation: User feedback is one of the best ways to learn how your site can be improved. User feedback can be direct, such as when a user sends an e-mail saying that a link is broken or that he or she cannot determine how to use a particular site feature. User feedback can also be indirect. For example, you might notice that a certain area of the site gets very little traffic. Server logs provide excellent user feedback about your site because they reveal every user's actions on your site without taking the user's time or effort to answer questions.

Domain 4: Advanced Web Technologies

This section will discuss CIW Web Design Specialist Domain 4 exam objectives covering advanced Web technology skills and concepts.

4.1: Multimedia and plug-in technologies

This subdomain includes skills and knowledge required to use multimedia and plug-in technologies to enhance a Web site.

4.1.1: Multimedia Web design principles

One of the most common misconceptions about Web design is that a good site must dazzle the user with a multimedia experience, and that the content of the site is of secondary importance. As a Web designer, you want your site users to have a satisfying experience, but dazzling them is not necessarily your goal. If multimedia makes sense and enhances the usability of a site, you should use it. If multimedia does not enhance the user experience, or if it degrades the user experience by creating an unnecessarily long download, then you should not use it.

Planning is the key to successful creation of a multimedia Web site. Although multimedia has existed for some time, applying it on the Web is not an exact science. Multimedia on the Web is made possible with various tools and technologies that are not always compatible. Multimedia is gaining popularity on the Web because advances in Internet technology now allow developers to mix different media objects in Web pages. Multimedia such as animation, audio and video can supplement bland text or two-dimensional graphics, as well as complement the visual design, tone and message of a Web site.

However, multimedia technologies also present great challenges to Web developers. Predicting how they will react and display on various platforms and in different browsers requires planning, patience, and lots of trial and error. Further, the appeal of using complex technologies can sometimes overshadow the goals of a site, and even discourage or exclude some audience members from using it.

The current capabilities of multimedia on the Web are astounding. For a Web designer, it is tempting to utilize these tools and technologies. However, today's designers are limited by two major factors that inhibit rapid technology adoption:

- Bandwidth
- Browser support technology

To enjoy the full effect of many new technologies, a high bandwidth connection is usually required, as well as browser plug-in support or other third-party applications. Therefore, Web designers should resist the temptation to use the latest and most dazzling technology, because the standard user on a dial-up connection with a standard browser cannot support it. This limitation does not mean that you cannot use new technology, but you must be aware that you will exclude some of your audience. A balance in discipline is required to use the technology against the functionality of the site. If your defined audience is a particular segment that will be able to support the technology and you are not concerned about other users, then use any technology that enhances the site.

A question commonly asked specifically about Flash is whether or not to create a site entirely with Flash. The answer depends on several factors. Although simple in terms of using its graphical interface for development, creating Flash movies can still be time-consuming. You must also consider the updating requirements of your content. Updating content with X/HTML is becoming more common in the workplace, but updating Flash content is still a specialized skill. Therefore, if the site you are developing will require frequent content updates, or if inexperienced Flash developers will perform updates, then it is not advisable to develop a site entirely in Flash.

Common uses for Flash generally serve the purpose of rich multimedia design. For example, a splash page provides a quick movie introduction lasting from 10 to 30 seconds before refreshing automatically to the home page. When using the splash page technique, always provide a link for users to bypass the introduction. Return visitors may not want to see the splash page again. Flash can also provide effective navigation elements that offer an enhanced experience while users are navigating the site.

For more information about multimedia Web design principles, see *Web Design Specialist*, Lesson 1: Overview of Web Design Concepts, Lesson 10: Multimedia and the Web, and Lesson 25: Multimedia with Adobe Flash Professional CS5.

4.1.2: Accessibility issues with images and animation

It is important to design your Web sites so that they can be used by people with disabilities. Several Web browsers are available for specialized purposes, as described in the following table. The use of these types of browsers, as well as older-version standard browsers, presents accessibility issues for Web page designers.

Specialized Browser Type	Description
Mobile device browsers	The earliest Web browsers designed for mobile devices (such as mobile phones and PDAs) required that Web designers create special versions of their sites. Later releases were simplified versions of Web browsers optimized for small screens and low processing power. These typically were not full-featured, and some did not display images or multimedia. Today, however, many modern mobile browsers such as Opera Mini and Apple Safari offer desktop-class Web browsing (rich content, CSS 2.1, Ajax, etc.) on the touch screen of a handheld device.
Adaptive browsers/voice browsers	Web browsers that enable users with disabilities to use the Web are referred to as adaptive browsers. They typically feature screen magnification; the ability to use text readers, captioning and speech synthesis to convey Web page content; and voice recognition functionality to receive commands from users. They may also feature the ability to translate Web pages to Braille.

Good Web graphics must be aesthetically pleasing, relevant to the site's purpose and, most importantly, small in file size. File size directly affects download time and is a key consideration for accessibility. According to WebSiteOptimization.com (*www.websiteoptimization.com*), about 95 percent of the Internet connections in the United States are broadband (as of December 2009). Although this does mean that most users can view content that requires high bandwidth, it also means that some users are still connecting with a modem. The most common modem connection speed today is 56.6 Kbps. And even with a high-speed Internet connection, images that are unnecessarily large will take longer to download than users may be willing to wait.

Designing for accessibility also includes using the *alt* attribute with all images (especially navigation elements) so that they can be read by voice browsers and text readers. Images in Web pages should always include the *alt* attribute to specify descriptive alternative text. The *alt* attribute provides alternative text in place of an image on an XHTML page. The syntax for using the *alt* attribute is as follows:

```
<img src="filename.gif" alt="This is a link to the search page"/>
```

For reasons of user accessibility, the XHTML Transitional specification requires the *alt* attribute for your code to validate to the standard. The *alt* attribute is useful for several situations:

- Users who have disabled the image-viewing capability on their browsers can read the image name or link destination.
- Users can read the image name or description while an image loads.
- Software can provide accessibility options for vision- and hearing-impaired users.

- Netscape Navigator version 4.x (and later) and Microsoft Internet Explorer version 3.x (and later) can provide pop-up tool tips.

Animation is an important component that distinguishes the Web from other media (except television). Like other types of multimedia, however, animation poses some specific accessibility issues for Web users with sight or hearing disabilities or specialized software. If you include animation in your site, you should provide alternative content to maximize accessibility for all users. Short animations that can be described in a few words can also use the *alt* attribute. For longer animations, however, you must take additional steps to ensure accessibility. By providing text descriptions of an animation file's content, you ensure that your site visitors who use text readers will be able to use the content. If your animation includes narration, you can include captioning to make the content accessible to users with hearing disabilities.

Flash is a widely supported format for vector animation and rich Web content. According to its vendors, the Flash Player plug-in is currently installed on more than 99 percent of all Web browsers. However, this fact does not guarantee that 99 percent of Web users will be able to use the Flash movies you create. If your Flash movie requires the latest version of the Flash Player, some users will not be able to view it without upgrading their browser plug-ins. In addition, Flash files are not yet widely supported on mobile devices, and your Flash content may be inaccessible to users with disabilities.

To increase accessibility when using Flash content on your Web sites, you should advise your site users with a small note that states your site uses Flash content. You can check your users' browsers for the correct version of the Flash Player plug-in, and provide a link for users to easily download and install it. For content that is essential to a Web page or to understanding your site, be sure to provide an X/HTML version of the content whenever possible. You can also design Flash content in ways that increase its accessibility for users with mobile Web-access devices or specialized browsers. The Adobe site provides information about accessibility best practices and techniques for developing Flash files, which include creating text equivalents for visual elements, captioning audio content, giving users motion control and keyboard access, and validating Flash files for accessibility.

For more information about accessibility issues with images and animation, see *Web Design Specialist*, Lesson 7: Browsers, Lesson 9: Web Graphics, Lesson 10: Multimedia and the Web, and Lesson 29: Multimedia with Flash Professional CS5 — ActionScript, Masks and Practical Uses. For more information about accessibility and Flash, visit the following Adobe sites: Best Practices (*www.adobe.com/accessibility/best_practices.html),* Adobe Flash Professional CS5 Accessibility (*www.adobe.com/accessibility/products/flash/)* and Mobile and Devices Developer Center (*www.adobe.com/devnet/devices).*

4.1.3: SWF technology features and software

Flash technology (SWF) combines four elements that define its functionality: vector graphics, streaming capability, a timeline, and layers. These four elements combine to create versatile and compact animation files. Flash also has the ability to create some motion automatically. The following table describes these key aspects of Flash.

Flash Technology Feature	Description
Vector graphics	Flash uses vector graphics, rather than bitmapped graphics such as GIF, JPG or PNG. Vector graphics perform more efficiently on the Web because they are based on mathematical computations, rather than the pixel-by-pixel information used by bitmaps. As such, Flash graphics can be scaled without file size being affected. Compact file size is one of Flash's great advantages and is partly responsible for its success as a Web medium.
Streaming capability	Streaming capability is another important feature that increases Flash's Web compatibility. Streaming allows multimedia content to begin playing as soon as it reaches its destination — in this case, the client browser. The Flash movie begins to play as soon as the initial information about the Flash file reaches the browser. Therefore, the user can begin watching the movie while the rest of the data continues to download.

Table cont'd

Flash Technology Feature	Description
Timeline	The Flash timeline is a sequencing component that controls the way in which a Flash movie plays. The Flash movie's motion is controlled by the timeline, which is divided into frames. As you develop more frames (or longer timelines), the movie begins to take action. Each frame contains vector graphics that are opened at a designated sequence and speed, creating the animation. The timeline also enables you to specify the speed at which the movie is played, and the points at which the movie starts or stops playing, such as when the user moves his or her mouse.
Layers	Each Flash movie can have multiple layers, thus providing animation that is not only linear but also parallel: One animation sequence can run on top of another because each is on a different layer. In Flash, layers are contained within the program and handled by the plug-in; thus layers do not pose browser-compatibility problems in Flash files as they do in Web pages.
Automated animation ("tween")	Flash uses the term "tweening" to refer to its automated animation capability. In tweened animation, you provide beginning and ending frame information, then the Flash application creates the frame values in between. This process minimizes file size compared to frame-by-frame animation because Flash stores only the values for the changes rather than all values for each entire frame. There are two basic types of tweens: **-** Motion tween —objects move from one location to another on the stage. - Shape tween —objects change their shape or form on the stage.

For more information about SWF technology features and software, see *Web Design Specialist*, Lesson 25: Multimedia with Adobe Flash Professional CS5, Lesson 26: Multimedia with Flash Professional CS5 — Timeline, Layers, Symbols and Buttons, and Lesson 27: Multimedia with Flash Professional CS5 — Tweens.

4.1.4: Using SWF-authoring software

The Flash application is a development and testing tool that allows you to author Shockwave-Flash (SWF) animation files for use in your Web pages. Flash can be used to create objects and images, eliminating some of the need to use other applications to create graphics and then import them. Flash also supports the importation of graphics in common file formats such as GIF, JPG and PNG. The following table describes some of the main functionality features of the Flash authoring software.

Flash-Authoring Functionality	Description
Timeline frames	Each frame on the timeline has its designated moment in the Flash movie, and the movie's action in that frame depends on the action you have assigned to it. You need not assign content to every frame. In fact, many frames can create the content themselves (called tweening). The timeline consists of three basic types of frames, each of which serves a different purpose: - **(Normal) frame** — a Flash frame designed to extend the movie to a certain point along the timeline. No content can be placed in a normal frame; it inherits content from the preceding keyframe. - **Keyframe** — a Flash frame containing objects that will be displayed in all frames following it, until the timeline encounters another keyframe or a blank keyframe. - **Blank keyframe** — a milestone Flash frame that contains no content. When a blank keyframe is inserted, all preceding content is stopped, and new content can begin. If the only keyframe is placed at the beginning of the movie, the movie will start as soon as it reaches the browser, and will continue to play until the end without stopping. To allow user interaction with your site, it is not always desirable for the movie to begin playing without breaks when it loads. Therefore, you must insert keyframes along the timeline to control the movie.
Symbols	Flash has the ability to define and reuse objects called symbols. A symbol is a graphic, a button or a movie clip that is stored in a Flash movie's library. Symbols can be created after an object is imported into Flash. Flash automatically adds the file to the library for the current movie.

