**Course Description:**

Students will learn the dynamics of the Web environment while pursuing an in-depth study of both Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). Web based protocols such as FTP, TCP/IP, and HTTP will be addressed. Students will create a website with tag text elements, special characters, lines, graphics, hypertext links, and graphical tables."

### 

### Strand 2. IT Fundamentals

###### Learners apply fundamental principles of IT, including the history of IT and its impact on society, common industry terms, systems theory, information storage and retrieval, database management, and computer hardware, software, and peripheral device configuration and installation. This base of knowledge and skills may be applied across the career field.

**Outcome 2.4. Emerging Technologies**

Identify trending technologies, their fundamental architecture, and their value in the marketplace.

**Competencies**

2.4.1. Investigate the scope and the impact of mobile computing environments on society.

2.4.2. Describe the differences, advantages, and limitations of cloud computing (e.g., public cloud, private cloud, hybrid cloud) and on premises computing.

2.4.3. Utilize cloud computing applications (e.g., services, applications, virtual environments).

2.4.4 Describe emerging technologies (e.g., Bring your Own Device [BYOD], Services Virtualization, Augmented Reality [AR], SMART Devices, Additive Manufacturing [3D Printing]).

**Outcome: 2.7 Web Architecture**

Explain the fundamentals of delivering information and applications using web architecture.

**Competencies**

2.7.1. Describe methods of securely transmitting data.

2.7.2. Describe ways to present data (e.g., responsive web design, mobile applications, desktop applications, web applications).

2.7.3. Differentiate between a client and a server.

2.7.4. Identify how the use of different browsers and devices effects the look of a webpage (e.g., Americans with Disabilities Act [ADA]).

2.7.5. Explain the relationship between data transmission volumes, bandwidth, and latency.

2.7.6. Describe the characteristics and use of browser plug-ins.

2.7.7. Compare the advantages and disadvantages of running an in-house server or using a service provider.

2.7.8. Describe the difference between static and dynamic sites and the reasons for using each.

**Outcome: 2.9. Project Concept Proposal**

Develop a project concept proposal.

**Competencies**

2.9.1 Identify and incorporate branding strategies.

2.9.2. Determine the scope and purpose of the project.

2.9.3. Determine the target audience, client needs, expected outcomes, objectives, and budget.

2.9.4. Develop a conceptual model and design brief for the project.

2.9.5. Develop a timeline, a communication plan, a task breakdown, costs (e.g., equipment, labor), deliverables, and responsibilities for completion.

2.9.6. Develop and present a comprehensive proposal to stakeholders.

**Outcome: 2.11. Troubleshooting**

Select and apply troubleshooting methodologies for problem solving.

**Competencies**

2.11.1. Identify the problem.

2.11.2. Select troubleshooting methodology (e.g., top down, bottom up, follow the path, spot the differences).

2.11.3. Investigate symptoms based on the selected methodology.

2.11.4. Gather and analyze data about the problem.

2.11.5. Design a solution.

2.11.6. Test a solution.

2.11.7. Implement a solution.

2.11.8. Document the problem and the verified solution.

**Outcome: 2.12. Performance Tests and Acceptance Plans**

Develop performance tests and acceptance plans.

**Competencies**

2.12.1. Create a written procedure agreed by the stakeholders and project team for determining the acceptability of the project deliverables.

2.12.2. Develop a test system that accurately mimics external interfaces.

2.12.3. Develop test cases that are realistic, compare with expected performance, and include targeted platforms and device types.

2.12.4. Develop, perform, and document usability and testing integration.

2.12.5. Make corrections indicated by test results.

2.12.6. Seek stakeholder acceptance upon successful completion of the test plan.

**Outcome: 2.13. Rollout and Handoff**

Plan rollout and facilitate handoff to customer.

**Competencies**

2.13.1. Include overall project goals and timelines in the rollout plan.

2.13.2. Communicate rollout plans to key stakeholders in a timely manner.

2.13.3. Conduct final review and approvals according to company standards.

2.13.4. Identify support staff, training needs, and contingency plans in the rollout plan.

2.13.5. Test delivered application to assure that it is fully functional for the customer or user and meets all requirements.

2.13.6. Deliver support and training materials.

### 

### Strand 4. Infrastructure Systems

###### Learners apply principles of networking and infrastructure related to the installation, administration, and maintenance of computer networks and components. Knowledge and skills may be applied to network connectivity, cabling, protocols, architecture, classification, topologies, operating systems, Open Systems Interconnection (OSI) standards, data encoding, Quality of Service (QoS), Internet Protocol (IP) addressing, and wide area network (WAN) design.

**Outcome: 4.6. Network Protocols**

Compare network protocols.

**Competencies**

4.6.1. Explain network protocols (e.g., Transmission Control Protocol/Internet Protocol [TCP/IP], User Datagram Protocol [UDP], Internet Protocol Version 4 [IPv4], Internet Protocol Version 6 [IPv6]).

4.6.2. Identify the advantages of protocols (e.g., Domain Name System [DNS], File Transfer Protocol [FTP], Hypertext Transfer Protocol [HTTP], Telecommunications Network [Telnet], Remote Desktop Protocol [RDP]], Secure Shell [SSH]) and associated port numbers.

4.6.3. Explain the purposes of encapsulation and decapsulation and their relationship to the Open Systems Interconnection (OSI) model.

