Career & Technical Education | Information Technology
Programming
Subject Code: 145060
Outcome & Competency Descriptions

Course Description:
In this course, students will learn the basics of building simple interactive applications. Students will learn the basic units of logic: sequence, selection, and loop. Students will apply algorithmic solutions to problem-domain scenarios. Students will gain experience in using commercial and open-source languages, programs, and applications.

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: 1.2. Leadership and Communications

Process, maintain, evaluate, and disseminate information in a business. Develop leadership and team building to promote collaboration.

Competencies

1.2.1. Extract relevant, valid information from materials and cite sources of information.
1.2.2 Deliver formal and informal presentations.
1.2.3. Identify and use verbal, nonverbal, and active listening skills to communicate effectively.
1.2.4. Use negotiation and conflict-resolution skills to reach solutions.
1.2.5. Communicate information (e.g., directions, ideas, vision, workplace expectations) for an intended audience and purpose.
1.2.6. Use proper grammar and expression in all aspects of communication.
1.2.7. Use problem-solving and consensus-building techniques to draw conclusions and determine next steps.
1.2.8. Identify the strengths, weaknesses, and characteristics of leadership styles that influence internal and external workplace relationships.
1.2.9. Identify advantages and disadvantages involving digital and/or electronic communications (e.g., common content for large audience, control of tone, speed, cost, lack of non-verbal cues, potential for forwarding information, longevity).
1.2.10. Use interpersonal skills to provide group leadership, promote collaboration, and work in a team.
1.2.11. Write professional correspondence, documents, job applications, and resumés.
1.2.12. Use technical writing skills to complete forms and create reports.

Ohio Department of Education & Workforce

Outcome: 1.3. **Business Ethics and Law**

Analyze how professional, ethical, and legal behavior contributes to continuous improvement in organizational performance and regulatory compliance.

Competencies

1.3.7. Identify the labor laws that affect employment and the consequences of noncompliance for both employee and employer (e.g., harassment, labor, employment, employment interview, testing, minor labor laws, Americans with Disabilities Act, Fair Labor Standards Acts, Equal Employment Opportunity Commission [EEOC]).

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

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

Outcome: 1.5. **Global Environment**

Evaluate how beliefs, values, attitudes and behaviors influence organizational strategies and goals.

Competencies

1.5.1. Describe how cultural understanding, cultural intelligence skills and continual awareness are interdependent.

1.5.2. Describe how cultural intelligence skills influence the overall success and survival of an organization.

1.5.3. Use cultural intelligence to interact with individuals from diverse cultural settings.

1.5.4. Recognize barriers in cross-cultural relationships and implement behavioral adjustments.

1.5.5. Recognize the ways in which bias and discrimination may influence productivity and profitability.

1.5.7. Use intercultural communication skills to exchange ideas and create meaning.

1.5.8. Identify how multicultural teaming and globalization can foster development of new and improved products and services and recognition of new opportunities.

Outcome: 1.7. **Entrepreneurship / Entrepreneurs**

Analyze the environment in which a business operates, and the economic factors and opportunities associated with self-employment.

Competencies

1.7.13. Protect intellectual property and knowledge (e.g., copyright, patent, trademark, trade secrets, processes).

Outcome: 1.8. **Operations Management**

Plan, organize and monitor an organization or department to maximize contribution to organizational goals and objectives.

Competencies

1.8.2. Select and organize resources to develop a product or a service.

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: 2.1. **Security, Risks, and Safeguards**

Describe the need for security and explain security risks and security safeguards.

Competencies

2.1.1. Explain the need for confidentiality, integrity, and availability (CIA) of information.

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).

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.

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, AI, cloud).

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, IPO charts, UML, decision trees, logic tables), pseudocode techniques and artificial intelligence.

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 and contrast 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 and contrast 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 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 best practices according to information security policies (e.g., cross-site scripting, Structured Query Language [SQL] injection attack, bounds_-checking).

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).
5.5.7. Read inputs (e.g., user input, data file, sensors, databases, APIs).

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 and contrast 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 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 9. Cybersecurity

Learners apply principles of Cybersecurity to secure and defend information technology systems, selection and implementation of methods and tools to secure physical and digital assets, mange threats, deploy countermeasures, and establish strategies to protect business information using risk and incident management.

Outcome: 9.1. Cybersecurity
Examine and employ principles of Cybersecurity.

Competencies

9.1.1. Identify the goals, objectives and purposes of cybersecurity.
9.1.4. Mitigate threats by remaining abreast of industry information.

Outcome: 9.3. Application Development Security

Develop and maintain application security.

Competencies

9.3.1. Identify application vulnerabilities (e.g., Cross-site scripting, SQL injection, LDAP injection, XML injection, Directory traversal/command injection, Buffer overflow, Integer overflow, Zero-day, Cookies and attachments, Locally Shared Objects (LSOs), Flash cookies, Malicious add-ons, Session hijacking, Header manipulation, Arbitrary code execution/remote code execution).

9.3.3. Implement secure coding concepts (e.g., Error and exception handling, Input validation, Cross-site scripting prevention, Cross-site Request Forgery, (XSRF) prevention, OWASP).

9.3.4. Implement secure application configuration (e.g., Application hardening, Application patch management).