Table cont'd

Flash-Authoring Functionality	Description
Buttons	Flash buttons are used extensively in Flash movies. A button is triggered by mouse events. It can then perform as a standard rollover button, or it can be used to launch embedded movies. You can either use a button from the default Flash Library or you can create a button by using Flash or by importing images. You can add user-triggered interaction to buttons in Flash by using ActionScript, which is a client-side scripting language similar to JavaScript.
Tweening	Flash uses the term "tweening" to refer to its automated animation capability. In tweened animation, you provide beginning and ending frame information, then Flash creates the frame values in between. There are two types of tweens: - **Motion tween** — objects move from one location to another on the stage. - **Shape tween** — objects change their shape or form on the stage. When tweening objects, you must comply with certain rules: - Only one tween can exist in a layer at a time. - In a motion tween, the objects must be symbols or movie clips. - In a shape tween, the objects cannot be symbols.
Motion tweens	In versions prior to CS4, tweening was timeline-based. In Flash CS4 and CS5, however, motion tweens are object-based. The result is that there are two types of motion-related tweens: - **Classic tweens** — the old type of timeline-based motion tween. Classic tweens use keyframes. To create a classic tween, you place a symbol in a keyframe, then create another keyframe further down the timeline in which you reposition the symbol. You then right-click in the timeline between the two keyframes and select Create Classic Tween to add a classic tween to the layer. - **Motion tweens** — the object-based tween model. A motion tween consists of one target object over the entire tween span. Motion tweens can have only one object instance associated with them, and they use property keyframes (frames in which you explicitly define one or more property values for the target object of the tween span) instead of keyframes. A motion tween span is treated as a single object in the timeline To create a motion tween, you need only place an object on the stage, right-click the object and select Create Motion Tween. Flash will convert the object to a movie clip (if it is not already a tweenable object type), create a motion path, and insert frames to spread the motion across a one-second time span. You can then edit the object anywhere along the motion path. You can also edit the motion path itself.
Shape tweens	Shape tweens offer great flexibility because you can transform any shape into any other shape you want. For example, a shape tween can start with a circle and metamorphose into a square. You simply provide the beginning and ending shapes and positions for the tween. Remember that you cannot apply shape tweening to symbols in Flash because symbols are reusable shapes that do not change. They can move (and be modified slightly) but they cannot be reformed to a new object as a shape tween will do. Therefore, you can animate metamorphosis between shapes you have drawn, as well as text or images (such as your company logo) that you have not converted to Flash symbols.
Movie clips	Flash movie clips are different from movies only in that a movie clip uses a different timeline. A Flash movie clip is an animation that is embedded into a Flash movie yet runs independently of the movie. You use the same techniques to develop movie clips that you use to develop movies. However, the movie clip is a Flash symbol; therefore you use the Symbol Edit mode to develop the clip. The movie clip is called by an action in the main movie, and then it plays.
Masks	A mask is a special type of layer that covers an area of the stage, allowing a part you specify to show through. The effect is similar to looking through a stencil, in which the border defines the object. Masks are commonly used for effects such as a searching light moving over text, highlighting some text in passing. To create a mask, you simply change the layer type to Mask.

For more information and hands-on labs about using SWF-authoring software, see *Web Design Specialist*, Lesson 26: Multimedia with Flash Professional CS5 — Timeline, Layers, Symbols and Buttons, Lesson 27:

Multimedia with Flash Professional CS5 — Tweens, Lesson 28: Multimedia with Flash Professional CS5 — Movie Clips, and Lesson 29: Multimedia with Flash Professional CS5 — ActionScript, Masks and Practical Uses.

4.1.5: SVG characteristics

Scalable Vector Graphics (SVG) format is a W3C Recommendation that uses Extensible Markup Language (XML) to describe certain shapes.

Similar to other vector graphics, SVG is best for working with two-dimensional line art and shapes. Because SVG is a vector graphics format, it is also scalable. A circle image stored as a vector graphic has the same file size whether it is saved with a diameter of 90 pixels or 900 pixels. In addition, the use of XML in SVG files allows the graphic to become an object in the X/HTML page, enabling access to the full XML document object model (DOM). Therefore, filters, masks, scripting and mouse events can be included in SVG files. And because SVG graphics are stored as simple XML files, they are searchable and can be easily manipulated with Web programs to support zooming, rotation, movement and other types of image manipulation.

For more information about SVG characteristics, see *Web Design Specialist*, Lesson 9: Web Graphics. To read the W3C's SVG 1.1 specification, visit *www.w3.org/TR/SVG11*. For more information about SVG, visit *www.w3.org/Graphics/SVG* and *www.mozilla.org/projects/svg/*.

4.1.6: SWF and SVG files in X/HTML pages

When you create a Flash movie, you produce a file with the .fla file name extension (on the Windows platform). The FLA file format can then be converted and compressed into a SWF file, which is then inserted into the X/HTML code using the <object> tag (for newer browsers) or the <embed> tag (for some old browser versions) to display in the browser. When the browser encounters the SWF file, the Flash Player plug-in is used to display the Flash movie in the browser. Thus, the only X/HTML code on which Flash relies is the <object> and <embed> tags.

Adding a Flash file to an X/HTML page is very similar to adding graphics. The easiest way to add a Flash movie to an X/HTML file is to let Flash create the required HTML code, then you can insert that code into the X/HTML page that will display the Flash movie.

In any implementation, you use the X/HTML <object> tag to embed SWF and SVG files into Web pages, as you would with other Web page objects such as Java applets. The following code will embed an SVG file within an X/HTML file:

```
<object type="image/svg+xml" data="myanimation.svg" name="myAnimation" width="400"
height="150">
</object>
```

However, remember that some older browsers do not support the <object> tag. To provide backward compatibility with older Netscape browsers, it is good practice to include the <embed> tag in your X/HTML pages in addition to the <object> tag. You can do this by placing the <embed> code between the <object> </object> tags. The following code generated by Flash when publishing a SWF file demonstrates this:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0
" width="226" height="129" id="header_movie" align="middle">
    <param name="allowScriptAccess" value="sameDomain" />
    <param name="allowFullScreen" value="false" />
    <param name="movie" value="header_movie.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#ffffff" />
```



```
    <embed src="header_movie.swf" quality="high" bgcolor="#ffffff" width="226" height="300"
    name="header_movie" align="middle" allowScriptAccess="sameDomain" allowFullScreen="false"
    type="application/x-shockwave-flash" pluginspage="http://get.adobe.com/flashplayer/" />
</object>
```

Remember also that the end-user's browser must have the correct plug-ins to view these files.

For more information and hands-on labs about SWF and SVG files in X/HTML pages, see *Web Design Specialist*, Lesson 25: Multimedia with Adobe Flash Professional CS5, Lesson 26: Multimedia with Flash Professional CS5 — Timeline, Layers, Symbols and Buttons, Lesson 28: Multimedia with Flash Professional CS5 — Movie Clips, and Lesson 29: Multimedia with Flash Professional CS5 — ActionScript, Masks and Practical Uses.

4.1.7: SWF and SVG in instructional design

SWF and SVG files are popular formats for Web applications that require a greater degree of interactivity than can be provided with X/HTML. One such type of application for these formats is instructional design, which includes multimedia tutorials, demonstrations, online-learning environments, and online assessments such as quizzes. In addition, companies frequently use Flash animations and SVG to demonstrate products or processes on the Web.

SWF combines the benefits of desktop software with the benefits of the Web. Like desktop software or CD-ROMs, Flash movies can include audio, video, animation, database access and user interaction. Unlike traditional desktop software (but like the Web), Flash can enable interaction between people at different locations. Following are some more of the many benefits that make SWF and SVG excellent formats for online education and training (i.e., e-learning) environments:

- SWF files can provide a high level of interactivity while not requiring software to be installed on the user's computer (except for the Flash Player plug-in).
- Students can search SVG images to readily find information they need to learn.
- Because SWF and SVG files rely on browser plug-ins, the same files can be used on most computers and with the most popular operating systems (including Windows, Linux and Macintosh). This platform-neutrality helps ensure that all students can access the information.
- Vector graphic formats such as SVG and SWF can have very small file sizes. Small file sizes and client-side processing mean that these applications can create a much richer online learning experience, with a large amount of rich content in a relatively small download.
- SVG and some SWF files can be resized to allow sight-impaired users to more easily see material.

Remember to consider your audience and the accessibility of your content before you incorporate SWF and SVG files in your Web pages for any purpose.

For more information and a hands-on lab about SWF and SVG in instructional design, see *Web Design Specialist*, Lesson 29: Multimedia with Flash Professional CS5 — ActionScript, Masks and Practical Uses.

4.1.8: Plug-in/viewer technologies

Plug-ins extend the browser's capabilities to view and interact with multimedia file formats not directly supported by a browser, which includes most of the available Web multimedia that you might consider using on your Web site. Plug-ins make it possible to view videos, experience three-dimensional worlds, listen to audio files and interact with dynamic multimedia objects.

A plug-in is a dynamic application associated with a specific platform and browser, such as Windows 7 or Internet Explorer. When a browser encounters a file type that is not directly supported, it launches a plug-in application that is associated with that file type. When the application launches, it may take the form of a new user interface, or it may open or play the file within the browser window. Plug-ins translate, filter, produce and/or consume Internet and intranet data.

Before you can take advantage of many plug-in features, you often must first process your multimedia files through an encoding application provided by the plug-in manufacturer. These applications compress and streamline your multimedia files for later interpretation and playback. Unfortunately, many encoders create a proprietary format that can be deciphered only by the specific manufacturer's plug-in. Many of these plug-in compressors minimize download time, minimize file size, and tremendously increase the interaction potential of multimedia. The Web developer's initial plug-in selection is critical to long-term site maintenance. Creators of new multimedia standards often make the plug-in applications freely available; however, content developers must purchase licensed tools for creating the content to be used by the associated plug-in. Examples of this include Flash, the RealAudio format and QuickTime video. Many plug-ins can also use standard file formats, such as SVG or MP3, for which authoring tools are free or inexpensive.

The following table discusses some of the most common file types supported by plug-in/viewer technology.

Plug-In / Viewer	File Format	Description
Adobe Shockwave and Flash Players	Shockwave-Flash (SWF)	Adobe Shockwave and Flash Players can display and play back a wide range of SWF multimedia content, including interactive games, multimedia user interfaces and audio. Shockwave was introduced to the Web in 1995 and Flash appeared in 1997. Since then, thousands of sites have been designed using their multimedia content. The Shockwave and Flash development applications are available for Windows, Macintosh, Linux and some other UNIX platforms.
Adobe Reader	Portable Document Format (PDF)	Adobe introduced an application called Acrobat to help Web professionals create and publish distributable documents from existing files quickly. This tool creates PDF files, which can be read by the Adobe Reader, a browser plug-in that is downloadable for free. Many paper-based documents that already exist within a company or organization can be converted to PDF files to maintain their original formatting. This capability ensures that all documents will render the same as originally intended.

Your job as a Web developer is to shape the user's experience at your site. A critical aspect of the planning process is providing effective ways for first-time users to download and install plug-ins. Although plug-in installation procedures are quickly becoming standardized, sometimes unpredictable results can occur. This is because so many different configurations are possible depending on the user's choice of browser, operating system and plug-in. Plug-ins can be installed in three ways, as described in the following table.

Plug-In Installation Method	Description
Online installation	Performed without exiting a browser session. If possible, you should design your Web pages so the user has something else to see or read during plug-in installation or large file downloading. Some online plug-in installations require you to restart the computer to take effect. Like offline installation, this type of online installation interrupts the browsing experience and should be avoided.
Offline installation	Requires the user to download the plug-in file, quit the browser and launch the plug-in installation file. After installation is complete, the user may need to restart his or her computer for the changes to take effect. Users should be advised to read installation directions carefully before starting the download. This type of plug-in installation process requires more effort for the user. Users must re-establish their Internet connection and then revisit your site. Offline installation interrupts the browsing experience, and inexperienced users may be intimidated by this installation process. Consider this requirement when choosing multimedia.
Pre-installation	Some of the more common plug-ins are now pre-installed by browser manufacturers. This pre-installation benefits the Web developer as well as the user. However, it presents a new issue: determining which browser is used by your likely audience. Some of the plug-ins now commonly pre-installed in browsers are Flash Player and Acrobat Reader.

To streamline the process of presenting Flash multimedia to your users, it is good practice to use JavaScript to test the user's browser for the Flash Player plug-in. The Adobe Dreamweaver site-development application can write the necessary JavaScript code to determine whether the user has Flash capability. If so, the user sees the Flash presentation by default. If the user does not have Flash capability, the code can redirect the user to a designated alternative page (one that does not use Flash).

For more information about plug-in/viewer technologies, see *Web Design Specialist*, Lesson 28: Multimedia with Flash Professional CS5 — Movie Clips, and Lesson 31: Plug-Ins and Java Applets.

4.1.9: X/HTML and downloadable files

A browser will display any file it recognizes. If the browser does not recognize a particular file type, it looks for an associated plug-in that supports the file type. If one is not found, it prompts you to download that file. A downloadable file is one that users must save to their client systems before they can open it.

To create a downloadable file, you must link to a file type that is not natively supported by the browser or its plug-ins. When coding in X/HTML, you create a hyperlink to another Web page by using the anchor tag, Click Here . A browser requests the specified page from the server, then the page is displayed in the browser window.

Following is an example of a download hyperlink using a PDF file:

