**Course Description:**

Students will learn to create applications for mobile devices using a variety of commercial and open source software. They will install these applications, modify them, and develop customer service skills to handle user issues. Knowledge and skills related to customer service in professional offices, small businesses, departments, work groups, and corporate information services will be addressed.

### 

### 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: 2.9. 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: 2.13. 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.

2.12.3. 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 3. Information Security

###### Learners apply principles of information security to implement and maintain security compliance and network security. Learners select components and mechanisms required for a multilayer defense structure and evaluate and minimize security risks to wired and wireless networks and devices.

**Outcome: 3.2. General Security Compliance**

Implement and maintain general security compliance.

**Competencies**

3.2.1. Identify and implement data and application security.

3.2.2. Implement backup and verification procedures (e.g., tape, disk, cloud).

3.2.3. Describe and assign permissions (e.g., read-only, read-write).

3.2.4. Provide user authentication (e.g., assign and reset user accounts and passwords).

3.2.5. Install, test, implement, and update virus and malware detection and protection software.

3.2.6 Identify sources of virus and malware infection and remove viruses and malware.

3.2.7. Provide documentation, training, and support to users on established security procedures.

3.2.8. Identify the need for disaster recovery policies and procedures.

### Strand 5. Programming and Software Systems

Learners apply principles of computer programming and software development to develop code; build, test, and debug programs; create finished products; and plan, analyze, design, develop, implement, and support software applications.

**Outcome: 5.1. Programming Concepts**

Describe programming concepts.

**Competencies**

5.1.1. Describe how computer programs and scripts can be used to solve problems (e.g., desktop, mobile, enterprise).

5.1.2. Explain how algorithms and data structures are used in information processing.

5.1.3. Model the solution using both graphic tools (e.g., flowcharts) and pseudocode techniques.

5.1.4. Describe, compare, and contrast the basics of procedural, structured, object-oriented (OO), and event driven programming.

5.1.5. Describe the concepts of data management through programming languages.

5.1.6. Analyze the strengths and weaknesses of different languages for solving a specific problem.

5.1.7. Compare the functions and operations of compilers and interpreters.

5.1.8. Describe version control and the relevance of documentation.

**Outcome: 5.2. Computational and String Operations**

Develop code that performs computational and string operations.

**Competencies**

5.2.1. Compare primitive types of numeric and nonnumeric data (e.g., integers, floats, Boolean, strings).

5.2.2. Identify the scope of data (e.g., global versus local, variables, constants, arrays).

5.2.3. Write code that uses arithmetic operations.

5.2.4. Write code that uses subtotals and final totals.

5.2.5. Write code that applies string operations (e.g., concatenation, pattern matching, substring).

**Outcome: 5.3. Logical Operations and Control Structures**

Develop code that uses logical operations and control structures.

**Competencies**

5.3.1. Explain Boolean logic.

5.3.2. Solve a truth table.

5.3.3. Write code that uses logical operators (e.g., and, or, not).

5.3.4. Write code that uses relational operators and compound conditions.

5.3.5. Write code that uses conditional control structures (e.g., if, if-then-else).

5.3.6. Write code that uses repetition control structures (e.g., while, for).

5.3.7. Write code that uses selection control structures (e.g., case, switch).

5.3.8. Write code that uses nested structures and recursion.

5.3.9. Write code that creates and calls functions.

5.3.10. Code error handling techniques.

5.3.11. Write code to access data repositories.

5.3.12. Write code to create classes, objects, and methods.

**Outcome: 5.4. Integrated Development Environment**

Build and test a program using an integrated development environment (IDE).

**Competencies**

5.4.1. Configure options, preferences, and tools.

5.4.2. Write and edit code in the integrated development environment (IDE).

5.4.3. Compile or interpret a working program.

5.4.4. Define test cases.

5.4.5. Test the program using defined test cases.

5.4.6. Correct syntax and runtime errors.

5.4.7. Debug logic errors.

**Outcome 5.5. Programming Conventions**

Develop programs using applications security practices.

