

Computer Programming 1

Exam Information	Description
Exam number 820 Items 40 National Career Cluster Information Technology Certification available Yes	The Computer Programming 1 industry certification exam assesses knowledge of foundational computer programming skills. Learners demonstrate their understanding to design, code, and test programs while applying mathematical concepts. Basic coding concepts and problem-solving skills are evaluated during the exam.

Exam Blueprint and Performance Skills

Standard	Percentage of exam score
1. Programming language IDE	10.87%
2. Program development methodology	15.22%
3. Key programming concepts	34.78%
4. Control structures	32.61%
5. Career Opportunities	2.17%
6. Ethical Issues and	4.35%

*Includes performance skill evaluation

Standard 1 - Programming language IDE

Candidates will be familiar with and use a programming language IDE (Integrated Development Environment).

Objective 1 Demonstrate concept knowledge of different languages.

1. Candidates will be aware of several programming languages and their purposes or use cases
 - a. Python - data science
 - b. Java - desktop application and server-side software
 - c. SQL - databases
 - d. JavaScript - web apps
 - e. C# - desktop applications
 - f. R - math and graphs
 - g. C++ - video games, desktop applications
 - h. C - operating systems, embedded systems
2. Describe the difference between an interpreted language vs a compiled language
3. Identify characteristics of high-level and low-level languages

Objective 2 Demonstrate the ability to use an IDE.

1. Use an IDE to develop, compile, and run programs
2. Understand the difference between syntax, run-time, and logic errors
 - a. Please see vocabulary list for specific definitions
3. Understand the process of debugging (finding and fixing errors)
 - a. Strategies candidates might learn for debugging could include:
 - Guess and Check
 - Deactivating sections to identify problematic code
 - Looking for typos, missing tags, or incorrect syntax
 - Making the problem smaller - identifying important points (changing variable values, getting input, etc.)
 - Asking a friend or team member for help
 - Printing, watching, or changing variable values while the program runs
 - Using a debugging tool
 - Thinking about when the code last worked and what has been added since then

Optional: Standard performance evaluations on the final page of this document

Standard 2 - Program Development Methodology

Candidates will understand program development methodology and best practices.

Objective 1 Demonstrate the ability to use a good programming style.

1. Demonstrate proper use of white space (between lines and indentation)
2. Use appropriate naming conventions for identifiers (variables, methods, functions, and file names)
3. Understand the appropriate use of constants versus variables in programming style
4. Construct identifiers with meaningful format; camelCase and underscore
5. Implement appropriate output formatting (decimal places, dollar signs, and correct placement of variable data in a sentence)

Objective 2 Understand the order of the software development life cycle.

1. Identify the members of a computer programming/software engineering team:
 - a. team leader
 - b. analyst
 - c. senior developer
 - d. junior developer
 - e. client/subject matter expert
2. Project Analysis: Identify specifications and understand requirements to create a solution to a problem
3. Planning/Design: Design an algorithm to solve the problem using appropriate documentation (single, multi-line, and/or documentation comments, flowcharts, etc.)
 - a. Define an algorithm
 - b. Break the problem down into its subcomponents using top-down design
4. Implementation: Write the code, with comments, to implement the algorithm
5. Testing: Test program for verification of errors and proper functionality
6. Release and Maintenance: Release the solution and provide updates when necessary

Optional: Standard performance evaluations on the final page of this document

Standard 3 - Key Programming Concepts

Candidates will understand and implement key programming concepts.

Objective 1 Understand and implement input and output commands.

1. Understand the difference between input and output

2. Understand there are different types of input (file, keyboard, mouse, microphone)
3. Understand there are different types of output (speakers, monitor, printer, file)
4. Write a program that receives input from a keyboard and produces output to the display

Objective 2 Understand and implement data types and variables.

1. Differentiate between primitive data types (boolean, integer, float and string)
2. Identify proper use of primitive data types (when to use one versus another)
3. Declare a variable and assign it a value using the assignment operator
4. Understand the difference between declaring and initializing a variable

Objective 3 Understand and implement operators and operands.

1. Use basic arithmetic operators (modulus [MOD or %], multiplication, integer division, float division, addition, subtraction)
3. Use basic comparison operators (<, >, ==, >=, <=)
4. Use basic assignment operator (=)
5. Understand order of operations for all operators
 - a. Parenthesis
 - b. Exponent
 - c. Multiplication
 - d. Division
 - e. Modulus [MOD or %]
 - f. Addition
 - g. Subtraction
6. Use basic logical operators (AND, OR, NOT)
7. Use operands in conjunction with arithmetic, relational, and logical operators

Objective 4 Understand and implement expressions in a program.

1. Understand how operators and operands are used to form expressions
2. Identify and implement syntactically correct expressions
 - a. Possible examples: A OR B, 5==6, x !=3.142, x = 4, y + 7

Objective 5 Understand and implement functions.

1. Understand and properly define scope, local variable, and global variable
2. Understand what functions are and what they are used for (readability, reusability, modularity, abstraction)
3. Understand the difference between a built-in function and user defined function
4. Utilize built-in functions
5. Understand that functions may or may not require arguments (input(s))
6. Understand that functions may or may not return value(s) (output(s))

Objective 6 Understand and implement complex data types.