```
<a href="myNewsLetter.pdf"> Click Here To Download My Newsletter </a>
```

Two possible outcomes can occur when the user clicks this link:

- If the user has already downloaded and installed the Adobe Reader on his or her system, Adobe Reader will open the newsletter in the browser and add a new set of toolbars to work with the file.
- If the user has not installed the Adobe Reader, the browser will prompt him or her to download the file. The user must then access the Adobe Web site, and download and install the Adobe Reader to be able to view the file.

Whenever you provide a link to a downloadable file, you should also provide two more pieces of information: the file type users are going to download, and the file size so users know how long it will take. Even if users must download a file, this extra information makes the process easier for them

because they will know what to expect and how long it will take. You can also include the image for the associated file type. The previous example could best be portrayed on a site as follows:

 Click Here To Download My Newsletter (PDF 355KB)

As a Web developer, you cannot always be certain that your users have the proper plug-ins to view the file formats you use on your site. If you are using Adobe PDF files, for example, provide a link to the Adobe Web site so users can download Adobe Reader if they choose. Any other resources you think your users might need to view and navigate your site should be only a click away.

For more information about X/HTML and downloadable files, see *Web Design Specialist*, Lesson 31: Plug-Ins and Java Applets.

4.1.10: Rich media streaming ads

Rich media ads typically require some sort of plug-in for the Web user to be able to view them. Most frequently, rich media ads use Flash or other video or audio formats to present dynamic content that will catch the user's attention and persuade him or her to click through to the advertiser's site. Whereas traditional banner ads are limited to static text and images or basic animation using animated GIFs, rich media ads can feature user interaction, sound, vector animation and even streaming media.

Although they are more likely to annoy users than static ads or simple animated ads, advertisers have found that users are more likely to click on rich media ads than on traditional ads. Rich media ads are typically larger in file size than conventional online ads, and they take longer to download. Some rich media ads use streaming in order to minimize the user's wait time between beginning to download the rich media ad and playing it. Because the effectiveness of streaming depends upon the browser's ability to download data at a rate high enough to keep up with the playing media, advertisers must consider their users' possible bandwidth limitations even when designing rich media ads with streaming technologies.

You can minimize user inconvenience and annoyance (or the chance that your ad will be unusable by your target audience) by displaying rich media ads with other rich media content.

Rich media ads can be created using video production software, audio software, 3-D animation software, and other types of hardware and software devices. The actual technique and software used to create an ad is not an essential component of rich media. The final product, however, must be in a multimedia format supported by a widely deployed browser plug-in.

For more information and hands-on labs about rich media streaming ads, see *Web Design Specialist*, Lesson 31: Plug-Ins and Java Applets.

4.1.11: Java applet functionality and use

Java applets have often been used on the Web to provide animation or multimedia effects. However, applets are capable of much more. Other examples of uses for Java applets that can be embedded in Web pages include chat applications, secure e-commerce applications, dynamic menus, calculators, games, shopping applications and more.

Applets have a number of characteristics that make them ideal for adding functionality to Web pages, as discussed in the following table.

Applet Characteristic	Description
Small in file size	Because they are typically limited in functionality, Java applets are generally small in file size. Applets are often downloaded to a user's computer but can be run in the browser without downloading if the size is prohibitive.
Secure	Applets run within what is known as a sandbox. The sandbox model prevents Java applets from accessing your computer's file system or operating system directly. This prevents Java applets from harming anything on your computer.
Fast	Because Java applets are compiled, they run faster than scripts written in interpreted languages such as Perl or JavaScript.
Cross-platform compatible	Java applets do not need to be written differently for different computers, operating systems or Web browsers. One Java applet will work on any computer that supports Java.
Multi-threaded	Java support multi-tasking, also referred to as multi-threading. Multi-threading allows more than one thread of execution, in which each thread is a separate process within a document. Thus, Java does not make the user wait for one program to finish an operation before starting the next one. Stand-alone Java runtime environments and applets have excellent real-time behavior, meaning they work quickly and efficiently.
Client-side programs	Many server-side tasks can be sent to a Java-enabled Web browser that is running an applet, shifting the processing from exclusively server-side to client-side as well. Applets require very little space and operate efficiently on the client side.

You need two pieces of information to insert an applet into your Web page:

- The name of the Java applet class file that will perform the function
- The parameters that the class file needs to run

The class file is the Java applet, and the parameters are the X/HTML elements. Without each of these, the other is useless.

In HTML 4.0 and later, the proper way to embed an applet is to use the <object> tag. In previous versions, applets were embedded with the <applet> tag, and this method is still the most often used. All major browsers now support applet embedding using the <object> tag. However, some older browsers that are still in use may provide incomplete or buggy support for <object>. For this reason, it is important to always test your Web pages on a wide range of browsers.

The <object> tag identifies the name of the class file to load and run. Applets use the *classid* attribute to reference the location and type of object (in this case, a Java applet file). Java applet files always use the .class file name extension. The class file is the compiled Java code that runs the applet function. The *codetype* attribute tells the browser the type of object that is being embedded. Two other attributes required by the <object> tag for embedding applets are *height* and *width*. These attributes tell the browser how much space the applet needs to display. The parameters of the applet are passed to the class using the <param> tag. All <param> tags are placed between the opening and closing <object> tags. A <param> tag is required for each parameter. The *name* and *value* attributes are used with each <param> tag. The *name* attribute holds the name of the attribute (e.g., text color). The *value* attribute supplies the desired effect (e.g., red). Therefore, the text color would be red. Following is an example of code used to embed an applet:

```
<object classid="java:Fade.class" codetype="application/java" width="300" height="50">
    <param name="bgcolor" value="0000ff" />
    <param name="txtcolor" value="ff0000" />
    <param name="changefactor" value="5" />
    <param name="text1" value="Welcome to Habitat for Humanity" />
    <param name="url1" value="http://www.habitat.org" />
    <param name="font1" value="Helvetica,PLAIN,14" />
    <param name="text2" value="donate" />
    <param name="url2" value="http://www.habitat.org/support" />
    <param name="font2" value="TimesRoman,ITALIC,18" />
```



```
        <param name="text3" value="Come back soon!" />
        <param name="url3" value="http://www.habitat.org" />
        <param name="font3" value="Courier,BOLD,18" />
    </object>
```

Because there are still browsers in use that do not fully support the <object> element, and because many Web development programs still use the <applet> element, you should know how to use <applet>. The only difference between using <applet> and using <object> for embedding applets is that the <applet> element uses the *code* attribute instead of *classid*, and that <applet> does not need a *codetype* attribute.

One of the limitations of Java applets is that the whole applet file must be downloaded before it can begin execution. Currently, Java applets have no streaming capabilities. In addition, they do not have caching capability, which means that the user must download the applet again upon each subsequent visit to the site.

For more information and hands-on labs about Java applet functionality and use, see *Web Design Specialist*, Lesson 31: Plug-Ins and Java Applets.

Practice exam questions

Following are some practice exam questions intended for student review.

1. Which of the following describes the process of "tweening" when creating Flash SWF files?

a. Flash converts animated GIFs into Flash symbols for display in browsers.
b. You create each frame, and Flash plays the frames in rapid succession to display animation.
c. You convert objects to symbols in Flash, and this process displays in the browser as animation.
d. You provide a beginning and ending frame, and Flash creates transitional frames to display animation.

Correct response: d. You provide a beginning and ending frame, and Flash creates transitional frames to display animation.

Explanation: In tweened animation, you provide beginning and ending frame information, then Flash creates the frame values in between. This process minimizes file size compared to frame-by-frame animation because Flash stores only the values for the changes rather than all values for each entire frame. Flash refers to this animation or motion as "tweening."

2. You are creating a multimedia Web site that will require plug-ins. What should you provide on your site?

a. A list of required plug-ins and their prices
b. Links for users to download and install plug-ins
c. An order form users can submit to vendors to obtain the plug-ins
d. A mirror site that does not require the plug-ins but offers the same content

Correct response: b. Links for users to download and install plug-ins

Explanation: Your job as a Web developer is to shape the user's experience at your site. A critical aspect of the planning process is providing effective ways for first-time users to download and install plug-ins. As a Web developer, you cannot always be certain that your users have the proper plug-ins to view the file formats you use on your site. If you are using Adobe PDF files, for example, provide a link to the Adobe Web site so users can download Adobe Reader if they choose. Any other resources you think your users might need to view and navigate your site should be only a click away.

4.2: Client-side and server-side technologies

This subdomain includes skills and knowledge required to use client-side and server-side programming to enhance Web site functionality.

4.2.1: Client-side vs. server-side technologies

Server-side Web technologies put the burden of processing on the Web server, whereas client-side technologies make the browser or client computer do the work. Most server-side technologies require the developer to know some programming or scripting languages. Client-side technologies tend to require less advanced training.

Server-side technologies perform functions on the Web server based on client input, then return relevant X/HTML content. Handling processing on the server side eliminates some compatibility issues with browsers. However, server-side processing comes at a price: It slows down the server, especially on high-traffic sites whose servers interpret hundreds, even thousands of scripts simultaneously.

The following table describes some client-side and server-side technologies used for Web applications.

Technology	Client or Server Side?	Description
JavaServer Pages (JSP)	Server-side technology	Enables Web developers to include program instructions within their X/HTML pages that will run on the server when the page is requested by a Web browser. After the instructions in the page are processed, the server removes all the coding instructions it just processed, so that the client browser receives only X/HTML. Thus the technology is not reliant on browser type, yet allows real-time data to be sent. JSP can run on multiple platforms (as long as Java code processing is enabled). JSP was developed by Sun Microsystems.
Active Server Pages (ASP)	Server-side technology	Similar to JSP. ASP was designed to be used on Windows servers, although it can be ported to other platforms. ASP was developed by Microsoft.
Common Gateway Interface (CGI)	Server-side technology	Not a language; but a simple protocol that can be used to communicate between X/HTML forms and an application. CGI permits Macintosh, PC and UNIX computers to post data to or retrieve data from an HTTP server through a Web browser. This technology is very powerful but requires some programming knowledge to be used effectively. A number of languages can be used to write CGI scripts, including Perl, C or shell scripting. One of the most common uses of CGI is the guest book. CGI is not implemented when a browser opens a guest book page from a Web server; it is initiated upon submission of the page. The X/HTML form page specifies within it a path to the directory containing the CGI and the CGI file itself. Upon receipt of data, the server launches its interpreter and performs the function specified. After completion, the server returns data to the browser, confirming that the submission was successful. CGI can also be used to extract data from a database or file on the Web server.
JavaScript	Mostly client-side uses	An object-based scripting language. Although JavaScript can be used outside of a Web browser, it is most often used within X/HTML documents to add interactivity to Web pages without using server-based applications, such as CGI programs.
VBScript	Mostly client-side uses	Similar to JavaScript, VBScript is Microsoft's subset of the Visual Basic programming language. The VBScript visual approach might be easier to understand initially, but the language is only supported by the Internet Explorer browser. JavaScript works relatively smoothly with nearly every Web browser, and is therefore a better choice for client-side scripting.
JScript	Mostly client-side uses	The Microsoft implementation of the Netscape JavaScript language. Several minor differences exist between these two implementations of essentially the same language. However, these differences can cause problems. If you decide to learn JScript, be sure to test your code's execution in browsers other than Internet Explorer.

Table cont'd

Technology	Client or Server Side?	Description
ECMAScript	Client-side scripting language	In an effort to standardize, Netscape and Microsoft are moving toward the European Computer Manufacturers Association (ECMA) version of scripting language called ECMAScript. At this time, all major browsers are compliant with the ECMAScript standard. ECMAScript and JavaScript are essentially the same, and the terms ECMAScript and JavaScript are often used interchangeably.
ActionScript	Client-side scripting language	A language similar to JavaScript, implemented in the Adobe Flash development application. Provides Flash developers the ability to add user-triggered interaction to buttons and other elements in Flash animations. Compatible with ECMAScript.

Many GUI HTML development applications make it possible to write client-side or server-side code without knowing a scripting or programming language (such as Perl, JavaScript or VBScript). Examples of such applications include Dreamweaver, Expression Web and NetObjects Fusion. With Expression Web, you can build a Web site that accepts data from an X/HTML form, writes the data to a variety of file formats, then sends the data to a specified recipient via e-mail or writes it to a database dynamically. Before such applications were available, these tasks would have required you to write a custom script and configure the server to use Perl (or another language interpreter) to interpret the submission and format the output data. With applications such as Expression Web and its extensions on the HTTP server, you can perform these functions with a few clicks of the mouse.

For more information about client-side vs. server-side technologies, see *Web Design Specialist*, Lesson 26: Multimedia with Flash Professional CS5 — Timeline, Layers, Symbols and Buttons, Lesson 30: JavaScript and DHTML Fundamentals, and Lesson 32: HTTP Servers and Web Applications. To learn more about JavaScript, visit *www.javascript.com* and *http://javascript.internet.com*. To learn more about ASP, visit *http://msdn.microsoft.com/en-us/library/aa286483.aspx*. To learn more about JSP, visit the Sun Java site at *http://java.sun.com/products/jsp/*.

4.2.2: JavaScript objects, properties and methods

Programming and scripting languages use objects, properties and methods, which are described in the following table.

Programming Term	Description
Object	A programming function that models the characteristics of abstract or real objects. Objects are often grouped with similar objects into classes. In object-oriented and object-based languages, every object has attributes (properties) that characterize it and behaviors (methods) that it can perform.
Property	Properties represent various attributes of the object, such as height, color, font size, age and so forth.
Method	Methods are the behaviors or actions an object can be made to perform, such as a calculation, onscreen movement, or the writing of text.

You already use and change objects, properties and methods, although you may not have thought of them in these terms. Each time you load a Web page in your browser, you access the window and document objects. When you specify a background color for your page, you specify a property for that document object. When you click the Back and Forward buttons on your browser, you access the history method associated with the window object, allowing you to navigate among previously loaded pages.

The following table describes some basic JavaScript methods that you can write to communicate with your Web page users.

JavaScript Method	Description	Example Syntax
alert()	The *alert()* method allows you to communicate with the user by making a pop-up window appear with your message. To call the *alert()* method, you need only have a simple line of code, called a statement, in a script block somewhere in your document. If you have no other code in your script, the message will pop up every time the page is loaded. If your script has other code, the message window will pop up when it is called in the code sequence.	The syntax for the *alert()* method is as follows: alert("message");
prompt()	The *prompt()* method requests user input through a text field within a dialog box. Whatever the user types into the text box is returned as the result of *prompt()*. This result can then be used as an argument to another method. Note that the text to appear in the initial prompt and the response must each be enclosed within quotation marks. Also, the comma between messages is required to separate them.	The basic syntax for the *prompt()* method is as follows: prompt("Message to user", "Default response text");

open()	JavaScript gives you the ability to open new windows at will and populate them with existing information, information created on the fly (dynamically), or no information at all. You can open a new window by using the *open()* method.	The syntax to open a new window using the *open()* method is as follows:
		open("URL","WindowName", "FeatureList");
		For example, to open a new browser window showing the Yahoo! Search page, you could use the following statement:
		open("http://www.yahoo.com", "SearchWindow","toolbar=1, location=1,menubar=1,scrollbars=1,status=1,resizable=1");

The feature list in the *open()* method consists of a series of attributes that you specify as either displayed or not. The accepted values for each of these attributes are 1 (or "yes") if the attribute is to be displayed, and 0 (or "no") if it is not. The following table lists possible attributes of the *window* object.

Property of *window* Object	Description
toolbar	Creates the standard toolbar
location	Creates the location entry field
directories	Creates the standard directory buttons
status	Creates the status bar
menubar	Creates the menu at the top of the window
scrollbars	Creates scrollbars when a document grows beyond the current window
resizable	Enables window resizing by the user
width	Specifies the window width in pixels
height	Specifies the window height in pixels
top	Specifies the top *y* coordinate onscreen where the window will open
left	Specifies the left *x* coordinate onscreen where the window will open

The *navigator* object is part of the JavaScript object model. It allows access to information specific to the browser. Within the *navigator* object are several properties that can be tested. The following table lists some of the properties of the *navigator* object.

Property/Method	Description
appCodeName	Code name of browser
appName	Official browser name
appVersion	Version of browser (e.g., 3.x, 4.x, 5.x)
plugins	Plug-in installed in browser
userAgent	User agent header

The primary event is the first event that occurs on any Web page, which is the loading of the Web page into the browser. No user events can take place before the page is loaded in the browser.

For more information about JavaScript objects, properties and methods, see *Web Design Specialist*, Lesson 30: JavaScript and DHTML Fundamentals.

4.2.3: Using JavaScript

Web designers commonly target specific browsers that can support certain design features, then write code to take advantage of these browsers' added functionality. Therefore, if you determine which browser is visiting your page, you can use JavaScript to design certain elements for the page. Similarly, by determining which plug-ins a browser has, you can include content that the plug-in supports. JavaScript can be used to detect browser type and installed plug-ins. The following JavaScript example uses the JavaScript *navigator* object to identify various properties of a browser:

