PCEP Certified Entry-Level Python Programmer (PCEP-30-0X) Practice Exam (Sample)

Study Guide

Everything you need from our exam experts!

Questions

1. Which function is designed to receive data from the console?

A. input()

B. output()

C. read()

D. print()

2. Which keyword is used to check if a specific key exists in a dictionary?

A. exists

B. is

C. in

D. contains

3. Which of the following is NOT one of the four fundamental elements that makes a programming language?

A. Alphabet

B. Syntax

C. Colors

D. Semantics

4. What is the name of the tool that allows you to execute your code step-by-step and inspect it at each moment?

A. Debugger

B. Compiler

C. Interpreter

D. Code Editor

5. What type of result does the division operator produce?

A. An Integer

B. A Float

C. A String

D. A Boolean

6. What is a sequence type in Python?

A. A variable that can change

B. A data structure holding an ordered collection of items

C. A function that returns multiple values

D. A type of conditional statement

7. Which command would you use to delete an entire dictionary?

A. delete dictionary_name

B. remove dictionary_name

C. del dictionary_name

D. clear dictionary_name

8. What is the term for the area where a variable is defined and can be accessed?

A. Scope

B. Range

C. Limit

D. Domain

9. What is the name given to source programs that are written in scripting languages?

A. Scripts

B. Modules

C. Functions

D. Libraries

10. What are fixed values written directly in the code that represent constant data called?

A. Variable

B. Reference

C. Literal

D. Constant

Answers

1. A
2. C
3. C
4. A
5. B
6. B
7. C
8. A
9. A
10. C

Explanations

1. Which function is designed to receive data from the console?

A. input()

B. output()

C. read()

D. print()

The function that is specifically designed to receive data from the console is the input() function. This function allows the user to enter information during the execution of a program. When called, input() prompts the user to type something on the keyboard and then captures the entered data as a string. For example, invoking input() might look like this: ```python user_input = input("Please enter your name: ") ``` In this case, the program will display the message "Please enter your name: " and will wait for the user to type their name. Once the user hits enter, the entered data is returned and can be stored in a variable for further use. The other options do not serve this purpose. The output() function does not exist in standard Python and is not used for receiving data. The read() function may refer to file handling or reading data, but it is not used to obtain user input from the console. The print() function, on the other hand, is used to display output to the console rather than to receive input. Thus, input() is the only correct choice for capturing user data from the console.

2. Which keyword is used to check if a specific key exists in a dictionary?

A. exists

B. is

C. in

D. contains

The correct choice is "in," which is the keyword used in Python to check for the existence of a specific key in a dictionary. When you use "in," you can directly evaluate whether a key is present in the dictionary's keys. For example, if you have a dictionary named `my_dict`, you can check if a key called `key_name` exists by using the expression `key_name in my_dict`. This will return `True` if the key is found and `False` otherwise. The other options don't serve this purpose in Python. "exists" and "contains" are not keywords in Python that can be used to check for the presence of keys in dictionaries. "is" is a keyword used for identity comparisons, checking if two references point to the same object in memory, and does not relate to key existence in dictionaries. Thus, "in" is the appropriate keyword for this operation.

3. Which of the following is NOT one of the four fundamental elements that makes a programming language?

A. Alphabet

B. Syntax

C. Colors

D. Semantics

The four fundamental elements that make a programming language typically include the alphabet, syntax, semantics, and sometimes a fourth element like pragmatics or a similar concept. The alphabet refers to the set of symbols or characters that can be used in the programming language, such as letters, digits, and special characters. Syntax involves the rules governing how those symbols can be combined to create valid statements or expressions. Semantics deals with the meanings of those statements—what happens when a particular statement is executed. Colors, on the other hand, do not constitute a fundamental aspect of programming languages. While some programming languages might use colors in their syntax highlighting to aid programmers in readability and understanding, the concept of color itself is not inherent to the structure or functionality of a programming language. Thus, this option does not fit within the core elements of a programming language, confirming it as the correct answer.

4. What is the name of the tool that allows you to execute your code step-by-step and inspect it at each moment?

A. Debugger

B. Compiler

C. Interpreter

D. Code Editor

The correct answer is the debugger because it is specifically designed to allow developers to execute their code line-by-line, monitor the state of their program, and inspect variables and data at any moment during execution. This tool is invaluable for identifying and resolving bugs or logical errors in the code. With debugging capabilities, programmers can pause execution, step through the code, and evaluate the behavior of the program in real-time. Compilers are tools that translate code written in a programming language into machine code but do not provide interactive execution and inspection features. Interpreters execute code directly but typically do not offer the detailed step-by-step inspection capabilities that a debugger does. Code editors are environments where you write code but do not inherently include debugging features unless they are equipped with debugging tools.