1. Understand the difference between simple and complex/primitive and reference data types
2. Declare a string variable in a program

Optional: Standard performance evaluations on the final page of this document.

Standard 4 - Control Structures

Understand and implement IF statements in a program.

Objective 1 Understand and implement IF statements in a program.

1. • IF
 - a. Understand when to use an IF statement
 - b. Demonstrate correct use of an IF statement
2. • ELSE-IF
 - a. Understand when to use an ELSE-IF statement
 - b. Demonstrate correct use of ELSE-IF statements
3. • ELSE
 - a. Understand when to use an ELSE statement
 - b. Demonstrate proper use of an ELSE statement
4. • Nesting IF statements
 - a. Understand when to use a nested IF statement
 - b. Demonstrate proper use of a nested IF statement
5. Candidates will be able to read a program and know which if statement will execute during a given situation and which will be ignored

Objective 2 Understand and implement basic loop structures in programs.

1. For-loops
 - a. Understand when to use a for-loop
 - When you know the number of times you are going to iterate
 - b. Understand the three components of a for-loop
 - An initial value ($i = 0$)
 - A condition ($i < 7$)
 - An update expression ($i = i + 1$)
 - c. Demonstrate proper use of for-loops
2. While-loops
 - a. Understand when to use a while-loop
 - When you want to stop iterating when a boolean condition is no longer true
 - b. Demonstrate proper use of a while-loop
3. Nested loops (loop within a loop)
 - a. Understand when to use nested loops
 - b. Demonstrate proper use of nested loops
4. Identify the various ways that loops can end (break, met condition, condition fail)
5. Design loops so they iterate the correct number of times

6. Understand what causes an infinite loop

Objective 3 Understand and implement expressions and complex conditions in control structures.

1. Create expressions using relational operators
 - a. Example: (a > 6, x != 7, y > 4)
2. Form complex conditions using logical operators
 - a. Example: (a > 6 AND x != 7 OR y > 4)
3. Incorporate complex conditions in loop structures
 - a. Example: While a player's health is greater than 50 and player is not dead
4. Understand a Truth Table

ex:)

A	B	A && B	A B
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Optional: Standard performance evaluations on the final page of this document

Standard 5 - Career Opportunities

Candidates will be aware of career opportunities in the Computer Programming/Software Engineering Industry.

Objective 1 Investigate career opportunities, trends, and requirements related to computer programming/software engineering careers.

1. Understand the role each team member plays in a computer programming/software engineering team:
 - a. team leader
 - b. analyst
 - c. senior developer
 - d. junior developer
 - e. client/subject matter expert
2. Describe work performed by each member of the computer programming/software engineering team
3. Investigate trends and traits associated with computer programming/software engineering careers (creativity, technical, leadership, collaborative, problem solving, design, etc.)

4. Discuss related career pathways

Optional: Standard performance evaluations on the final page of this document

Standard 6 - Ethical Issues &

Candidates will be aware of the ethical issues in computer programming and the impact AI is making on the industry.

Objective 1 This is a statement from the USBE Board on Artificial Intelligence. Candidates will develop the knowledge and skills to use Artificial Intelligence responsibly and effectively in CTE courses.

1. Responsible use of AI, which includes recognizing when to use AI, using appropriate prompts to produce desired output, determining accuracy of information, how AI has contributed to a product, and communicating that clearly.
2. The importance of transparency and documentation, both in their own use of AI tools and in verifying the authenticity of others' work.
3. How AI influences all work, including the evolving boundaries and tightening restrictions around intellectual property and copyright.

Objective 2 Understand current ethical issues dealing specifically with computer programming and information in society.

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

Objective 3 AI use in Programming: Candidates will understand the ethical use of AI in coding and how it impacts the programming field.

1. Academic integrity and attribution
 - a. Know when AI use is appropriate
 - b. Explain why you should not rely only on the use of AI to write your code
 - c. Understand disclosing when you use AI
2. Understanding vs. over-reliance on AI
3. Verifying and debugging AI generated code
 - a. Reviewing the code to make sure there are no coding errors
 - b. Understand that AI tools can be helpful for debugging code, but it's important to double-check their suggestions using your own programming knowledge
4. Ethical implication of AI written software

Optional: Standard performance evaluations on the final page of this document.

Performance Skills Evaluation

Exam name: Computer Programming 1

Candidate Name: _____

Performance assessments may be completed and evaluated any time before the certification is issued. The following performance skills can be used in connection with the associated standards and exam. The evaluator will function as the subject matter expert to determine if the candidate meets the skills requirements with a Yes/No for passing.

Standard 1 – Programming Language IDE • Use an IDE to create a solution to solve a problem.	Pass:
Standard 2 – Program Development Methodology • Demonstrate knowledge of program development methodology through a project.	Pass:
Standard 3 – Key Programming Concepts • Write one or more programs that demonstrate effective use of the key programming concepts defined in Strand 3	Pass:
Standard 4 – Control Structures • Write one or more programs that demonstrate effective use of control structures.	Pass:
Standard 5 – Career Opportunities • Develop awareness of career opportunities in the computer programming/software engineering industry.	Pass:
Standard 6 – Ethical Issues and • Develop an awareness of all the ethical implications of Artificial Intelligence (AI)	Pass:

Evaluator Name: _____

Date: _____