```
</head>

<script language="JavaScript" type="text/javascript">
<!--
function yourBrowser() {
document.details.Name.value=navigator.appName;
document.details.Version.value=navigator.appVersion;
document.details.Code.value=navigator.appCodeName;
document.details.Agent.value=navigator.userAgent;
}
// -->
</script>

<body bgcolor="#FFFFFF">
```

Although you can write JavaScript to perform this function transparently, your users might be interested to see this functionality in action.

Suppose you want to add to some browser-specific functionality to your Web pages. To ensure that all Web users can access your site regardless of the browsers they use, you need the ability to test and identify your users' browsers. You can write JavaScript to perform this function transparently, then provide each user with the appropriate content based on which browser he or she is using to view your site. Using a sniffer and redirection in this way makes your Web site more accessible to more users while allowing you the option to incorporate browser-specific functionality if you choose. A sniffer is a code application that determines status. In the following example, the sniffer code will direct the user based on the browser in use:


```
<head>
<script language="JavaScript" type="text/javascript">
<!--

function checkBrowser() {
    var name=navigator.appName;

    if (name.indexOf('Netscape') != -1) {
            document.location.href="firefox.htm"

    } else {
    if (name.indexOf('Microsoft') != -1) {
            document.location.href="microsoft.htm"
            }
    }
}
//-->
</script>
</head>
```

Web developers commonly use rollover elements on their pages. A rollover occurs when a user moves his or her cursor over an image and the image changes. A minimum of two images are involved in a rollover. The first image is the one that the user sees when the page initially loads. The second image is seen when the user moves the cursor over the image and it changes. The problem that Web developers face is that when the mouse event (mouseover) occurs, the browser must replace the current image with the second image. Unless the second image is preloaded into the cache, a lag time exists and the image will not change until the browser downloads the second image from the server. To prevent this lapse from occurring, the images must be preloaded into the cache. The JavaScript *Image* object accomplishes this task by defining the location of the required images. Before the preloading begins, a simple condition is tested to see whether the browser understands the *Image* object. If so, then the preloading begins; otherwise, nothing happens:

```
<head>
<title>Image Preloader</title>
<script language="JavaScript" type="text/javascript">
<!--

if (document.images) {
    one = new Image
            one.src = "images/getinvolved.gif"
    two = new Image
            two.src = "images/howitworks.gif"

    oneOver = new Image
            oneOver.src = "images/getinvolved_over.gif"
    twoOver = new Image
            twoOver.src = "images/howitworks_over.gif"
}
//-->
</script>
</head>
```

For more information and hands-on labs about using JavaScript, see *Web Design Specialist*, Lesson 28: Multimedia with Flash Professional CS5 — Movie Clips, and Lesson 30: JavaScript and DHTML Fundamentals.

4.2.4: JavaScript dot notation

Dot notation shows the relationships between objects and their parts by listing them from general to specific, separating each part name by a period (or dot). Dot notation is used to associate an object's name with its properties or methods. The syntax for JavaScript dot notation is as follows:


```
objectName.objectProperty
objectName.objectProperty = value

objectName.objectMethod()
objectName.objectMethod(argument1,argument2,...)

parentObjectName.objectName.objectProperty
parentObjectName.objectName.objectMethod
```

For example, a statement to close a window uses dot notation to associate the *close()* method with the *window* object named *myWindow*, as follows:

```
onClick="myWindow.close()"
```

Dot notation is also used to show the hierarchical relationship between objects. For example, you can use the *document.write()* method to write text to the screen. Strictly speaking, the method could also be written as *window.document.write()*. It is not incorrect to write code in this manner, but it is ultimately less efficient. Because *window* is the default object, you need not write it out; in this case, the *window* object is the parent object and also the default object.

For more information about JavaScript dot notation, see *Web Design Specialist*, Lesson 30: JavaScript and DHTML Fundamentals.

4.2.5: Rollover images with scripting technology

Web developers commonly use rollover elements on their pages. A rollover occurs when a user moves his or her cursor over an image and the image changes. A minimum of two images are involved in a rollover. The first image is the one that the user sees when the page initially loads. The second image is seen when the user moves the cursor over the image and it changes.

Suppose you are creating a Web page with some rollover effects that use several fairly large and detailed images. The number and file sizes of the images are causing your page to download somewhat slowly, and you also notice that the rollovers do not work right away. Because this pause in functionality spoils the effect you want to display, you can incorporate the JavaScript *Image* object to define the location of the required images and preload them in the browser. The following code shows an example of image preloading script in the <head> section:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Image Preloader</title>
<script language="JavaScript" type="text/javascript">
<!--
if (document.images) {
    one = new Image
            one.src = "images/getinvolved.gif"
    two = new Image
            two.src = "images/howitworks.gif"

    oneOver = new Image
            oneOver.src = "images/getinvolved_over.gif"
    twoOver = new Image
            twoOver.src = "images/howitworks_over.gif"
}
//-->
</script>
</head>
```

You can use Adobe Dreamweaver to apply a rollover behavior to an image using scripting technology. Dreamweaver provides a useful dialog box designed specifically for rollover creation. You simply instruct Dreamweaver as to which images you want to use and where you want the links to point. Dreamweaver automates the rest of the process and writes DHTML code for you to create the rollover function. Whether

you know scripting technologies or not, you can use Dreamweaver to create effects that use these technologies. You simply specify the effects you want and the page elements to which you want them applied — and Dreamweaver writes the script code for you. This feature offers you more advanced design options for your pages without significantly increasing the scope of your Web project.

For more information and hands-on labs about rollover images with scripting technology, see *Web Design Specialist*, Lesson 22: Site Development with Dreamweaver CS5 — Advanced Features, and Lesson 30: JavaScript and DHTML Fundamentals.

4.2.6: Dynamic HTML (DHTML)

Dynamic HTML (DHTML) is a set of technologies that allows the Web author to create more interactive pages. DHTML is essentially the ability to use script to change a Web page without reloading it. DHTML is not a single technology but a term used to refer to a group of interrelated technologies. DHTML is made possible through the use of script (JavaScript or VBScript, for example), the Document Object Model (DOM), and two specifications that work together: X/HTML and Cascading Style Sheets (CSS).

These standards extend X/HTML by adding new methods for formatting text and X/HTML page elements, as well as new scripting event handlers that can trigger script with a wider variety of mouse and keyboard events. With DHTML, any X/HTML element (including plain paragraphs) can be made to change after the page has loaded. The entire page is considered "active" and awaits user input, which can be controlled by script. X/HTML elements constitute programmable objects on a Web page. X/HTML objects can be referenced by script through the use of the *ID* property. The various attributes provided by X/HTML and the CSS specification constitute programmable properties.

Coupled with the CSS standard and linked by script, DHTML can be used in some cases to replace functions currently performed by Java applets, ActiveX controls or animated GIFs. DHTML can be described as X/HTML with a scripting language's ability to interact with the tags. In addition, DHTML can often serve as an alternative to some server-side technologies such as Active Server Pages (ASP) and CGI. DHTML provides dynamic content to the browser rather than the typical static X/HTML, but without the need to access the server each time it receives browser input.

Unlike X/HTML or XML, DHTML is not a W3C standard, but instead simply a term that has been used to describe the combination of several standards that make it possible to manipulate the elements in a Web page dynamically. Although the newer versions of Web browsers are all moving toward a single standard version of DHTML, enormous differences still exist in the way DHTML is implemented in older browsers. The result is essentially three different variations of DHTML: the Netscape 4.x version, the Internet Explorer 4 version, and the W3C's standard DOM. The number of pre-W3C DOM browsers in use today is small, but you still need to test any DHTML code on each of the major browser types (i.e., Internet Explorer and Firefox). Depending on whether your pages use browser-specific code, you may need to use code for detecting the user's browser type and redirecting some browsers to display the correct page.

Today, it is becoming increasingly possible to write DHTML code that will work in any browser. But despite the move toward the adoption of standards, differences still exist in browsers that will make some types of DHTML code work differently or not at all in some browsers. As a Web developer, you must always consider such accessibility issues. One option is to avoid using DHTML in your site pages. Or you could provide alternative content or technologies for users who cannot view your DHTML. You could also provide cross-browser DHTML customized for each of the three DHTML types (Netscape, Internet Explorer and newer browsers including Firefox). You first test each user's browser to detect its type, then direct the browser to one of three different versions of your DHTML code, whichever is appropriate for the detected browser type. The following JavaScript *if/then/else* statement will accomplish this direction (note the specific code for each version is not given):


```
if (document.getElementById) {
    IE5+ and NN6+ code
}
else if (document.layers) {
    NN4 code
}
else {
    IE4 code
}
```

However, because of all the possible variations, the best way to ensure that your code is cross-platform compatible today is to use the DHTML tools built in to many GUI HTML applications, such as Dreamweaver and Expression Web, and to fully test every page you create in as many browsers as you can.

For more information about Dynamic HTML (DHTML), see *Web Design Specialist*, Lesson 30: JavaScript and DHTML Fundamentals.

4.2.7: XML vs. HTML and XHTML

Extensible Markup Language (XML) is a long-established specification that enables users and programmers to exchange data in a document. XML allows you to effectively create custom data structures for use in documents and in programs. Using XML, you can create documents and programming that contain more specific information about content than ever before, adding a certain level of "intelligence." XML was created to function transparently, providing machines with information about documents that will facilitate decision-making at the machine level.

XML is derived from the Standard Generalized Markup Language (SGML). Both SGML and XML are metalanguages, which are languages for creating other languages. Many people think XML is an addition of tags to HTML. This assumption is incorrect. Unlike HTML, XML is not an application of SGML but a subset of SGML. As such, XML qualifies as a metalanguage and can be used to write other languages, thus allowing for an infinite number of tags. XML offers users the ability to define their own set of markup tags — to write their own version of a markup language, so to speak. HTML is limited to the finite number of tags that are specified by the HTML recommendations.

XML, on the other hand, offers far more freedom. Web page developers can now create tags named anything at all. The following is an example of XML markup:

```
<greeting> Hello, World! </greeting>
```

"Hello, World!" becomes not just two words and some punctuation, or even a heading in a document, but specifically a greeting. Because XML is not a defined language but a metalanguage, it has no predefined tags to use. You have the ability to create any tags you desire.

However, there is a price for this freedom. In HTML, a user can often omit the ending </p> tag, for example, because the browser can usually infer from other tags where the next block of data begins. But with XML, the browser has no way of knowing when the greeting ends without the closing </greeting> tag. For this reason, the requirements for XML are stricter than those for HTML. XML requires developers to follow a set of rules for "well-formed" documents. This strictness may frustrate page developers, but it will be worthwhile. The more specifically the markup relates to the content, the more powerfully that content can be used.

HTML presents severe limitations to page authors, but it will continue to be the appropriate language for documents not intended for future use. For data that needs persistence over time, XML will better serve the purposes of retrieving and reusing that data. XML will redefine the way we code HTML. In the short run, XML will be used in conjunction with HTML. As browsers develop and style technologies progress, XML may be used without HTML. But HTML and XML are separate and different, and one technology is not necessarily better or worse than the other. Both are tools that have appropriate and inappropriate uses. The more you know about each, the better equipped you are to choose the best tools to meet your enterprise's needs.

Extensible Hypertext Markup Language (XHTML) 1.0 is the latest version of HTML. XHTML combines characteristics of HTML and XML to create a transition from HTML toward XML, while still providing current browsers with backward-compatibility. This means that the XHTML developer who adheres to the requirements of XML well-formedness can use HTML tags and XML tags within the same document and have it be valid.

