**Course Description:**

### Students will learn the theory and practice of software testing and develop an understanding of the analysis and design phases of software development. Students will effectively use appropriate programming languages and software patterns to improve software development. A variety of commercial and open source programs, applications, and tools will be used.

**Strand 1. Business Operations/21st Century Skills**

Learners apply principles of economics, business management, marketing, and employability in an entrepreneur, manager, and employee role to the leadership, planning, developing, and analyzing of business enterprises related to the career field.

**Outcome: 1.4. Knowledge Management and Information Technology**

Demonstrate current and emerging strategies and technologies used to collect, analyze, record, and share information in business operations.

**Competencies**

1.4.1. Use office equipment to communicate (e.g., phone, radio equipment, fax machine, scanner, public address systems).

1.4.2. Select and use software applications to locate, record, analyze and present information (e.g., word processing, e-mail, spreadsheet, databases, presentation, Internet search engines).

1.4.3. Verify compliance with security rules, regulations and codes (e.g., property, privacy, access, accuracy issues, client and patient record confidentiality) pertaining to technology specific to the industry pathway.

1.4.4. Use system hardware to support software applications.

1.4.5. Use information technology tools to maintain, secure and monitor business records.

1.4.6. Use an electronic database to access and create business and technical information.

1.4.7. Use personal information management and productivity applications to optimize assigned tasks (e.g., lists, calendars, address books).

1.4.8. Use electronic media to communicate and follow network etiquette guidelines.

### 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 2.3. Data Encoding**

Explain and describe data encoding basics.

**Competencies**

2.3.1. Identify and explain coding information and representation of characters (e.g., American Standard Code for Information Interchange [ASCII], Extended Binary Coded Decimal Interchange Code [EBCDIC], Unicode).

2.3.2. Convert between numbering systems (e.g., binary, hexadecimal, decimal).

**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).

**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: 2.12. 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.

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

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

**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).

**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.

**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.

5.6.12. 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.

**Outcome: 5.7. Configuration 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.