STRANDS AND STANDARDS
COMPUTER PROGRAMMING 1

Course Description

An introductory course in program engineering and applications. The course introduces students to the fundamentals of computer programming. Students will learn to design, code, and test their own programs while applying mathematical concepts. Teachers introduce basic coding concepts and problem-solving skills.

Intended Grade Level	10 – 12
Units of Credit	0.5
Core Code	35.02.00.00.030
Concurrent Enrollment Core Code	35.02.00.13.030
Prerequisite	Suggested – Digital Literacy, Computer Science Principles, or Teacher Approval
Skill Certification Test Number	820, 941, 9830
Test Weight	0.5
License Area of Concentration	CTE and/or Secondary Education 6-12
Required Endorsement(s)	
Endorsement 1 OR	Intro to Computer Programming
Endorsement 2	Intro to Programming & Software Development
Endorsement 3	

ADA Compliant: August 2019

STRAND 1

Students will be familiar with and use a programming language IDE (Integrated DevelopmentEnvironment).

Standard 1

Demonstrate concept knowledge of different languages.

- Describe the difference between an interpreted language vs a compiled language.
- Identify characteristics of high-level and low-level languages.

Standard 2

Demonstrate the ability to use an IDE.

- Use an IDE to develop, compile, and run programs.
- Understand the difference between syntax, run-time, and logic errors.
- Use the debugger to identify errors.

Performance Skills

- Use an IDE to create a solution to solve a problem.

STRAND 2

Students will understand program development methodology.

Standard 1

Demonstrate the ability to use good programming style.

- Demonstrate proper use of white space (between lines and indentation).
- Use appropriate naming conventions for identifiers (variables, methods, functions,and file names).
- Construct identifiers with meaningful format; camelCase andunderscore.

Standard 2

Understand the software development life cycle.

- Identify specifications and understand requirements to create a solution to aproblem.
- Develop a program using external documentation (flowcharts, abstracts, andpseudocode) to break down the problem into sub-components.
- Design solutions using algorithms.
- Write the code to implement the algorithm.
- Test program for verification of errors and proper functionality.
- Provide internal comments in the IDE that explain functionality throughdocumentation (i.e comments, notes, program instructions)
- Redo all steps as needed.

Standard 3

Identify the components of a programming language syntax.

- Understand keywords, identifiers, operators, and operands.
- Understand statements and expressions in a program.
- Understand program components such as functions, methods, or procedures.

Performance Skills

- Demonstrate knowledge of program development methodology through a project.

STRAND 3

Students will demonstrate effective use of commands and operations.

Standard 1

Employ basic use of elements and data types of a programming language.

- Declare, initialize, and assign values to constants and variables.
- Demonstrate the ability to use input and output commands.
- Declare and use variable types (primitives, reference, or object).
- Identify proper data types for a specified application (boolean, integer, floatingpoint, strings).

Standard 2

Employ basic arithmetic expressions.

- Use basic arithmetic operators (modulus, multiplication, division, addition,subtraction).
- Understand order of operation of expressions.
- Write expressions that mix floating-point and integer expressions.

Performance Skills

- Demonstrate effective use of basic commands and operations.

STRAND 4

Students will properly employ control and loop structures.

Standard 1

Demonstrate the ability to use relational and logical operators in programs.

- Compare values using relational operators (<, >, ==, >=, <=, etc.)
- Form complex expressions using logical operators

Standard 2

Demonstrate the ability to use decisions in programs.

- Employ simple IF structures.
- Use IF-ELSE and nested IF-ELSE structures.

Standard 3

Demonstrate the ability to use loops in programs.

- Demonstrate knowledge between for-loops, while-loops, and do-while loops.
- Describe the various ways that loops can end (i.e., sentinel, break, condition fail,etc.).
- Design loops so they iterate the correct number of times (i.e., off by one errors, infiniteloops, etc.).
- Utilize nested loops.

Performance Skills

- Properly employ control and loop structures.

STRAND 5

Students will be aware of career opportunities in the Computer Programming/Software Engineering industry and ethical applications.

Standard 1

Investigate career opportunities, trends, and requirements related to computer programming/software engineering careers.

- Identify the members of a computer programming/software engineering team: team leader, analyst, senior developer, junior developer, and client/subject matter expert.
- Describe work performed by each member of the computer programming/software engineering team.
- Investigate trends and traits associated with computer programming/software engineering careers (creativity, technical, leadership, collaborative, problem solving, design, etc.).
- Discuss related career pathways.

Standard 2

Have an understanding of current ethical issues dealing with computer programming and information in society.

- Explain the impact software can have on society (i.e., privacy, piracy, copyright laws, ease of use, etc.).
- Explain the ethical reasons for creating reliable and robust software.
- Describe how computer-controlled automation affects a workplace and society.

Performance Skills

- Develop awareness of career opportunities in the computer programming/software engineering industry ethical applications.

Skill Certificate Test Points by Strand

Test Name	Test #	Number of Test Points by Strand										Total Points	Total Questions
		1	2	3	4	5	6	7	8	9	10		
Computer Programming 1	820	3	8	14	21	5						51	42