For more information about XML vs. HTML and XHTML, see *Web Design Specialist*, Lesson 13: XML and XHTML. For more information about XML visit the W3C site at the following URL: *www.w3.org/XML/*.

4.2.8: Well-formed XML documents

Well-formedness is essential in XML. Documents that are not well-formed will not load in the browser, according to the XML Recommendation. The W3C instructs us that "violations of well-formedness constraints are fatal errors." Therefore, the page will not appear in the browser at all unless this minimum requirement is met.

For reasons of forward compatibility, the first element in an XML document specifies the version of XML to which the document conforms. This element is the XML declaration. Although the opening <?xml?> declaration statement is considered optional, it is highly recommended for forward compatibility. As the language evolves, it will be helpful to future browsers to know which version of the XML specification was used to create the document. If used, this declaration must be all lowercase. The following example begins with an XML declaration providing version information:

```
<?xml version="1.0"?>
<greeting> Hello, World! </greeting>
```

Some other basic rules will help you construct well-formed XML documents:

- Tags must be explicit; they cannot be inferred. All opening tags must have corresponding closing tags, and all closing tags must have corresponding opening tags.
- Empty tags require a forward slash (/) character before the closing angle bracket.
- All attribute values must be enclosed in quotation marks.
- The document must have a root element that encloses all other tags, and all tags must be properly nested.
- Tags are case-sensitive and must match each other in every implementation.

If an XML document conforms to these strict syntax rules, it is said to be "well-formed." An XML document is said to be "valid" if it is well-formed and also contains a Document Type Definition (DTD). A DTD is a set of rules contained in a simple text file that defines the structure, syntax and vocabulary as it relates to tags and attributes for a corresponding document. The DTD defines the validity of all subsequent tags. The DTD is often the document author's own creation, and thus must be declared if it is to be referenced.

Note: Do not confuse the XML Document Type Definition (DTD) with the document type declaration or <!DOCTYPE> tag used at the beginning of XHTML and XML files for identification purposes.

Following is an example of a DOCTYPE declaration that could reference your own DTD for internal letter documents:

```
<?xml version="1.0"?>
<!DOCTYPE letter SYSTEM "InternalLetter.dtd">
<letter>
</letter>
```

The XML DTD defines the meanings and structure of all XML tags. Remember that XML allows you to create a language that describes your text. XML is not a specified language, like HTML. Therefore, you

must think ahead and define every element. Alternatively, you can reference established DTDs or the XML schema to define your documents' language.

For more information about well-formed XML documents, see *Web Design Specialist*, Lesson 13: XML and XHTML. For more information about XML, visit the W3C site at *www.w3.org/XML/*.

4.2.9: Secure XML

XML is increasingly being used to transfer data between servers and programs. XML data is designed to be very simple to read and understand. However, suppose you want or need to restrict the users who are able to read your XML data. For example, a bank and a merchant may need to exchange private XML data about a transaction. If the data is sent "in the clear" (i.e., as plaintext), then anyone who has the ability to intercept HTTP traffic between the servers could read the transaction data.

Secure XML is XML data that is encrypted before being transmitted between computers, and that is transmitted using digital signatures to ensure that the person or machine sending the XML data is actually the person or machine it claims to be.. The W3C standard for encrypting XML data is called XML Encryption Syntax and Processing. The standard for XML digital signatures is XML-Signature Syntax and Processing.

For more information about Secure XML, see *Web Design Specialist*, Lesson 32: HTTP Servers and Web Applications. To learn more about the upcoming XML Encryption Syntax and Processing standard, you can read the current specification at *www.w3.org/TR/xmlenc-core/*. You can read more about the XML-Signature Syntax and Processing standard at *www.w3.org/TR/xmldsig-core/*.

4.2.10: Web form processing with CGI

Common Gateway Interface (CGI) is a simple protocol that can be used to communicate between X/HTML forms and an application. CGI permits Macintosh, PC and UNIX computers to post data to or retrieve data from an HTTP server through a Web browser.

CGI is initiated upon submission of a user's Web page (e.g., a form) to a Web server. The X/HTML form page specifies within it a path to the directory containing the CGI and the CGI file itself. Files used for CGI commonly use the .cgi or .pl file name extension, and typically reside in a "cgi-bin" or "scripts" directory that permits "execute" privileges. Upon receipt of the data, the server launches its interpreter and performs the function specified. After completion, the server returns data to the browser, confirming that the submission was successful. Although returning a confirmation page to the browser is not required, it prevents confusion for the submitter of the data, who might otherwise submit his or her data more than once.

Suppose your Web site has a form that requests your users to submit some information in order to receive your catalog. You can use a CGI script to process the information your users submit to your site and then send a confirmation page to your users showing the information you received. This process not only helps you collect and maintain the data submitted by your site users, but also provides your users with feedback to let them know you received their information. Providing your users with feedback is an important component of good Web design. The following code shows an example of an XHTML form using CGI (the destination in the <form> tag) to process the form input:

```
</body>
<form method="post" action="http://ss1.ciwcertified.com/cgi-bin/process.pl">
<p>
Enter your first name: <input type="text" name="FNAME" size="20" />
</p>
<p>
Enter your e-mail address: <input type="text" name="ADDRESS" size="20" />
</p>
<p>
Choose as many catalogs as you like:<br />
<input type="checkbox" name="CATALOG1" value="Places" /> Places to Visit <br />
```



```
<input type="checkbox" name="CATALOG2" value="Schools" /> Schools to Attend <br />
<input type="checkbox" name="CATALOG3" value="Skills" /> Skills to Learn <br />
</p>
<p>

<input type="submit" value="Send your request" />
<input type="reset" value="Clear form" />

</p>
</form>
</body>
```

For more information and a hands-on lab about Web form processing with CGI, see *Web Design Specialist*, Lesson 32: HTTP Servers and Web Applications.

4.2.11: Site functionality with cookies

HTTP is a stateless protocol, which means that it transmits information without first establishing a connection between two parties. The result of this statelessness is that complex applications such as e-commerce (especially shopping carts) require a Web server to have some way to identify individual users of the site in order to provide each user with a continuous session of activity involving more than one page. The solution is cookies.

Cookies are small text files that have a variety of uses. Cookies are sent from a server to the user's browser in the HTTP response header. The browser accepts the response from the page assigning the cookie, then accepts (or gives the user the chance to accept or reject) the cookie. After the cookie has been accepted, the browser program stores the cookie information in a file on the user's system. When an HTTP request is generated by the browser, any cookies on the user's system with a path and domain name matching the current HTTP request will be passed along in the request header. The server can then test for and evaluate the contents of the incoming cookies.

Cookies store name-value pairs as text strings. Cookies can expire as soon as the user exits his or her browser, or they can be set to expire on some future date. For this reason, cookies are formally known as *persistent client state HTTP cookies*. They persist until they expire or are deleted by the user. A cookie header appears to your browser as follows:

Set-Cookie: name=value; expires=date; path=path; domain=domain; secure

Each cookie header contains a set of parameters. Not all these parameters must be assigned to use cookies, nor must they be assigned in the following formats. The cookie header parameters include the following. The parameter values written in italic type are variable information specified at the time the cookie is set.

Cookie Parameter	Description
name=*value*	The only information required to generate a cookie. All other cookie attributes are optional.
expires=*date*	Determines when the cookie will expire. If this attribute is not present, the cookie will expire at the end of the browser session. This attribute can also be used to overwrite and expire cookies that currently have future expiration dates by reassigning them an expiration date in the past. The date is formatted as follows (where GMT stands for Greenwich Mean Time): *weekday, DD-MM-YY HH:MM:SS GMT*

Table (cont'd)

Cookie Parameter	Description
path=*path*; domain=*domain*	The path portion and domain name of the URL that issued the cookie. When an HTTP request header is formed, the path and domain name pairs are checked; if they match the page being requested, any cookies pertaining to that path (and higher) are passed back to the server for evaluation. A cookie in a request header appears to the server as follows: *Cookie: name=value; name2=value2; etc.* The server does not see any information other than the name=value pairs. Only URLs matching the path and domain values in the cookie file can be read again and evaluated by the server.
secure	If this parameter is present, the cookie is sent only when a secure protocol such as Secure Sockets Layer (SSL) is present. The *secure* parameter does not specify a value; either it appears or it does not.

One of the most common uses of cookies is to store information about a user to maintain state. "State maintenance" means keeping track of a process. An e-commerce Web site (or any Web site that needs to track users) will write a cookie to your computer when you first arrive at the site. This cookie will contain a piece of information, such as a user ID, that uniquely identifies you to that server. At any time during your visit, the server can then check the cookie on your browser and customize the information it sends you based on your previous actions at that Web site. For example, when you add an item to a shopping cart on a Web site, the Web site stores your unique identifier in a database along with the item in your cart. When you click the button to view your shopping cart, the Web server checks the cookie on your computer, looks up your record in the database, and displays the item you placed in your cart earlier.

Following are some other ways that cookies can be used to enhance Web site functionality.

- Cookies can track the sites you have browsed.
- Cookies can store personal information. However, they store only information that you provided to the server that wrote the cookie. An example is address information that you submitted to a site in a Web-based form.
- Cookies are not encrypted; however, a server can encode a cookie so that it is unreadable except to that server. Secure cookies are encrypted only because all data transmissions across a secure connection are encrypted.
- Cookies can be used to target you for marketing and banner advertisements.

When developing your own Web sites, you may want to have the ability to track your site users and their preferences. If you run an e-commerce site, you will need this ability to enable a shopping cart and provide users with an organized, efficient and personalized shopping experience. One way for your Web site to set a cookie on a visitor's system is by using JavaScript.

For more information and a hands-on lab about site functionality with cookies, see *Web Design Specialist*, Lesson 32: HTTP Servers and Web Applications.

4.2.12: Pop-up/pop-under windows

Every Web browser has the ability to display multiple windows and to create new windows, and this functionality can be controlled programmatically. This ability is necessary (for example, so that you can access more than one Web site at a time). However, the ability to launch additional windows is often abused by programmers to force users to view things (usually advertisements) that they did not request.

These new browser windows are called pop-ups or pop-unders (depending on whether they appear above or below the browser window that launched them). Pop-ups are fine if the user requests the content and if

they complement the content of the main browser window. For example, a link to a movie or audio clip may be labeled, "Click here to launch this clip in a separate window."

However, pop-ups that appear unexpectedly and present advertising or other unwanted content to the user tend to be annoying. Another reason to avoid using pop-ups is that many browsers now feature pop-up blockers, which prevent pop-ups from appearing as the user browses sites. As a result, pop-ups and pop-unders are much less effective as advertising today than they were when they were new, and using them for legitimate purposes may exclude many users from some valid functionality of your site.

Consider your Web site's purpose and audience before adding pop-up and pop-under windows to it. If pop-ups will serve a legitimate and useful purpose in your site, you can create them using a scripting language such as JavaScript. The JavaScript *alert()* method allows you to communicate with the user using a pop-up window. To call the *alert()* method, you need only have a simple line of code, called a statement, in a script block somewhere in your document. The syntax for the *alert()* method is as follows:

```
alert("message");
```

The text to appear in the alert must be enclosed within quotation marks. If you have no other code in your script, the message will pop up every time the page is loaded. If your script has other code, the message window will pop up when it is called in the code sequence.

For more information about pop-up/pop-under windows, see *Web Design Specialist*, Lesson 7: Browsers, and Lesson 30: JavaScript and DHTML Fundamentals.

4.2.13: CAPTCHA

A CAPTCHA is an automatically generated graphic presented to a user who has just submitted information or otherwise made a request of a Web server. After the user provides the correct response to a CAPTCHA, then the user's input is accepted for processing. The term CAPTCHA is an acronym for Completely Automated Public Turing Test to Tell Computers and Humans Apart.

A classic application of CAPTCHA is for protecting Web site registration from automated programs. Use of CAPTCHAs in blog sites can ensure that only humans can enter comments on a blog. A free CAPTCHA implementation is available on the Web at *www.snapshot.com*. When implementing a CAPTCHA, it is recommended that you ensure that the CAPTCHA is accessible. For example, if the CAPTCHA relies only on reading text, then you may make your site inaccessible to people who are visually impaired.

For more information about CAPTCHA and a hands-on lab for creating a CAPTCHA for a Web site, see *Web Design Specialist*, Lesson 7: Browsers.

4.2.14: TinyURL

Short URL aliases are useful because they are easier to remember and type. They are also easier to use than long URLs in situations in which space may be limited, such as in e-mail signature lines and microblogs, and with e-mail clients that create line breaks after a certain number of characters are entered.

TinyURL is a free Web service that provides short aliases for long URLs, aliases that will not break in e-mail postings and will never expire. The TinyURL home page (*http://tinyurl.com/*) includes a form in which you can enter a long URL for shortening.

A potential disadvantage to using TinyURLs is that they are subject to linkrot. If the URL of the original Web page changes, all URLs related to the short URL alias will become invalid.

For more information and a hands-on lab on creating an alias with TinyURL, see *Web Design Specialist*, Lesson 7: Browsers.

Practice exam questions

Following are some practice exam questions intended for student review.

1. Which JavaScript method requests and captures user input through a text field within a dialog box?

a. The alert() method
b. The open() method
c. The prompt() method
d. The window() method

Correct response: c. The prompt() method

Explanation: The JavaScript prompt() method requests user input through a text field within a dialog box. Whatever the user types into the text box is returned as the result of prompt(). This result can then be used as an argument to another method. The prompt(), alert() and open() methods are all methods of the window object in JavaScript.

