



COMPUTER SCIENCE

Test Framework

	Content Domain	Range of Competencies	Approximate Percentage of Test Score
I.	Computational Thinking and Data Analysis	0001–0003	28%
II.	Programming Concepts	0004–0007	36%
III.	Computing Systems, Networks, and the Internet	0008–0009	18%
IV.	Impacts of Computing and Learning Environments	0010–0011	18%

Copyright © 2017 Pearson Education, Inc. or its affiliate(s). All rights reserved.
Evaluation Systems, Pearson, P.O. Box 226, Amherst, MA 01004

NES, the NES logo, Pearson, the Pearson logo, and National Evaluation Series are trademarks, in the U.S. and/or other countries, of Pearson Education, Inc. or its affiliate(s).

This document may not be reproduced for commercial use but may be copied for educational purposes.

I. COMPUTATIONAL THINKING AND DATA ANALYSIS

0001 Understand problem solving and algorithm development.

- ▶ Demonstrate knowledge of the iterative problem-solving process.
- ▶ Apply knowledge of decomposing a problem into simpler parts.
- ▶ Apply characteristics of algorithmic thinking, including the use of flowcharts and pseudocode.
- ▶ Demonstrate knowledge of the use of sequence, selection, iteration, and recursive thinking in algorithm design.
- ▶ Demonstrate knowledge of the concept of abstraction for managing program complexity.
- ▶ Demonstrate knowledge of how program components fit together to complete the problem-solving process.

0002 Understand characteristics of algorithms.

- ▶ Demonstrate knowledge of the practical application of algorithms.
- ▶ Interpret flow of control in an algorithm.
- ▶ Predict output of a given algorithm (e.g., program tracing).
- ▶ Select an algorithm to solve a given problem.
- ▶ Evaluate an algorithm in terms of correctness.
- ▶ Evaluate algorithms in terms of complexity and efficiency, including a basic interpretation of big-O notation.
- ▶ Apply knowledge of the characteristics and uses of searching (i.e., linear and binary) and sorting algorithms (e.g., bubble, selection).

0003 Understand data analysis, modeling, and simulation.

- ▶ Demonstrate knowledge of tools and techniques for locating, collecting, and storing small-scale and large-scale data sets from a variety of sources (e.g., files, database).
- ▶ Identify data and use a variety of tools (e.g., database, spreadsheet, algorithms) and techniques (e.g., search, filter, transform, visualize) to analyze data and identify patterns
- ▶ Demonstrate knowledge of the use of models and representations to describe and evaluate processes and phenomena.
- ▶ Apply knowledge of computer models and data analysis techniques for analyzing and simulating real-world problems.

II. PROGRAMMING CONCEPTS

0004 Understand programming concepts and program design and development.

- ▶ Apply knowledge of characteristics of programming languages (e.g., functional, compiled, object oriented), including block-based languages.
- ▶ Apply knowledge of the software development process (e.g., design, testing, maintenance) and methodologies (e.g., waterfall, iterative, agile).
- ▶ Apply knowledge of programming style and good programming practices (e.g., indenting, spacing, comments, naming conventions, camelCase) and program documentation.
- ▶ Apply knowledge of tools related to the development of computer artifacts (e.g., IDEs, APIs, libraries, mobile device simulators).
- ▶ Apply knowledge of common programming errors and procedures for debugging computer programs.
- ▶ Demonstrate knowledge of a variety of concepts related to programming (e.g., event driven, heuristic algorithms, parallel processing, artificial intelligence).

0005 Understand characteristics and uses of data types.

- ▶ Demonstrate knowledge of language-defined data types (e.g., integer, float, Boolean).
- ▶ Apply properties of strings and string methods (e.g., length, substring, concatenation).
- ▶ Apply knowledge of constants, variables, and classes in various contexts.
- ▶ Apply a basic knowledge of data structures (e.g., one-dimensional arrays).