5. What type of result does the division operator produce?

A. An Integer

B. A Float

C. A String

D. A Boolean

The division operator in Python, represented by the symbol "/", always produces a floating-point number as a result, regardless of whether the operands are integers or floating-point values. This means that even if both operands are integers, the result will still be a float. For instance, if you perform a division like this: ```python result = 5 / 2 ``` The result will be `2.5`, which is a float. This behavior is designed to prevent loss of precision that could happen if integer division were always performed, especially useful in scientific calculations and other applications reliant on decimal precision. The operating behavior of the division operator highlights Python's emphasis on providing more intuitive and useful results in mathematical computations. Understanding that division leads to a floating-point result is key in writing and debugging Python code, ensuring that you manage the data types returned from operations correctly and avoid unexpected behaviors when performing calculations.

6. What is a sequence type in Python?

A. A variable that can change

B. A data structure holding an ordered collection of items

C. A function that returns multiple values

D. A type of conditional statement

A sequence type in Python is indeed defined as a data structure that holds an ordered collection of items. This includes built-in types such as lists, tuples, and strings. Sequence types are characterized by their ability to store a collection of items in a specific order, allowing for indexing and iteration through the items. For instance, in a list, you can access items by their index (e.g., `my_list[0]` retrieves the first item). This ordered nature allows for predictable retrieval and manipulation of data. Additionally, operations like slicing are available on sequence types, enabling you to obtain sub-portions of the collection effectively. The other options refer to different concepts within Python. A variable that can change refers to mutable vs. immutable types, a function that returns multiple values is typically handled using tuples or other structures, and a type of conditional statement pertains to control flow, which is unrelated to the concept of sequences. Thus, choice B accurately describes what a sequence type is in Python.

7. Which command would you use to delete an entire dictionary?

A. delete dictionary_name

B. remove dictionary_name

C. del dictionary_name

D. clear dictionary_name

The command used to delete an entire dictionary in Python is "del dictionary_name." This command removes the specified dictionary from memory, meaning that the variable name that referred to the dictionary will no longer be valid, and trying to access it afterward will result in an error. Deleting a dictionary using this command effectively clears all references to the dictionary and frees up the memory it was using. It is important to note that this action is irreversible; once the dictionary is deleted, all the key-value pairs it contained are also lost. The other options do not perform the same function. For instance, "delete" and "remove" are not valid Python commands for deleting a dictionary; Python does not recognize these terms in this context. Moreover, the "clear" method is used to remove all items from the dictionary but retains the dictionary itself. The dictionary would still exist as an empty dictionary after using "clear," rather than being completely deleted from memory.

8. What is the term for the area where a variable is defined and can be accessed?

A. Scope

B. Range

C. Limit

D. Domain

The correct term for the area where a variable is defined and can be accessed is "Scope." In programming, scope refers to the visibility of variables and functions in different parts of the code. When a variable is declared, its scope determines where in the program that variable can be accessed or modified. For instance, a variable defined inside a function has a local scope, meaning it can only be accessed within that function. In contrast, variables defined at the module level usually have a global scope, allowing them to be accessible from any function within that module. Understanding scope is crucial for managing variable visibility and avoiding conflicts, such as naming collisions. The other terms listed do not pertain to the concept of variable accessibility in programming. "Range" is often used in mathematics and programming to describe the span of values, "Limit" refers to constraints or bounds, and "Domain" can indicate the set of possible input values for functions but does not specifically relate to variable accessibility. Thus, scope is the precise term that captures the essence of where variables can be accessed in code.

9. What is the name given to source programs that are written in scripting languages?

A. Scripts

B. Modules

C. Functions

D. Libraries

The term used to describe source programs that are written in scripting languages is "scripts." Scripts are typically files that contain a sequence of instructions written in a scripting language, which is interpreted and executed at runtime. Scripting languages are often used for automating tasks or running short programs, and the code in these files is usually meant to be executed in a specific environment, such as a web server or within an application. Modules refer to files that contain Python code that can be included in other Python programs. Functions are reusable pieces of code that perform a specific task and are defined within a module or script. Libraries are collections of modules and functions that provide additional functionalities. Understanding the distinction between scripts, modules, functions, and libraries is important for organizing code and making it reusable and maintainable. In the context of scripting, "scripts" is the appropriate term to describe the source programs devised in scripting languages.

10. What are fixed values written directly in the code that represent constant data called?

A. Variable

B. Reference

C. Literal

D. Constant

Fixed values that are embedded directly within code to represent constant data are termed literals. A literal is a notation for representing a fixed value in a programming language. For example, when you write a number like `42` or a string like `"hello"` directly in your code, those are considered literals because they represent specific constant values that do not change during program execution. The concept of literals is fundamental in programming, as they allow developers to define values for variables, perform calculations, or represent strings. Understanding literals helps in comprehending how data is handled in programming and how different types of data (like numbers, strings, booleans, etc.) can be directly utilized within the code. Other terms, while related, do not accurately describe fixed values as literals do. Variables, for instance, hold data that can change, and constants are typically defined using a variable-like syntax that indicates their values should not change throughout the program's execution, but they are not referred to as literals themselves. In contrast, a reference usually pertains to a location in memory rather than a direct fixed value.