National Evaluation Series™

NES PROFILE

Computer Science (315)

Pearson

NES Profile: Computer Science (315)

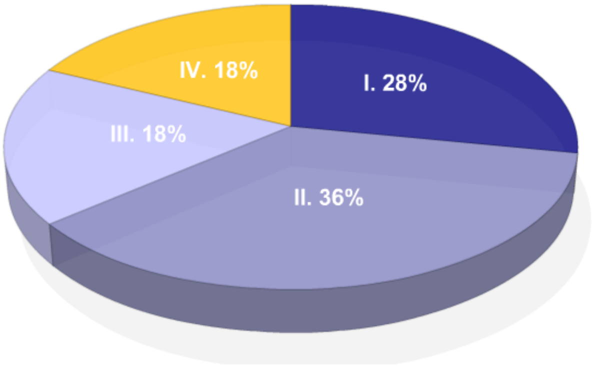
Overview

The resources below provide information about this test, including the approximate percentage of the total test score derived from each content domain. The complete set of the content domains, the test framework, is provided here and contains all of the competencies and descriptive statements that define the content of the test.

This *NES Profile* includes the following materials:

- the test competencies associated with the content domain,
- a set of descriptive statements that further explain each competency,
- a sample test question aligned to each competency.

Test Field	Computer Science (315)
Test Format	Multiple-choice questions
Number of Questions	Approximately 100
Test Duration	Up to 2 hours
Reference Materials	NES Computer Science Style Conventions



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

Content Domain I: Computational Thinking and Data Analysis

Competency:

0001 Understand problem solving and algorithm development.

Descriptive Statements:

- 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.

Sample Item:

The iterative design process would be most appropriately used in which of the following situations?

A. A large budget for project designers is available.

B. A small project must be completed quickly by a single person.

C. The project is well defined and has strict specifications.

D. The major project requirements are known but details can evolve over time.

Correct Response and Explanation
D. The iterative design process is a cyclic approach to designing, testing, analyzing, and refining a project. In each iteration, modifications can be made, incorporating details that change during the course of the project. Once the modifications are made, the project is again tested, analyzed, and refined.

Competency:

0002 Understand characteristics of algorithms.

Descriptive Statements:

- 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).

Sample Item:

Use the pseudocode below to answer the question that follows.

```
function(list) // index of a list starts at zero //
    len = list.length() // finds the length of list //
    answer = 0
    for (i = 0, i <= len - 1, i = i + 1)
        answer = answer + list[i]
    answer = answer / len
    print(answer)
```

What will `function([3, 4, 5])` **print?**

A. `3`

B. `4`

C. `5`

D. `6`

Correct Response and Explanation
B. There are 3 elements in the list, so `len = 3` and the for loop will iterate when `i = 0, 1, 2`. At each iteration, the list element is added to the value of `answer`, which is initially assigned the value of `0`. When the loop is complete, the quotient of `answer` and `len` is computed. The trace table for the loop is shown below.

	i	answer+ list[i]	answer
Initial	0	0	0
Loop 1	0	0+3	3
Loop 2	1	3+4	7
Loop 3	2	7+5	12

When the loop terminates, `answer = 12` and `len = 3`, therefore `12 / 3 = 4`.

Competency:

0003 Understand data analysis, modeling, and simulation.

Descriptive Statements:

- 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.

Sample Item:

Students have access to a data set that relates the air pressure as measured by a barometer to the amount of precipitation (rain) for the area in which they live. The students would like to create a computer program to analyze the data to determine whether knowing air pressure can help predict rain. Building this program requires application of which of the following mathematical concepts?

A. quadratic functions

B. complex numbers

C. regression models

D. Pythagorean triples

Correct Response and Explanation
C. In statistics, regression models are used to analyze the relationship between a dependent variable and an independent variable. They are commonly used to make predictions by estimating the expected value of the dependent value, in this case the chance of rain, given the independent variable (air pressure). Linear regression models can produce the equation of the line that best fits the data, along with a measure of how well the two variables are correlated.

Content Domain II: Programming Concepts

Competency:

0004 Understand programming concepts and program design and development.

Descriptive Statements:

- 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).

Sample Item:

When a smartphone manufacturer releases a new feature, such as fingerprint scanning, the manufacturer needs to provide developers with the ability to incorporate this new feature into their software. This is typically accomplished through the use of:

A. Open Systems Interconnection (OSI) model.

B. graphical user interface (GUI).

C. application programming interface (API).

D. Cascading Style Sheets (CSS).

Correct Response and Explanation
C. An API is a set of requirements that determine how one application (app) can talk to another. APIs allow developers limited access to a program's internal functions. In this case, a smartphone manufacturer would typically provide enough access for mobile device app developers to be able to integrate the fingerprint scanner into their apps without exposing all of the fingerprinting application code.

Competency:

0005 Understand characteristics and uses of data types.

Descriptive Statements:

- 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).

Sample Item:

Use the information below to answer the question that follows.

```
// string.characterAt(i), returns the character at index i. String
index begins at zero //
email = "last.first@school.edu"
```

Which of the following methods will return `@`?

A. `email.characterAt(8)`

B. `email.characterAt(9)`

C. `email.characterAt(10)`

D. `email.characterAt(11)`

Correct Response and Explanation
C. In the string `last.first@school.edu`, the `@` is the 11th character in the string. However, since the string index begins at `0`, the index of the `@` is `10`.

