Texas A&M University (TAMU) ENGR102 Engineering Lab I - Computation Practice Exam (Sample)

Study Guide

BY EXAMZIFY

Everything you need from our exam experts!

Notice: Examzify makes every reasonable effort to obtain from reliable sources accurate, complete, and timely information about this product.

Questions

1. What is a common use for the `for` loop in Python?

   A. To handle exceptions in code

   B. To iterate over a sequence

   C. To define functions

   D. To create conditionals

2. In MATLAB, how do you evaluate the square root of a number?

   A. By using the sqrt() function

   B. By using the squareRoot() function

   C. By using the power() function

   D. By using the root() function

3. What does the term 'object' signify in object-oriented programming?

   A. A distinct data type

   B. A unit containing data and behavior

   C. A single-use variable

   D. A collection of functions

4. Functions in programming are designed to enhance which of the following aspects?

   A. Monolithic structures

   B. Code organization and modularity

   C. Data integrity only

   D. Real-time processing

5. What is the output of the function int(4.9)?

   A. 4.9

   B. 5

   C. 4

   D. 4.0

6. Which of the following is NOT a type of variable?

A. Booleans

B. Strings

C. Datasets

D. Integers

7. What is 'pseudocode'?

A. A coding language used for developing software

B. A method for debugging code

C. A simplified way of expressing algorithms

D. An advanced programming tool

8. How do you structure a switch-case statement in MATLAB?

A. By using: switch variable, case condition, statements, otherwise, default statements, end

B. By using: switch variable, condition case, then statements, else, end statements

C. By using: case condition, switch variable, end statements, else default

D. By using: condition, switch variable, case statements, end

9. Which function would you use to load variables from a MAT-file?

A. import()

B. load()

C. fetch()

D. read()

10. In Python, what does the 'len()' function do?

A. Returns the last element of a list

B. Returns the length of an object like a list, string, or tuple

C. Returns the largest element in a list

D. Returns a formatted string representation of an object

Answers

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

Explanations

1. What is a common use for the `for` loop in Python?

A. To handle exceptions in code

B. To iterate over a sequence

C. To define functions

D. To create conditionals

The `for` loop in Python is commonly used as a mechanism to iterate over a sequence, which can include lists, tuples, strings, and other iterable objects. This allows you to execute a block of code multiple times, once for each item in the sequence, which is particularly useful for processing or manipulating data within the sequence.   For example, if you wanted to print each element of a list, you would use a `for` loop to traverse the list and perform an action for each item. This capability makes the `for` loop an essential tool in programming, as it facilitates automation and efficiency in handling repetitive tasks. In contrast, handling exceptions in code pertains to error management, which typically involves `try` and `except` blocks. Defining functions involves the use of the `def` keyword and is unrelated to the iterative nature of the `for` loop. Creating conditionals involves `if`, `elif`, and `else` statements, which are used to execute code based on specific logical conditions rather than iteration.

2. In MATLAB, how do you evaluate the square root of a number?

A. By using the sqrt() function

B. By using the squareRoot() function

C. By using the power() function

D. By using the root() function

In MATLAB, the square root of a number is evaluated using the specific function called `sqrt()`. This built-in function is designed to compute the square root of a given value efficiently. When you provide a number as an input to the `sqrt()` function, it returns the principal square root of that number. For example, using `sqrt(25)` would yield `5`, since 5 is the square root of 25. The other options do not represent valid ways to compute the square root in MATLAB. The `squareRoot()` function does not exist, and attempting to use it would result in an error. Similarly, while the `power()` function can be used to raise a number to a specified exponent, it is not the preferred method for directly obtaining the square root. The `power()` function requires using an exponent of 0.5 to achieve the same result as `sqrt()`, which is less straightforward. Lastly, the `root()` function is also not defined in MATLAB for this purpose, and using it would likewise lead to an error. Therefore, the use of `sqrt()` is the clear and correct choice.

3. What does the term 'object' signify in object-oriented programming?

A. A distinct data type

B. A unit containing data and behavior

C. A single-use variable

D. A collection of functions

In object-oriented programming, the term 'object' signifies a unit that encapsulates both data and behavior. This encapsulation is fundamental to object-oriented design, as it allows developers to create self-contained structures that represent real-world entities or concepts. Each object has attributes (data) that describe its properties and methods (behavior) that define what actions can be performed on the object or what it can perform.   For example, if you consider an object representing a car, the data might include properties such as color, make, and model, while the behavior could include methods such as start, stop, and accelerate. This combination of data and behavior promotes modularity and reusability in programming, making it easier to manage complex systems by breaking them down into simple, interacting components.   Understanding this concept is crucial for applying object-oriented principles effectively in software development.

4. Functions in programming are designed to enhance which of the following aspects?

A. Monolithic structures

B. Code organization and modularity

C. Data integrity only

D. Real-time processing

Functions in programming are essential for advancing code organization and modularity. When a program is structured with functions, it allows developers to break down complex tasks into smaller, more manageable, and reusable components. This modularization of code helps in several ways: 1. **Readability**: Programs become easier to read and understand. Each function can focus on a specific task, and descriptive names can clarify their purpose. 2. **Reusability**: Functions can be reused in different parts of a program or even in different programs. This reduces redundancy and saves time when developers need to implement similar functionality. 3. **Maintainability**: When code is organized into functions, it is easier to update or debug. Changes can be made in one location (the function) without affecting the entire program, thus minimizing the chances of introducing bugs. 4. **Collaboration**: In team environments, functions allow different team members to work on different parts of a project simultaneously, enhancing productivity and coordination. In contrast, options related to monolithic structures, data integrity only, and real-time processing do not capture the comprehensive advantages that functions provide in terms of organization and modularity, making them less relevant in the context of enhancing programming practices.