2. Which of the following is a required element of a well-formed XML document?

a. Tags must be implicit.
b. Tags are case-sensitive and must match.
c. Attribute values must be enclosed in parentheses.
d. Empty tags require a forward slash (/) before the opening angle bracket.

Correct response: b. Tags are case-sensitive and must match.

Explanation: Some basic rules will help you construct well-formed XML documents: Tags are case-sensitive and must match each other in every implementation. Tags must be explicit; they cannot be inferred. All attribute values must be enclosed in quotation marks. Empty tags require a forward slash (/) character before the closing angle bracket.

4.3: Web databases

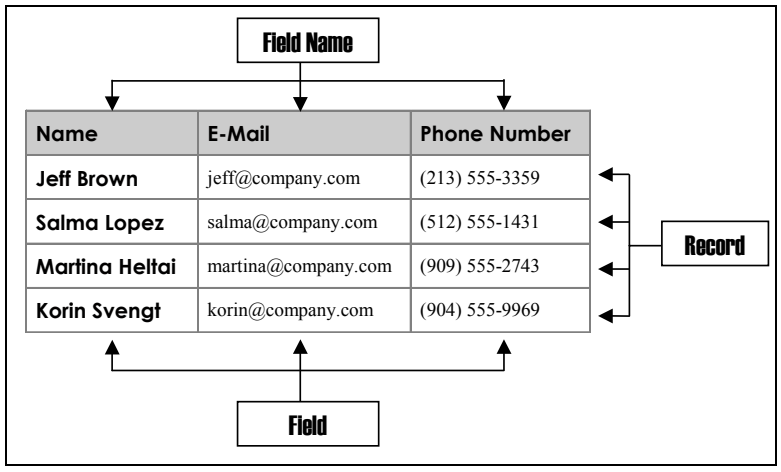
This subdomain includes skills and knowledge required to connect Web pages to a database.

4.3.1: Database elements

Databases provide information storage. The information can be retrieved, in whole or in part, by a program designed to accept requests from users. A database can be housed in a single location, such as a PC or mainframe. It can also be constructed from multiple files and housed at different locations on the enterprise; this structure is known as a distributed database.

Traditional databases store and organize information in fields, records and files. This strategy is useful for analyzing information from different points of reference. It also provides a powerful numerical analysis tool.

Schema (pronounced "skeema") is the structure of a database system and often depicts the structure as a graphical reference. The schema defines tables and fields, and the relationships between them. All database information is contained in tables. A table is a repository of information divided into columns and rows, as shown in the following figure.



The table consists of fields (columns) and records (rows). A record includes a complete set of information, such as name, e-mail address and phone number for one person. A set of records is referred to as a file. The table also has field names by which information can be sorted and retrieved, as it would when outputting a list of all phone numbers in the table, for example.

For more information about database elements, see *Web Design Specialist*, Lesson 33: Databases.

4.3.2: Database query types

The information stored in a database can be accessed using a query. You can think of queries as questions from the user to the database table. The query returns a value; this reply to the query provides information stored in the database. This value can be static (e.g., a list of phone numbers) or used in another query (e.g., all phone numbers beginning with 555). This value can also be used as a scripting formula variable. You make queries using the following general query types.

Database Query Type	Description
Menu query	The user is offered a list of options from which to choose.
Query by example	The user states which fields and values are to be used in the query.
Query language	A specialized language called Structured Query Language (SQL) is used to retrieve or manipulate information in the database.

Structured Query Language (SQL) (pronounced "see-kwell") was developed by IBM in 1974 and was originally called Structured English Query Language (SEQUEL). Oracle Corporation released the first commercial SQL application in 1979. Since then, SQL has become a standard language for accessing information from a database.

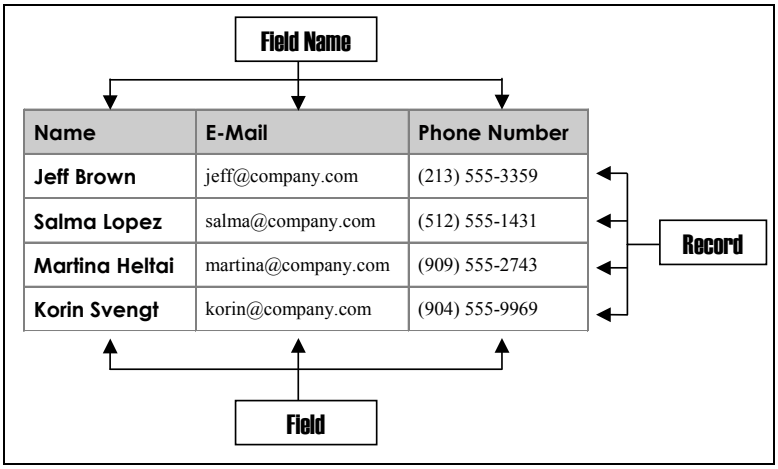
There are several types of SQL queries, which manipulate the database in different ways. The four main types of SQL queries are as follows.

SQL Query Type	Action
SELECT statement	Retrieves information from the database
INSERT statement	Creates new database record
UPDATE statement	Modifies existing database record
DELETE statement	Removes record from a database table

The basic structure of a SQL query is as follows:

```
[query type] [column name] FROM [table name] WHERE [conditions to be met]
```

For example, consider the database table named *Customers* shown in the following figure.



In this example, suppose you wanted to retrieve Jeff Brown's e-mail address from this database table. You would use the following query:

```
SELECT E-Mail FROM Customers WHERE Name = "Jeff Brown"
```

The database would respond to this query by returning the value of the E-Mail column from the record in which the name is equal to *Jeff Brown*.

For more information and a hands-on lab about database query types, see *Web Design Specialist*, Lesson 33: Databases.

4.3.3: Database Management System (DBMS) types

Database queries require a database management system (DBMS). A DBMS is an application that allows users to manipulate information in the database. These systems vary widely and include the following types.

DBMS Type	Description
Flat-file DBMS	Stores information in a single table consisting of multiple rows and columns.
Relational DBMS (RDBMS)	Stores related information in a collection of tables. The data in each field can be related to any field of any table, and a field can be related to many fields or many fields can be related to a single field. New information can form a new table, in which each field can in turn be related one-to-one, one-to-many or many-to-many. Because queries to a relational database can create new tables, a relational database requires the ability to grow significantly to handle the demands of new input.
Multidimensional DBMS	Uses common field values and stores information in organized groups of records. This type of DBMS is usually generated from an RDBMS.

The most widely used type of DBMS is the relational DBMS. A DBMS gives users and administrators more control over additional aspects of the database, including organization, security, data integrity and retrieval. It also allows an organization to expand or change its database structure with minimal disruption to the existing system.

For more information about Database Management System (DBMS) types, see *Web Design Specialist*, Lesson 33: Databases.

4.3.4: Adding site search capability

A site search form is very useful to your visitors, and thus should be implemented when possible. You can add a search form by using a server-side scripting language such as ColdFusion, ASP, JSP, PHP or Perl.

There are also tools available on the World Wide Web that you can use to create a site search form. Online sources include Google's Custom Search Engine (CSE) and Wrensoft's Zoom Search Engine. Your site must be published to the Web in order for you to create a site search form that is usable.

For more information and a demonstrative lab about adding site search capability, see *Web Design Specialist*, Lesson 22: Site Development with Dreamweaver CS5 – Advanced Features.

4.3.5: Database information types

Databases provide information storage. The information can be retrieved, in whole or in part, by a program designed to accept requests from users.

The data can be stored in the database as various information types: plaintext, X/HTML, XML, images, or any combination of these formats. In addition, an online store may also keep a database containing dynamic inventory information, which can inform the user whether the product being viewed is currently in stock.

Hypertext databases store information as objects. This format is useful for storing different types of information, such as text, images and multimedia (video or sound files). All data within this database is treated as objects; the data type is irrelevant. Any object within this database can be linked to any other object.

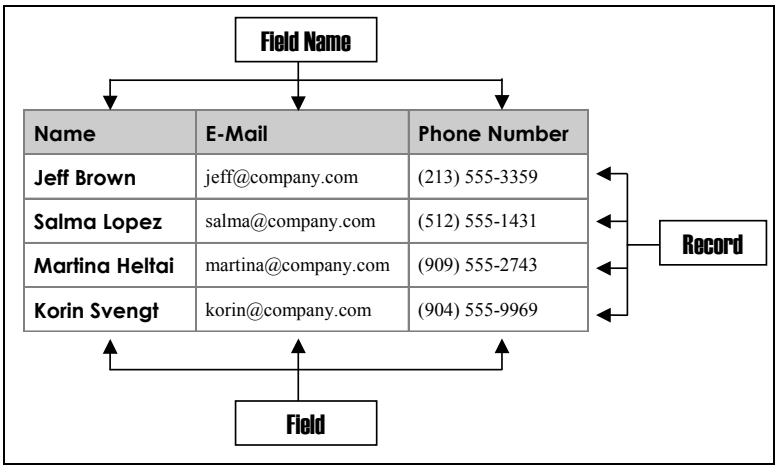
For more information about database information types, see *Web Design Specialist*, Lesson 33: Databases.

4.3.6: Connecting Web pages to databases

Data-driven Web sites are sites that use a server-side program to combine data from a database with page templates to generate the X/HTML pages that a Web site visitor sees. For example, the product pages on e-commerce Web sites are usually data-driven. In this case, data about each product is generally stored in a database table and combined with a product page template to create the X/HTML page. The basic structure of the page stays the same while different content appears in a specified area of the page. The page templates offer a consistent look and feel, with the database providing product information to the page based on user requests. This provides the online store with organizational and searching capability that allows users to query for specific products, and to receive descriptions, pictures and inventory information about items they might purchase.

Regardless of the language in which they are written, data-driven Web sites typically use SQL queries to manipulate a database. The programming language used on the site (such as Java, ASP, PHP, Perl or ColdFusion) simply creates a connection to the database so that SQL queries can be sent and the results can be used to build Web pages.

For example, consider the database table named *Customers* shown in the following figure.



The following code shows a simplified example of the way an Active Server Pages (ASP) file uses a template and a SQL query to create a data-driven Web page using the example *Customers* database table.

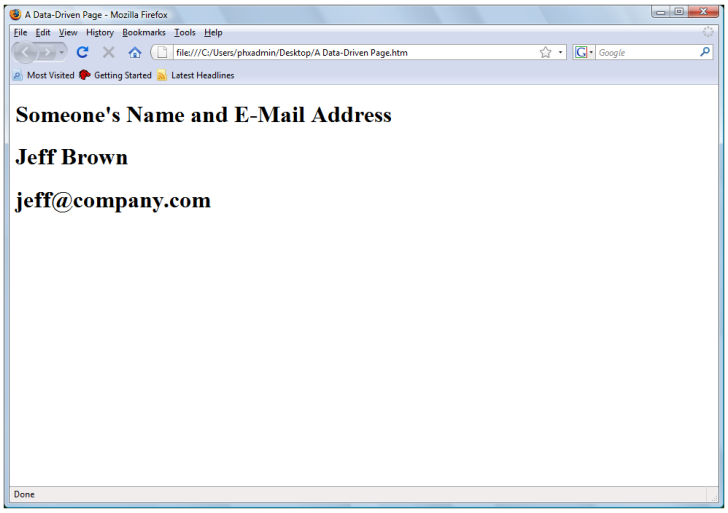
(*Note: This code is advanced and not required knowledge for the exam. However, consider what you have learned about dot notation, and you may be able to glean a basic understanding of this process.)*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>A Data-Driven Page</title>
</head>

<body>
<h1>Someone's Name and E-Mail Address</h1>

<!--Note: ASP code is written inside of special brackets (<% %>) in order
to distinguish it from XHTML code. Comments in ASP begin a line with a single quote.-->

<!--The data-driven part of the page starts here-->

<%
'Create a connection to the database
Set conn = Server.CreateObject("ADODB.Connection")
```



```
conn.Open myDatabase

'Create a SQL query
sql = "SELECT Name,E-Mail FROM Customers WHERE Name = 'Jeff Brown'"

'Create a recordset to hold the results
set rs=server.createobject("adodb.recordset")