Competency:

0006 Understand operators and control structures.

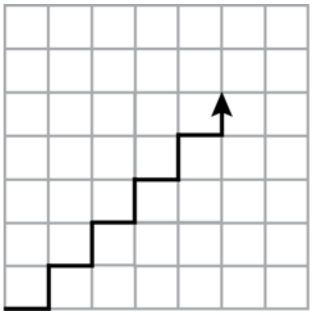
Descriptive Statements:

- 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.

Sample Item:

Use the information below to answer the question that follows.

The diagram above shows the path taken by a robot that starts facing in the positive y-direction at the point (0, 0) on a coordinate grid. It can move forward n units (`Forward n`), turn left 90 degrees (`Turn Left`), or turn right 90 degrees (`Turn Right`).



Which of the following pseudocode segments will create the path shown in the diagram?

A.
```
i = 2
while (i < 12)
    if i is even
        Turn Right
        Forward 10
    else
        Turn Left
        Forward 10
    i = i + 1
```

B.
```
i = 2
while (i <= 12)
    if i is even
        Turn Right
        Forward 10
    else
        Turn Right
        Forward 10
    i = i + 1
```

C.

```
i = 2
while (i < 12)
    if i is even
        Forward 10
        Turn Right
    else
        Forward 10
        Turn Left
i = i + 1
```

D.

```
i = 2
while (i <= 12)
    if i is even
        Forward 10
        Turn Right
    else
        Forward 10
        Turn Left
    i = i + 1
```

Correct Response and Explanation

A. In this question, the robot begins at (0, 0), facing in the positive *y*-direction. To follow the path, the robot must first turn to the right and move `10` units in the positive *x*-direction, then turn left and move `10` units in the positive *y*-direction. This is accomplished by turning right when `i` is even and left when `i` is odd. Since `i` begins at `2`, and the robot travels for `10` segments, the values for `i` will be `2` through `11`; the loop will not execute once `i = 12`.

Competency:

0007 Understand concepts of object-oriented design and programming.

Descriptive Statements:

- 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.

Sample Item:

Use the pseudocode below to answer the question that follows.

```
CLASS Flower

    private integer numOfPetals
    private string color
    private bloom()
    public grow()

    SUBCLASS Rose inherits from Flower

        Rose(numOfPetals, color)

MAIN


    Rose r1 = Rose(5, "red")
```

Which of the following demonstrates how to access an element of the class `Rose`**?**

A. `r1.numOfPetals`

B. `r1.color`

C. `r1.bloom()`

D. `r1.grow()`

Correct Response and Explanation

D. In the program, `r1` is an object, or instance of class `Rose`. `Rose` is a subclass of `Flower`, and inherits from it. Since `Rose` is a subclass, it can only access the public variables and methods, in this case the public method, `grow()`. Therefore, `r1.grow()` is the only way to access an element of `r1`.

Content Domain III: Computing Systems, Networks, and the Internet

Competency:

0008 Understand terminology and concepts related to computing systems.

Descriptive Statements:

- 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.

Sample Item:

The hexadecimal number 4F is equivalent to what decimal number?

A. 19

B. 79

C. 154

D. 415

Correct Response and Explanation
B. In hexadecimal notation, the symbols 0 – 9 are used to represent the numbers 0 – 9 and the symbols A – F represent the numbers 10 – 15. Hexadecimal is a base-16 system. The hexadecimal number 4F can be converted to a decimal number as follows: $4 \text{ x } 16^1 + 15 \text{ x } 16^0 = 4(16) + 15(1) = 64 + 15 = 79$.

Competency:

0009 Understand networks and the Internet.

Descriptive Statements:

- 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).

Sample Item:

A client-side scripting language would most likely be used for which of the following tasks?

A. rerouting messages

B. storing information in a database

C. adding interactivity to a web page

D. how to track print executing code on a web server

Correct Response and Explanation
C. In client-side scripting, source code is transferred from the web server to a user's computer over the Internet and run directly in the browser. Client-side scripting allows a web page to behave differently in response to mouse or keyboard actions. Rollover buttons, pop-up balloons, and filters for search criteria are all examples of interactive features that can be produced with client-side scripting.

Content Domain IV: Impacts of Computing and Learning Environments

Competency:

0010 Understand social and global issues related to computer technology.

Descriptive Statements:

- 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).

Sample Item:

Which of the following statements best describes how Grace Hopper contributed to the field of computer science?

A. She combined transistors, capacitators, and other components onto a single chip.

B. She designed the logic gate architecture to perform various computational operations.

C. She wrote the source code for the protocol used for early networks.

D. She developed one of the first high-level programming languages.

Correct Response and Explanation
D. Grace Hopper (1906–1992) was a rear admiral in the United States Navy and a computer scientist. She invented one of the first compiler-related tools and her work on machine-independent programming languages led to the development of COBOL.

Competency:

0011 Understand effective learning environments.

Descriptive Statements:

- 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.

Sample Item:

A computer science curriculum that begins with a unit on problem solving instead of beginning with computer programming is likely to foster inclusive learning primarily because beginning with problem solving:

A. provides students with opportunities to succeed regardless of their previous coding experience.

B. teaches students the importance of working collaboratively in computer science.

C. allows students without previous coding experience to take extra time to pre-learn code syntax.

D. helps students of all backgrounds and abilities determine whether computer science is of personal interest.

Correct Response and Explanation
A. Problem solving is a necessary skill in computer science and many other disciplines. When a unit begins with problem solving, students have the opportunity to engage with material that is concrete and relevant, and to use strategies with which they may already be familiar. This helps students build confidence in their ability to solve problems before moving on to coding computer projects.