0006 Understand operators and control structures.

- ▶ Apply arithmetical operators as defined by the language (e.g., addition, subtraction, multiplication, integer division, modular arithmetic).
- ▶ Apply relational operators in various contexts (e.g., greater than, less than, equal to).
- ▶ Interpret logical operators in various contexts (e.g., AND, OR, NOT).
- ▶ Apply knowledge of conditional selection structures (e.g., if, if-else), including nesting.
- ▶ Apply knowledge of iterative control structures (e.g., while, for), including nesting.

0007 Understand concepts of object-oriented design and programming.

- ▶ Demonstrate knowledge of principles of modularization and characteristics of program modules (e.g., functions/methods, objects, classes).
- ▶ Analyze the characteristics and uses of inheritance and classes in object-oriented programming.
- ▶ Demonstrate knowledge of concepts and principles related to object-oriented design and programming (e.g., abstraction, information hiding, encapsulation, constructors).
- ▶ Apply knowledge of function calls, parameters, and parameter-passing techniques.

III. COMPUTING SYSTEMS, NETWORKS, AND THE INTERNET

0008 Understand terminology and concepts related to computing systems.

- ▶ Demonstrate knowledge of how information (e.g., text, image, sound) can be represented digitally by binary data.
- ▶ Demonstrate knowledge of characteristics of binary, decimal, and hexadecimal number systems.
- ▶ Demonstrate knowledge of basic computer architecture and peripherals (e.g., processors, memory, storage, sensors).
- ▶ Demonstrate knowledge of the characteristics and functions of operating systems.
- ▶ Apply knowledge of strategies for troubleshooting basic hardware and software problems.

0009 Understand networks and the Internet.

- ▶ Demonstrate knowledge of terminology and concepts related to computer networks (e.g., client, server, bandwidth, protocol).
- ▶ Identify the basic structure and features of the Internet.
- ▶ Demonstrate knowledge of concepts related to data transfer on the Internet (e.g., routing, packet switching, https).
- ▶ Demonstrate knowledge of terminology and concepts related to mobile technologies (e.g., apps, wireless connectivity, security) and the interaction between mobile devices and networks.
- ▶ Demonstrate knowledge of cybersecurity issues related to networks and the Internet (e.g., firewalls, data encryption, malware).

IV. IMPACTS OF COMPUTING AND LEARNING ENVIRONMENTS

0010 Understand social and global issues related to computer technology.

- ▶ Demonstrate knowledge of significant events (e.g., vacuum tube, transistors, integrated circuit, Turing test) and influential contributors (e.g., Charles Babbage, Ada Lovelace, Grace Hopper) in the history of computer science.
- ▶ Demonstrate knowledge of the responsible use of technology, social media, and digital citizenship, including appropriate etiquette, cyberbullying, and consequences of misuse.
- ▶ Demonstrate knowledge of the impact of technology on society (e.g., environment, economy, financial markets, medicine, education) and the influence of society on technology.
- ▶ Demonstrate knowledge of issues related to the ethical and legal use of computers and information (e.g., privacy, confidentiality, information sharing, intellectual property rights, hacking).
- ▶ Demonstrate knowledge of issues related to the equitable use of technology (e.g., language, disabilities, access to technology, culture, socioeconomic status).

0011 Understand effective learning environments.

- ▶ Demonstrate knowledge of issues related to the safe and effective use of technology and digital citizenship, including online safety and privacy of electronic student information.
- ▶ Apply knowledge of principles and concepts related to an inquiry-based computer science curriculum.
- ▶ Apply knowledge of issues related to diversity in computer science education and strategies for fostering an inclusive computing learning environment.
- ▶ Apply knowledge of developmentally appropriate instructional strategies and practices to support the diverse needs of all learners.
- ▶ Demonstrate knowledge of effective approaches for promoting collaboration and developing communication skills.
- ▶ Demonstrate knowledge of issues related to mobile computing, computer literacy, computational thinking, and collaboration and communication in computer science.