'Query the database (using the SQL query), put the results in the recordset

rs.open sql,conn

'Display the results of the query
%>

<h1><%Response.write("rs.Name")%></h1>
<h1><%Response.write("rs.E-Mail")%></h1>

<!--The data-driven part of the page ends here-->
</body>
</html>
```

Note that if you changed the name in the conditional part of the SQL query from "Jeff Brown" to "Salma Lopez," for example, you could find Salma Lopez's e-mail address and display it in this same page template. The first header ("Someone's Name and E-Mail Address") would not change. The name and e-mail address that are displayed in the data-driven portion of the page would change based on the specific record you requested in your query. When this example ASP page is accessed on a Web server that supports ASP and that can connect to the example database table given, the resulting page would resemble the following figure.



Pages featuring dynamic content are common today, and it is likely that you will need to display database data on one or more of your pages. GUI site development applications such as Expression Web allow you to connect to, display and manipulate data stored in SQL databases, Oracle databases, Microsoft Access databases and XML data stores.

To integrate data from databases on your site using Expression Web, you need to use ASP.NET pages. To build an ASPX page, drag one or more ASP.NET controls onto the page from the Toolbox panel. The ASP.NET controls offer no-code data binding. That is, you can connect these controls to a data source and display, update, insert or delete data without having to write code. Expression Web handles the code generation for you. By connecting to a database, you can guarantee that every time the browser renders the ASPX page, the most current data will display in the page.

For more information and a hands-on lab about connecting Web pages to databases, see *Web Design Specialist*, Lesson 19: Site Development with Expression Web 3 – Advanced Features, and Lesson 33: Databases.

Practice exam questions

Following are some practice exam questions intended for student review.

1. A database consists of tables of information. Within a database table, which term is used to describe a row?

a. Field
b. Query
c. Record
d. Schema

Correct response: c. Record

Explanation: All database information is contained in tables. A table is a repository of information divided into columns and rows. The table consists of fields (columns) and records (rows). A record includes a complete set of information, such as name, e-mail address and phone number for one person. Schema is the structure of a database system and often depicts the structure as a graphical reference. Queries are questions from the user to the database table.

2. Which choice lists the four main types of SQL queries?

a. GET, POST, CREATE and EDIT
b. SELECT, INSERT, GET and POST
c. CREATE, UPDATE, EDIT and DELETE
d. SELECT, INSERT, UPDATE and DELETE

Correct response: d. SELECT, INSERT, UPDATE and DELETE

Explanation: There are several types of SQL queries, which manipulate the database in different ways. The four main types of SQL queries are SELECT, INSERT, UPDATE and DELETE.

4.4: Internet marketing

This subdomain includes skills and knowledge required to conduct effective Internet marketing.

4.4.1: SEO vs. PPC strategies

Search engine optimization (SEO) is the use of specific techniques to improve a Web page's or site's ranking in a search engine. The use of SEO techniques improves the volume and quality of traffic to a Web site based on the relevance of the site to search terms entered into a search engine. SEO techniques are said to be "organic," in that these tactics do not include paid advertisements of any kind.

You can also use pay per click (PPC) to help increase your search engine rankings. In PPC, you work with established services that help your site to be ranked highly in a search engine result page. PPC search results are not considered organic because instead of optimizing pages to make them appear naturally more relevant, you simply pay to have your page listed as highly as possible. Most Internet marketing campaigns combine SEO and PPC strategies.

For more information about organic and non-organic strategies for improving hit rates, see *Web Design Specialist*, Lesson 16: Site Content and Metadata.

4.4.2: SEO terms

The following table discusses some of the more common SEO terms.

Term	Definition
Conversion	Leading users to a specific result. The most typical conversion is a purchase of a good or service on a site. Conversion can also include capturing a user's personal information, demographic information or preferences.
Keywords	Words that search engines have identified as important enough to index on their sites. Keywords are used to determine page ranking. Keywords should be repeated naturally, and should be used consistently in Web and print copy.
Keyword density	The frequency with which a keyword is used on a page. The goal is to have sufficient keyword density without making the page seem artificially enhanced. Overuse of keywords is known as keyword stuffing.
Stop words	Common words (such as "a," "of," "the," "with," etc.) that are filtered out by search engines. Including stop words in your title tag weakens the title tag's keyword density.
Reciprocal link	An off-page practice wherein one site links back to another. Use of reciprocal links often lowers page ranking, because it is seen as a quick, artificial way to increase ranking.
Clickthrough rate	To obtain the clickthrough rate, divide the number of users who clicked on an advertisement by the number of times the advertisement was shown. The clickthrough rate is a measurement of how successful a campaign has been in terms of Web hits.
Results page	The page returned by a search engine. Contains ranked entries, as determined by the search engine's algorithms and PPC policies.
Black-hat SEO	The use of techniques that are considered by search engines to be flawed or unethical.

For more information about common search engine optimization terms, see *Web Design Specialist*, Lesson 16: Site Content and Metadata.

4.4.3: Common SEO techniques

You can use several on-page practices to optimize your Web pages for search engines. The following table describes several common, valid search engine optimization techniques.

Technique	Definition
Structure your pages properly	Use CSS to properly structure content on a page. Search engine crawlers will rank a page that uses CSS and the tag more highly than a page that uses HTML tables.
Use keywords properly	Choose keywords wisely and use them properly in your Web copy, as well as in all <meta>, <title> and <head> tags. Make sure your copy incorporates your keywords as part of the natural flow of the content. Avoid keyword stuffing, because search engines have become very good at differentiating between natural and artificial usage of words and phrases.
Write quality Web copy	Make your copy concise: brief and to the point. Make sure that your copy has a strong central idea and quickly describes your site's main purpose. Use a spelling checker, a grammar checker, and a trusted advisor who has good writing and editing skills.
Validate your code	Pick an HTML or XHTML standard and stick to it. Validate your HTML and CSS often.
Interlink your pages	Sites with an internal linking structure allow visitors to find what they want, and they also receive higher ranking by search engines because good navigation structure allows Web spiders to find all the pages on your site.
Use page titles	Search engine crawlers try to determine the content of a Web page. The <title> tag is mandatory on pages validated to HTML 3.2 and later anyway, so use your keyword or keyword phrase in the page title.
Use metadata	Metadata is content that is not displayed on the page, but is read and used by browsers and search engines. You can include a description of your page and keywords in your metadata.
Evaluate the use of images	Flash, video and images are, of course, valuable. However, bots do not read this data, and a page that relies too much on multimedia can cause a site to be ranked lower. Carefully combine text with multimedia to achieve a balance.
Add text links	Web spiders cannot follow image links or links created in JavaScript. Add extra text links in your navigation or page footers that lead to your main pages or important pages.
Embed text within images	If you use images, insert keywords and descriptions as meta information inside images.
Use the <alt> tag	Use the <alt> tag wisely by including a short, relevant description of each image.

For more information about SEO techniques, see *Web Design Specialist*, Lesson 16: Site Content and Metadata.

4.4.4: Web analytics

Web analytics is the collection, measurement and analysis of Internet data for the purpose of understanding and optimizing Web site usage. Using Web analytics can help you increase market share and generate sales. The following table describes the two types of Web analytics.

Technique	Definition
On-site analytics	On-site analytics involves studying visitor behavior once a visitor has accessed your site. On-site analytics tactics include: -**Log analysis** — You can use trend analysis software to analyze server logs and view Web site statistics including visitor demographic information, the number of hits, and how long visitors are staying. The most useful information includes determining repeat versus new visitors, determining the number of times an advertisement appears on a screen and determining the length of visitors' sessions. -**Page tagging** — You can use a language such as JavaScript to specially mark a page and create more sophisticated entries that determine exactly how long a visitor used a page. Page tagging includes determining mouse events that could be interpreted as clicking on a shopping cart or abandoning a particular page.
Off-site analytics	In off-site analytics, you analyze your audience and how well your site has addressed and penetrated that audience. Strategies for off-site analytics include: -Identifying size of the market. -Identifying your competitors. -Determining your market penetration. -Conducting surveys. -Consulting market research sources for your particular niche.

For more information about Web analytics, see *Web Design Specialist*, Lesson 16: Site Content and Metadata.

4.4.5: Generating revenue with search engines

Search engines generate revenue in various ways, including:

- **Pay-per-click ads.**
- **Providing value-add services.**
- **Saving and processing searches** — By saving and processing search entries that users make, a search engine can hone its offerings or even sell these search results to other businesses that want to gather information about the most popular searches.

Internet Service Providers (ISPs) have also adopted similar strategies. They can track each click and user request to determine user preferences. As you might expect, the practice of saving and archiving searches and browsing history has generated considerable controversy.

For more information about generating revenue with search engines, see *Web Design Specialist*, Lesson 16: Site Content and Metadata. For online articles on the controversy over the growing practice of saving and tracking searches, visit "ISPs tracking user behavior" (*http://consumerist.com/ 5008801/charter-to-begin-tracking-users-searches-and-inserting-targeted-ads)*; and "Google tracking browsing history" (*www.itpro.co.uk/610152/google-starts-tracking-users-for-better-ads).*

4.4.6: Blogging

Blogging should not be overlooked as a free and easy way to generate a stream of targeted traffic to your Web site. A properly managed blogging campaign, partnered with a great product or service Web site, can generate both traffic and sales leads.

Tips for creating a successful blog include:

- Targeting an audience about which you are knowledgeable.
- Focusing on one topic per blog.
- Including a clever title that will spark the reader's interest.

Your blog should be based on one major keyword or keyword phrase that relates to your product or service, and you should use this keyword throughout your blog posts. You should also employ the same writing tactics that you would use on a Web page (i.e., use clear headlines; write in short, concise sentences; run a spelling and grammar check before posting.)

In addition, you should submit your URL to blog directories and blog search engines, and you should post to your blog on a regular basis. Most blog-creation software includes an option to automatically alert various blog services when you update your content. Use these options to automatically notify the blog services when you add new posts.

Including links to other blogs and Web sites will help you build credibility and position yourself as an expert in your field. You can also exchange links (reciprocal links) with other blogs of a similar subject. You can notify another blogger that you have linked to one of his or her posts from your blog by using a trackback link. The trackback automatically leaves a link back to your blog as a comment.

If you want to create a blog to support your product, the blog should work hand-in-hand with your product Web site. Links provide both visibility and credibility for your blog and your product site. Each page on your product Web site should include a link to your blog. Vice versa, you can use your blog to direct traffic to your product Web site. Sign each post to your blog with your name and a link to your product Web site. Readers who are interested will click through to your main site.

For more information about using blogging as a marketing strategy, see *Web Design Specialist*, Lesson 16: Site Content and Metadata.

4.4.7: Web marketing campaigns

It is important to understand that a Web site is part of a business's marketing effort. Marketing is essentially presenting your product to potential customers in such a way that they want to buy it, and they want to buy it from you instead of from someone else.

An effective Web site should communicate your central message, and should be targeted to and designed for a specific audience.

The central message informs the audience about the purpose of the Web site (and the purpose of the business or individual to whom the site belongs). If you are designing a site for a customer, that customer may have an idea of what he or she wants his Web site to say, but you should clearly understand the central message before you begin your work.

Apply listening and discernment skills in order to clarify the central message in your own mind. If you are still unsure about the central message, attend a focus group or work closely with members of the marketing or sales teams to get a clear understanding of what the Web site needs to communicate.

You must also identify the type of visitors you trying to reach. For example, are you marketing to teenagers, to IT professionals or to the elderly? After you have identified the type of visitor to whom you are marketing, you must keep those visitors in mind as you design your site so that your site will be appealing and useful to them.

Many companies engage in HTML e-mail marketing campaigns, and this responsibility may fall to you as a Web designer. An HTML e-mail marketing message is more than merely an e-mail message sent in HTML format. Like any marketing piece, an e-mail marketing message must convey the central message quickly and concisely. It must also be well formatted and make an impact.

To achieve the desired impact, your e-mail marketing piece should adhere to the guidelines shown in the following table.

E-Mail Marketing Guideline	Description
Keep the message brief	Try to keep the message to 60 words or fewer.
Target the message to a specific audience	Are you marketing an iPod to college students or a Mercedes Benz to corporate executives? The language and writing style of your message should be geared toward your target audience.
Keep the message professional	Make sure to run the text of your message through a spelling and grammar check before you send it.
Make sure the message is properly formatted	A properly formatted e-mail marketing piece will include the following elements: -Date and time -Greeting -Central message -An action item -A hyperlink to more details and information -Contact information -An opt-out clause

As part of a marketing campaign, your company may provide Webcasts so that customers can get more information about a product or upcoming event. A Webcast is an audio and/or video file that is distributed over the Internet using streaming media technology. At all times, you should keep the process as simple and user-friendly as possible.

To invite your customers to a Webcast, you will need to include the pertinent details in your e-mail marketing piece. These details include:

- The URL where the Webcast is available and the timeframe during which it is available.
- Simple instructions for accessing and viewing/downloading the Webcast. You might also want to include a toll-free telephone number that users can call for technical assistance with viewing the Webcast, if your company provides such resources.
- Minimum system and browser requirements, including any required plug-ins and links for obtaining the plug-ins.
- Details about any information that will be requested from Webcast participants, such as a user name. You should explain what data will be collected from participants and how it will be used.

For more information about Web marketing campaigns, see *Web Design Specialist*, Lesson 2: Web Development Teams.

Practice exam questions

Following are some practice exam questions intended for student review.

1. Using trend analysis software to review server logs is an example of which of the following?

a. On-page practices
b. Black-hat SEO
c. On-site analytics
d. Conversion

Correct response: c. On-site analytics

Explanation: On-site analytics involves studying visitor behavior once a visitor has accessed your site. On-site analytics tactics include log analysis and page tagging. An example of log analysis is using trend analysis software to analyze server logs and view Web site statistics including visitor demographic information, the number of hits, and how long visitors are staying.

2. Which of the following choices is a non-organic tactic for improving your Web site's ranking on a search engine?

a. Structuring the page using CSS
b. Using metadata
c. Interlinking the pages
d. Using pay per click (PPC)

Correct response: d. Using pay per click (PPC)

Explanation: You can use pay per click to help increase your search engine rankings. In pay per click (PPC), you work with established services that help your site to be ranked highly in a search engine result page. PPC search results are not considered organic because instead of optimizing pages to make them appear naturally more relevant, you simply pay to have your page listed as highly as possible. Most Internet marketing campaigns combine SEO and PPC strategies.

4.5: Syndicated feeds

This subdomain includes skills and knowledge required to create syndicated feeds using feed management services.

4.5.1: Definition of syndication

Syndication is the sharing of content among different Web sites. A Web feed (or news feed) is a document (usually XML-based) whose content items include hyperlinks to the source of the content. Web feeds are used to provide users with frequently updated content. News Web sites and blogs are common sources for Web feeds. The two main Web feed formats are RSS and Atom.

For more information about syndication and its benefit, see *Web Design Specialist*, Lesson 32: HTTP Servers and Web Applications.

4.5.2: RSS and Atom

The two main Web feed formats are RSS and Atom.

RSS (Really Simple Syndication, RDF Site Summary or Rich Site Summary) is the "original" family of Web feeds. Introduced in the late 1990s, RSS is an XML-based format intended for computers to understand and not for human consumption. RSS 2.0 has the widest acceptance of any feed format.

RSS is a Web content syndication format and is a dialect of XML. All RSS files must conform to the XML 1.0 specification. At the top of an RSS document is the <rss> element and its version attribute. Below the <rss> element is a single <channel> element, which contains metadata about the channel and contents. The contents of an RSS channel are called items.

The following code represents a sample RSS feed:

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
    <channel>

        <title>Tips From Red</title>
        <description>The latest tips from my fishing site</description>
        <link>http://www.fishingwithred.com/</link>
        <lastBuildDate>Sat, 11 Dec 2010 18:30:02 GMT</lastBuildDate>
        <managingEditor>redsmith@fishingwithred.com (Red Smith)</managingEditor>

    <item>
        <title>Land that mako</title>
        <link>http://fishingwithred.com/2010/12/11/mako</link>
        <guid isPermaLink="false">urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</guid>
        <pubDate>Sat, 11 Dec 2010 18:30:02 GMT</pubDate>
        <description>Tips for catching makos on a longline.</description>
    </item>

    </channel>
</rss>
```

Much of the widespread use of RSS 2.0 can be attributed to its early arrival on the scene and its de facto use by WordPress, a popular content management system provider. It has some limitations, however:

- RSS does not allow any well-formed XML markup in its content, which reduces the reusability of the content.
- The RSS payload can contain only plain text or escaped HTML, which makes it ugly to look at in its raw form.
- There is also no way to indicate which of the two types of content (plain text or escaped HTML) is provided in the payload.