4.6.4. Explain the difference between User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).

4.6.5. Identify Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) conventional ports (e.g., Simple Mail Transfer Protocol [SMTP], Telnet, Hypertext Transfer Protocol [HTTP], File Transfer Protocol [FTP]).

4.6.6. Explain Transmission Control Protocol/Internet Protocol (TCP/IP) protocol details (e.g., Internet addresses, Address Resolution Protocol [ARP], Reverse Address Resolution Protocol [RARP], IP datagram format, routing IP datagrams, TCP segment format, IPv4, IPv6).

4.6.7. Describe a Virtual Private Network (VPN) and identify associated protocols (e.g., Layer 2 Tunneling Protocol [L2TP], Point-to-Point Tunneling Protocol [PPTP]).

4.6.8. Capture and analyze data packets.

**Outcome: 4.7. Transmission Control Protocol/Internet Protocol (TCP/IP)**

Describe IP addressing schemes and create subnet masks.

**Competencies**

4.7.1. Explain Fully Qualified Domain Names (FQDNs) and how they are used.

4.7.2. Explain the IP addressing scheme and how it is used.

4.7.3. Identify Class A, B, and C reserved (i.e., private) address ranges and why they are used.

4.7.4. Identify the class of network to which a given address belongs.

4.7.5. Differentiate between default subnet masks and custom subnet masks.

4.7.6. Explain the relationship between an IP address and its associated subnet mask.

4.7.7. Identify the differences between classful and classless addressing schemes.

4.7.8. Identify multicasting addresses and explain why they are used.

4.7.9. Create custom subnet masks to meet network design requirements.

4.7.10. Compare Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6).

**Strand 6. Web Development**

Learners apply principles of design and technology, including programming standards and protocols, to create, test, host, and maintain web pages and websites with text, graphics, multimedia, scripting, linking, and data integration in a structure that is easy to navigate and accessible for all users via a variety of hardware and software platforms.

**Outcome: 6.1. Webpages**

Create basic web pages.

**Competencies**

6.1.1 Describe the basic principles of Hypertext Markup Language (HTML) and its functional relationship with web browsers.

6.1.2 Plan a webpage considering subject, devices, audience, layout, color, links, graphics, and Americans with Disabilities Act (ADA) requirements.

6.1.3 Format the text of a webpage in a WYSIWYG (What You See Is What You Get) editor and in a text editor using HTML formatting tags (e.g., hyperlink, e-mail, table formatting, graphic attributes).

6.1.4 Use writing process techniques (i.e., drafting, revising, editing, proofreading) to check the webpage for format and text accuracy.

6.1.5 Create and format ordered and unordered lists on a webpage using HTML list formatting tags.

6.1.6 Create and format a table in a webpage using HTML table formatting tags and attributes.

6.1.7 Integrate styles (e.g., inline or external Cascading Style Sheets [CSS]).

**Outcome: 6.2. Links and Multimedia**

Add links to a web page and insert multimedia files.

**Competencies**

6.2.1. Create absolute links and relative links.

6.2.2. Write a Hypertext Markup Language (HTML) anchor that links to another section of the same web page.

6.2.3. Create hyperlinks that send e‐mail messages and download files.

6.2.4. Insert image and wrap text around the image using Cascading Style Sheets (CSS).

6.2.5. Resize a graphic image in a web page using CSS.

6.2.6. Insert media files (e.g., audio, video,) into a web page using HTML tags.

6.2.7. Build a hover or mouseover effect to change the style of a link.

**Outcome: 6.3. Scripting**

Integrate scripting into a web page.

**Competencies**

6.3.1. Select and apply scripting languages used in web development.

6.3.2. Insert client-side script into a webpage.

6.3.3. Insert comments into client-side scripts.

**Outcome: 6.4. Web Forms**

Integrate forms into a web page.

**Competencies**

6.4.1. Design a data entry form from specifications that will accept a variety of user inputs (e.g., radio buttons, text entry fields, check boxes, drop-down menus).

6.4.2. Write the Hypertext Markup Language (HTML) code to add a form to a webpage.

6.4.3. Write the HTML code to add text entry fields, radio buttons, check boxes, drop-down menus, and other user inputs to a form.

6.4.4. Explain the concept of a form action.

6.4.5. Write the HTML code to add a working button (e.g., submit, reset) to a form.

6.4.6. Format a completed form using HTML and Cascading Style Sheets (CSS) (e.g., fieldset, tabindex).

6.4.7. Code scripting to interact with data sources (e.g., database, web services).

**Outcome: 6.5. Websites**

Create and update a website.

**Competencies**

6.5.1. Implement web programming standards and protocols (e.g., World Wide Web Consortium [W3C], Hypertext Markup Language [HTML] 5).

6.5.2. Plan a website’s structure for navigation and usability.

6.5.3. Use standard web programming languages (e.g., markup, scripting languages) in website development.

6.5.4. Install and configure a content management system (CMS)

6.5.5. Select an integrated development environment (IDE).

6.5.6. Create and edit a webpage template.

6.5.7. Create and attach cascading style sheets (CSS).

6.5.8. Format website layout (e.g., targeted platforms, text formatting, background color, text, tables, lists, iframes).

6.5.9. Incorporate audio and video, forms, and links on a website.

6.5.10. Develop and execute usability tests on a completed website, checking for information accessibility, ease of use, and navigation.

6.5.11. Code a website for cross-platform and cross-browser compatibility and validation.

6.5.12. Publish the completed website to a web server.