

Valid Foundations-of-Computer-Science Test Dumps | Foundations-of-Computer-Science Practice Test Engine



2026 Latest TorrentExam Foundations-of-Computer-Science PDF Dumps and Foundations-of-Computer-Science Exam Engine Free Share: https://drive.google.com/open?id=1OCFIIBk5q3ZZfãYPk8qD_w4anCZ5-81

Studying for attending Foundations-of-Computer-Science exam pays attention to the method. The good method often can bring the result with half the effort, therefore we in the examination time, and also should know some test-taking skill. The Foundations-of-Computer-Science quiz guide on the basis of summarizing the past years, the answers have certain rules can be found, either subjective or objective questions, we can find in the corresponding module of similar things in common. To this end, the Foundations-of-Computer-Science Exam Dumps have summarized some types of questions in the qualification examination to help you pass the Foundations-of-Computer-Science exam.

There are a lot of leading experts and professors in different field in our company. The first duty of these leading experts and professors is to compile the Foundations-of-Computer-Science exam questions. In order to meet the needs of all customers, the team of the experts in our company has done the research of the Foundations-of-Computer-Science Study Materials in the past years. And they have considered every detail of the Foundations-of-Computer-Science practice braindumps to be perfect. That is why our Foundations-of-Computer-Science learning guide enjoys the best quality in the market!

>> Valid Foundations-of-Computer-Science Test Dumps <<

Foundations-of-Computer-Science Practice Test Engine - New Foundations-of-Computer-Science Test Topics

Our exam prep material is famous among WGU exam candidates which help to polish the knowledge required to pass the WGU Foundations of Computer Science exam. The certification is organized by WGU internationally. Our WGU Foundations of Computer Science (Foundations-of-Computer-Science) exam questions are the most cost-effective as we understand that you need low-cost material but are authentic and updated. TorrentExam provides its WGU Foundations-of-Computer-Science Exam Questions in three forms, one is PDF eBook, the second is practice exam software for Windows-based systems, and the third is an online practice test.

WGU Foundations of Computer Science Sample Questions (Q52-Q57):

NEW QUESTION # 52

What is another term for the inputs into a function?

- A. Variables
- B. Outputs
- C. Procedures
- D. Arguments

Answer: D

Explanation:

In programming, a function takes inputs, performs computation, and may return an output. The standard term for a function's inputs is arguments (also commonly discussed alongside the closely related term parameters).

Textbooks typically distinguish the two: parameters are the names listed in the function definition, while arguments are the actual values supplied when the function is called. For example, in `def f(x, y):`, `x` and `y` are parameters. In the call `f(3, 5)`, 3 and 5 are arguments. Many introductory materials use "arguments" informally to refer to the inputs overall, which matches the wording of this question.

Options A, B, and C do not fit the textbook definition. "Variables" is too broad; inputs can be literals, expressions, or variables, but the conceptual role is "arguments." "Procedures" are callable units of code (often used in some languages to mean functions without return values), not the inputs. "Outputs" refers to returned results, not what you pass in.

Understanding arguments is important because it connects to call semantics, scope, and correctness.

Different languages support positional arguments, keyword arguments, default values, and variadic arguments (e.g., `*args`, `**kwargs` in Python). This flexibility shapes API design and influences how programmers structure reusable code.

NEW QUESTION # 53

What is an ndarray in Python?

- A. A built-in Python data array used to store collections of items.
- B. A module that provides network socket functions similar to XML.
- **C. An n-dimensional array object provided by the NumPy library.**
- D. A native Python object that represents a tree-like hierarchical data structure.

Answer: C

Explanation:

An ndarray is NumPy's fundamental data structure: an n-dimensional array designed for efficient numerical computation. The term stands for "N-dimensional array," and it is implemented as `numpy.ndarray`. Unlike Python's built-in list, an ndarray stores elements in a compact, homogeneous format defined by its dtype (such as integers or floating-point numbers). This uniform representation enables fast, vectorized operations and efficient use of memory, which is why ndarray is central in scientific computing and data analysis.

An ndarray supports multiple dimensions: a 1D array behaves like a vector, a 2D array like a matrix (rows and columns), and higher-dimensional arrays represent tensors. Textbooks emphasize that ndarray operations are typically element-wise by default (for example, `a + b` adds corresponding elements), and that slicing and broadcasting allow powerful computations without explicit loops. This approach is both expressive and efficient because the heavy lifting happens in optimized low-level code.

Option A is incorrect because ndarray is not built into core Python; it comes from NumPy. Option B describes a tree, which is a different data structure entirely. Option D is incorrect because sockets and XML-related functionality belong to other parts of Python's standard library, not to NumPy or ndarray.

In short, an ndarray is the primary array object of NumPy, providing high-performance multi-dimensional numerical storage and computation.

NEW QUESTION # 54

What is the time complexity of a quicksort algorithm?

- **A. $O(n \log n)$**
- B. $O(n)$
- C. $O(\log n)$
- D. $O(1)$

Answer: A

Explanation:

Quicksort is a divide-and-conquer sorting algorithm. It works by selecting a pivot element, partitioning the array into two subarrays (elements less than the pivot and elements greater than the pivot), and then recursively sorting those subarrays. In the average case, the partition step splits the array into roughly equal halves, so the recurrence is commonly written as $T(n) = T(n/2) + T(n/2) + O(n)$, where $O(n)$ is the cost of partitioning. This solves to $O(n \log n)$, which is why quicksort is widely taught as an efficient general-purpose sorting method.

However, textbooks also emphasize that quicksort has a worst-case time complexity of $O(n^2)$.

P.S. Free & New Foundations-of-Computer-Science dumps are available on Google Drive shared by TorrentExam:
https://drive.google.com/open?id=1OCFIIBk5q3ZZfaYPkqB8qD_w4anCZ5-81