Atom is a relatively recent development but is much more robust and feature-rich than RSS. Atom is designed solely for the syndication of entire news articles. Atom is an XML-based file format designed to allow information to be syndicated between applications. Unlike RSS, Atom provides not only the document's content, but also metadata about the document: what it is called, who created it, when it was created and where it is located.

An Atom feed consists of metadata about the feed, followed by one or more entries. An Atom entry is the equivalent of an RSS channel item.

A key issue when syndicating data is to make sure that information is not lost in the process. Atom is designed to prevent loss of data. The following code represents a sample Atom feed:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">

    <title>Tips from Red</title>
    <link href="http://fishingwithred.com/"/>
    <updated>2010-12-11T18:30:02Z</updated>
    <author>
        <name>Red Smith</name>
    </author>
    <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

    <entry>
        <title>Land that mako</title>
        <link href="http://fishingwithred.com/2010/12/11/mako"/>
        <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
        <updated>2010-12-11T18:30:02Z</updated>
        <summary>Tips for catching makos on a longline.</summary>
    </entry>

</feed>
```

Because an Atom feed is XML, it must follow the rules for being well-formed. All the elements must be in the *www.w3.org/2005/Atom* namespace. Atom also provides standardized elements called constructs. Constructs allow new elements within new namespaces to be added to an Atom feed in a controlled fashion.

The following table compares various characteristics of RSS and Atom.

Feature	RSS	Atom
Current version	RSS 2.0	Atom 1.0
Intended for human consumption	No	Yes
Feed structure	An RSS channel contains one or more channel items.	An Atom feed contains one or more feed entries.
Date format	RFC 882 format (e.g., *Sat, 11 Dec 2010 00:00:01 GMT*)	RFC 3339 format (e.g., *2003-12-13T18:30:02Z*)
Payload	Plain text or escaped HTML	Plain text, escaped HTML, well-formed XHTML markup, some other XML vocabulary, base 64-encoded binary content, a pointer to Web content not included in the feed

For more information about RSS and Atom and a hands-on lab for creating an RSS news feed, see *Web Design Specialist*, Lesson 32: HTTP Servers and Web Applications.

4.5.3: Aggregators

Web feeds can be read using software called a news reader, feed reader or aggregator. An aggregator is a modern Web browser (Firefox, Opera, Internet Explorer 8, Safari) or a client application that collects, or aggregates, syndicated Web content into a single location that users can use to easily view the content. Depending on the aggregator, users typically subscribe to a feed by manually entering the URL of the feed or clicking a link in a Web browser.

An aggregator asks the servers in its feed list if they have new content. When new content becomes available, the aggregator provides links to the new content.

For more information about aggregators, see *Web Design Specialist*, Lesson 32: HTTP Servers and Web Applications.

4.5.4: RSS in e-mail conversion

A feed-burning service provides the ability for individuals to syndicate content. This content can include your own offerings (your blogs, podcasts and videos) or those of others.

The following table describes the chief benefits and services of a feed-burning service.

Service/Benefit	Description
Access control	You can add and drop users from your list.
Dual feeds	You can create a "light" feed that is free, and create a more comprehensive feed for those who pay for your feed.
Analytic services	You can track user requests, as well as get information about which of your offerings is the most popular. Studying this information can help you create a more customer-driven news feed.
Publication services	Many news feeds provide the ability for you to inform groups of people about your news feed, often for a fee.
Advertisements	A news feed service can offer you targeted advertisements.

Some users want to view their Web feeds in the familiar setting of their e-mail inbox, rather than subscribing to a feed and viewing it through an aggregator. Applications known as Web feed management providers allow users to accomplish this. These applications allow users to receive their Web feeds in their e-mail inboxes, similar to an e-newsletter, thereby making RSS feeds available to novice users who may shy away from feed aggregators but who are comfortable with e-mail.

Popular Web feed management providers include Google FeedBurner (*http://feedburner.google.com*), FeedBlitz (*www.feedblitz.com*) and Aweber (*www.aweber.com*). These providers also offer feed-burning services.

Although Web feeds are an example of a pull technology (the subscriber makes an active choice to subscribe to a feed, and the feed aggregator polls the feed service and pulls in updates), converting RSS to e-mail crosses the threshold and becomes an example of a push service. In push technology, the server initiates the request for data exchange; data is sent to a client without the immediate request from the client. SMTP (used to send e-mail) is a push technology protocol pushing information to the Mail Delivery Agent, or mail server. When users retrieve e-mail messages from a server (using either POP3 or IMAP), they use the Mail User Agent, or e-mail client. This is an example of pull technology. Instead of thinking of push and pull as opposing technologies, it may be more accurate to view them as interrelated phases of the same service.

For more information about Web feed management providers and RSS-to-e-mail conversion, see *Web Design Specialist*, Lesson 32: HTTP Servers and Web Applications.

Practice exam questions

Following are some practice exam questions intended for student review.

1. Which choice lists two chief benefits of using a feed-burning service?

a. Access control and analytics services
b. Reciprocal links and improved page ranking
c. Automated review of server logs and page tagging
d. Increased keyword density and higher conversion rate

Correct response: a. Access control and analytic services

Explanation: Benefits of using a feed-burning service include access control (you can add or drop users from your list), dual feeds (you can create a "light" feed that is free, and create a more comprehensive feed for those who pay for your feed), analytic services (you can track user requests, as well as get information about which of your offerings is the most popular), publication services (many news feeds provide the ability for you to inform groups of people about your news feed, often for a fee) and advertisements (a news feed service can offer you targeted advertisements).

2. Which of the following is an advantage of using RSS-to-e-mail conversion?

a. Most e-mail programs automatically check spelling when you send messages.
b. You can push your updated content to users without waiting for them to request an update.
c. Users control when they receive updated content.
d. RSS-to-e-mail uses pull technology, which is always better than using push technology.

Correct response: b. You can push your updated content to users without waiting for them to request an update.

Explanation: Web feeds are an example of pull technology (the subscriber makes an active choice to subscribe to a feed, and the feed aggregator polls the feed service and pulls in updates). Therefore, you must wait until a user requests updated content before your updates can be viewed. Converting RSS to e-mail crosses the threshold and becomes an example of a push service. In push technology, the server initiates the request for data exchange; data is sent to a client without the immediate request from the client. SMTP, which is used to send e-mail, is a push technology protocol pushing information to the Mail Delivery Agent, or mail server. Therefore, you can push your updated content to users without waiting for them to request an update.

Works Consulted

The following resources were used as references for various topics found in the CIW Web Design Specialist Exam Study Guide. In addition, these resources are useful study tools for students preparing for the CIW Web Design Specialist exam. Objectives throughout this Exam Study Guide reference CIW Web Design Specialist coursebooks for more information, as well as other related resources for further study.

Brown, Jeffrey, Irina Heer, Kenneth A. Kozakis, James Stanger, Ph.D., Chris Minnick and Susan M. Lane. *Web Design Specialist (Adobe CS5 Web Edition),* version 1.0. Tempe, AZ: Certification Partners, LLC, 2010. (ISBN: 0-7423-2719-1)

Brain, Marshall. "How ASPs Work," *HowStuffWorks*. http://computer.howstuffworks.com/asp.htm.

"Cascading Style Sheets, level 1: W3C Recommendation," *World Wide Web Consortium.* http://www.w3.org/TR/REC-CSS1 (May 12, 1998). Copyright © World Wide Web Consortium. All rights reserved, http://www.w3.org/consortium/legal.

"Cascading Style Sheets, level 2: CSS2 Specification," *World Wide Web Consortium.* http://www.w3.org/TR/CSS2 (revised Jan. 11, 1999). Copyright © World Wide Web Consortium. All rights reserved, http://www.w3.org/consortium/legal.

Chisholm, Wendy, Gregg Vanderheiden, and Ian Jacobs. *W3C Recommendation: Web Content Accessibility Guidelines 1.0.* Cambridge, Mass.: World Wide Web Consortium. http://www.w3.org/TR/WAI-WEBCONTENT (May 5, 1999). Copyright © World Wide Web Consortium. All rights reserved, http://www.w3.org/consortium/legal.

Connolly, Dan, Rohit Khare and Adam Rifkin. "The Evolution of Web Documents: The Ascent of XML," *World Wide Web Journal, Special Issue on XML,* Vol. 2, No. 4, Fall 1997, pp. 119-128. http://www.w3journal.com/xml.

"CSS Authoring Tools," *World Wide Web Consortium.* http://www.w3.org/Style/CSS/#editors. Copyright © World Wide Web Consortium. All rights reserved, http://www.w3.org/consortium/legal.

"CSS1 Test Suite," *World Wide Web Consortium*. http://www.w3.org/Style/CSS/Test/CSS1 (June 28, 2000). Copyright © World Wide Web Consortium. All rights reserved, http://www.w3.org/consortium/legal.

Davies, Jason. "Cookies in JavaScript," *Netspade.* http://www.netspade.com/articles/javascript/cookies.xml.

"Extensible Markup Language (XML) 1.0 Recommendation," *World Wide Web Consortium.* http://www.w3.org/TR/1998/REC-xml-19980210 (Feb. 10, 1998). Copyright © World Wide Web Consortium. All rights reserved, http://www.w3.org/consortium/legal.

The Face of the Web, *Ipsos-Insight*. http://www.ipsos-insight.com/industryfocus/techandcomm/FOW.aspx.

"Floatutorial simple tutorial on CSS floats," *MaxDesign.com*. http://css.maxdesign.com.au/floatutorial.

Hunt, Ben. "SEO Basics, Introduction to Search Engine Optimisation," *Web Design from Scratch.* http://www.webdesignfromscratch.com/search-engine-optimisation-basics.php.

IT Facts.biz Web site. http://www.itfacts.biz.

Mayer-Schönberger, Viktor. "The Cookie Concept," *Cookie Central Web site.* http://www.cookiecentral.com/c_concept.htm.

Netcraft. Web Server Survey. *Netcraft Web site*.
http://news.netcraft.com/archives/web_server_survey.html.

Rappa, Michael. "Managing the Digital Enterprise," *North Carolina State University.*
http://digitalenterprise.org/index.html. Copyright © Michael Rappa. All rights reserved, http://digitalenterprise.org/legal.html

Robie, Jonathan. "What is the Document Object Model?" *World Wide Web Consortium*.
http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/introduction.html. Copyright © World Wide Web Consortium. All rights reserved, http://www.w3.org/consortium/legal.

Siegel, David. *Creating Killer Web Sites, Second Edition*. http://www.killersites.com.

Sitepoint Web site. http://www.sitepoint.com.

Site Reference Web site. http://www.site-reference.com.

St. Laurence, Simon. *XML: A Primer.* Foster City: MIS Press, 1998.

Stein, Bob. *VisiBone Webmasters Color Laboratory*. http://www.visibone.com/colorlab.

Web Address FAQs, Domain Search Help and Web Site Name Information, *Domain.com Web site.*
http://www.domain.com.

Valentine, Chelsea and Minnick, Chris. *XHTML*. Indianapolis, IN: New Riders, 2001. ISBN: 0735710341

Weinman, Lynda. Non-Dithering Colors Page. Ojai Digital Arts Center.
http://www.lynda.com/resources/webpalette.aspx.

Wikipedia site. http://www.wikipedia.org.

"XHTML 1.0 The Extensible HyperText Markup Language (Second Edition),"
http://www.w3.org/TR/xhtml1/ (revised Aug. 1, 2002). Copyright © World Wide Web Consortium. All rights reserved, http://www.w3.org/consortium/legal.

www.CIWcertified.com

ISBN 0-7423-2803-1

CCSSGCDWDDGPR1102