**Competencies**

5.5.1. Develop programs using data validation techniques.

5.5.2. Develop programs that use reuse libraries.

5.5.3. Develop programs using operating system calls.

5.5.4. Develop programs that call other programs.

5.5.5. Use appropriate naming conventions and apply comments.

5.5.6. Format output (e.g., desktop, mobile, enterprise, reports, data files).

**Outcome 5.6. Software Development Lifecycle**

Apply the software development lifecycle (SDLC).

**Competencies**

5.6.1. Determine requirements specification documentation.

5.6.2. Identify constraints and system processing requirements.

5.6.3. Develop and adhere to timelines.

5.6.4. Identify a programming language, framework, and an integrated development environment (IDE).

5.6.5. Identify input and output (I/O) requirements.

5.6.6. Design system inputs, outputs, and processes.

5.6.7. Document a design using the appropriate tools (e.g., program flowchart, dataflow diagrams, Unified Modeling Language [UML]).

5.6.8. Create documentation (e.g., implementation plan, contingency plan, data dictionary, user help).

5.6.9. Review the design (e.g., peer walkthrough).

5.6.10. Present the system design to stakeholders.

5.6.11. Develop the application.

5.6.12. Compare software methodologies (e.g., agile, waterfall).

5.6.13. Perform code reviews (e.g., peer walkthrough, static analysis).

5.6.14. Ensure code quality by testing and debugging the application (e.g., system testing, user acceptance testing).

5.6.15. Train stakeholders.

5.6.16. Deploy the application.

5.6.17. Collect application feedback and maintain the application.

**Outcome: 5.7. Configuration/Change Management**

Describe configuration management activities.

**Competencies**

5.7.1. Explain version management and interface control.

5.7.2. Explain baseline and software lifecycle phases.

5.7.3. Analyze the impact of changes.

**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: 6.2. 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 email 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. Configuration/Change Management**

Describe configuration management activities.

**Competencies**

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

6.3.2. Insert client-side script into a web page.

6.3.3. Insert comments into client-side scripts.

### Strand 7. Digital Media

Learners apply principles of digital media to produce interactive media; develop and produce multimedia applications; integrate typography into media; create 3D models and 2D and 3D animation; and create digital video, audio, and photographs.

**Outcome 7.2. Multimedia Tools**

Develop navigational structures, scripts, storyboards, and flowcharts for multimedia applications.

**Competencies**

7.2.1. Choose a navigational menu structure (e.g., rollovers, drop-downs, disjointed).

7.2.2. Construct and place navigational units.

7.2.3. Build in interactive elements

7.2.4. Determine uses and needs for site maps, multimedia scripts, storyboards, and flowcharts.

7.2.5. Make preliminary sketches showing placement of images and text on screen.

7.2.6. Place buttons and navigational graphics.

7.2.7. Select colors based on color theory and psychology.

7.2.8. Describe music, video, and special effects to be used.

7.2.9. Provide a sample layout to stakeholders for review.

7.2.10. Select and create visual design elements appropriate for the intended audience and use

7.2.11. Develop characters and narrative to support intended outcomes.

**Outcome 7.4. Graphics**

Construct and manipulate digital graphics.

**Competencies**

7.4.1. Identify the purpose and intended audience of graphics.

7.4.2. Select color, shape, size, and texture of objects.

7.4.3. Create or acquire graphics.

7.4.4. Manipulate and layer objects

7.4.5. Differentiate between vector and raster graphics.

7.4.6. Select an appropriate graphic file format and resolution

7.4.7. Optimize and export graphics files for intended use.

7.4.8. Select graphic software applications.

7.4.9. Manipulate graphic objects.

7.4.10. Compress and decompress graphic files.

7.4.11. Describe and select color profiles (e.g., Red Green Blue [RGB], Cyan Magenta Yellow Key [CMYK], Pantone)