5. What is the output of the function int(4.9)?

A. 4.9

B. 5

C. 4

D. 4.0

The output of the function int(4.9) is 4 because the int() function in Python is designed to convert a floating-point number to an integer by truncating the decimal portion. This means that when int(4.9) is executed, the decimal part (.9) is ignored, and only the whole number part is retained, resulting in the integer value 4. This behavior is consistent across many programming languages, where type conversion functions that convert a float to an integer typically discard the decimal portion, rather than rounding up or down. As a result, the value 4.9 becomes 4 when using the int() function.

6. Which of the following is NOT a type of variable?

A. Booleans

B. Strings

C. Datasets

D. Integers

Datasets are typically not classified as a variable type in the same way that Booleans, Strings, and Integers are. In programming and data analysis, a variable is a storage location identified by a name that can hold different values over time. Booleans are variables that can hold one of two values: true or false. They are frequently used to represent conditions in logic operations. Strings represent sequences of characters and are commonly used for text data. Integers are whole numbers without a decimal component, used for counting, indexing, or any mathematical operations where fractional values are not required. Datasets, however, often refer to structured collections of data, which could include various types of variables, but the dataset itself is not a type of variable. It can contain multiple variables, each potentially of different types (including Booleans, Strings, and Integers), but is considered more of a data structure or container than a variable type. This distinction clarifies why datasets do not fall under the same classification as the other options listed.

7. What is 'pseudocode'?

A. A coding language used for developing software

B. A method for debugging code

C. A simplified way of expressing algorithms

D. An advanced programming tool

Pseudocode is fundamentally a simplified way of expressing algorithms, making it easier for individuals to conceptualize the logic without the complexities of a specific programming language's syntax. This abstraction helps in planning out the structure and flow of the algorithm before converting it into actual code.   By using pseudocode, developers can focus on the logic and steps of the algorithm without being bogged down by coding rules, allowing for clearer communication of ideas and algorithms among team members. It serves as a bridge between the problem-solving phase and the actual implementation in a programming language.   The other options relate to aspects of software development and programming but do not accurately define pseudocode. A coding language is more formal and requires strict syntax, while debugging is the process of identifying and fixing errors in existing code. An advanced programming tool would imply software designed for specific tasks, which does not align with the conceptual and communicative nature of pseudocode.

8. How do you structure a switch-case statement in MATLAB?

A. By using: switch variable, case condition, statements, otherwise, default statements, end

B. By using: switch variable, condition case, then statements, else, end statements

C. By using: case condition, switch variable, end statements, else default

D. By using: condition, switch variable, case statements, end

The structure of a switch-case statement in MATLAB is designed to provide a clear and efficient way to handle multiple conditions based on the value of a given variable. The correct structure begins with the keyword "switch," followed by the variable that you want to evaluate. Each potential value of this variable is handled by the "case" statement, where you define the specific conditions and the associated actions or statements that should be executed for each case. Additionally, the structure includes an "otherwise" section, which acts as a default catch-all for any cases that do not match the specified conditions. This is useful for handling unexpected values gracefully. Finally, the entire construction is concluded with the "end" statement to denote the end of the switch-case block. In summary, the correct approach follows a clear and organized format: you start with "switch variable", followed by multiple "case condition" clauses for specified values, an optional "otherwise" clause for catching anything not explicitly handled, and end the statement with "end." This structure allows for readable and maintainable code, which is essential for effective programming practices in MATLAB.

9. Which function would you use to load variables from a MAT-file?

A. import()

B. load()

C. fetch()

D. read()

The function used to load variables from a MAT-file is load(). This function allows you to import data stored in MATLAB’s binary format, which is commonly used to save variables and arrays. When invoked, load() reads the contents of a specified MAT-file and assigns the variables stored in it to the MATLAB workspace, making it easy to access the data for further analysis or manipulation. Understanding the significance of this function is essential for effective data management in MATLAB, especially for engineering applications where data analysis and processing are critical. Other functions, while useful in different contexts, do not serve the same purpose. For instance, import() might be associated with importing data from different file formats, fetch() is generally related to obtaining data from databases or web services, and read() could imply various reading operations that do not specifically target MAT-files. Thus, recognizing the specialized role of load() in handling MATLAB's native data format is key to proper data handling in your engineering lab work.

10. In Python, what does the 'len()' function do?

A. Returns the last element of a list

B. Returns the length of an object like a list, string, or tuple

C. Returns the largest element in a list

D. Returns a formatted string representation of an object

The 'len()' function in Python is designed to provide the length of various data structures, such as lists, strings, and tuples. When applied to a list, it counts the number of elements within that list. For strings, it returns the number of characters, including spaces, and for tuples, it similarly counts the number of elements contained within the tuple. This function is fundamental in Python programming as it allows developers to quickly ascertain the size of an object, which is often necessary in loops, conditionals, or when manipulating data. For example, if you have a list of students' names, using 'len()' will help you determine how many names are present to process accordingly. Other options present different functionalities that do not align with what 'len()' provides. For instance, while some functions can return specific elements from a list or string, those tasks are outside the scope of what 'len()' is meant to do. Similarly, 'len()' does not perform value comparisons nor can it format representations of objects, making it clear that the best description of 'len()' in this context is its capability to return the length of an object like a list, string, or tuple.