

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

Study Guide



Everything you need from our exam experts!

Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

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

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	5
Answers	8
Explanations	10
Next Steps	16

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 - 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

SAMPLE

- 1. 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**

- 2. Which function would you use to concatenate arrays in MATLAB?**
 - A. concat()**
 - B. join()**
 - C. cat()**
 - D. arrayCombine()**

- 3. What is the output of the 'print' function in Python?**
 - A. It displays text in a new window**
 - B. It sends output to the console**
 - C. It logs data into a file automatically**
 - D. It produces a sound alert**

- 4. What is a 'for loop' used for in programming?**
 - A. To define a function**
 - B. To execute a block of code repeatedly**
 - C. To create a new list**
 - D. To handle errors in the program**

- 5. What does 'inheritance' signify in object-oriented programming?**
 - A. A feature that restricts data access**
 - B. A method of organizing data in tables**
 - C. A mechanism where classes share properties and methods**
 - D. A process of combining different classes**

6. How would you check if a number is odd in Python?

- A. Using division: if number / 2 != 0**
- B. Using addition: if number + 1 % 2 == 0**
- C. Using modulus: if number % 2 != 0**
- D. Using multiplication: if number * 1 % 2 != 0**

7. Which keyword is used to define a function in Python?

- A. function**
- B. def**
- C. method**
- D. declare**

8. What output will the following code produce if temperature is set to 100? if is_hot: print("It's hot")

- A. It's not hot**
- B. It's hot**
- C. No output**
- D. Error**

9. Which exception is raised when trying to perform an operation that is invalid for a specific data type?

- A. OSError**
- B. TypeError**
- C. ValueError**
- D. IndexError**

10. How can elements in a list be accessed in Python?

- A. Using the keys of the dictionary**
- B. Through string manipulation**
- C. Using indexing**
- D. By looping through the list**

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. 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.

2. Which function would you use to concatenate arrays in MATLAB?

- A. concat()**
- B. join()**
- C. cat()**
- D. arrayCombine()**

In MATLAB, the function used to concatenate arrays is 'cat()'. This function allows you to join multiple arrays along a specified dimension. The first argument of 'cat()' is the dimension along which the concatenation will occur, followed by the arrays you wish to concatenate. For instance, if you have two arrays and you want to concatenate them vertically, you would use 'cat(1, array1, array2)'. Here, the '1' indicates that the arrays are being joined along the first dimension (rows). Conversely, if you want to concatenate them horizontally, you would use 'cat(2, array1, array2)', where '2' refers to the second dimension (columns). The other functions listed do not serve this purpose. 'concat()' is not a built-in MATLAB function, 'join()' is used for joining strings rather than arrays, and 'arrayCombine()' is also not a standard MATLAB function for concatenation. Thus, 'cat()' is definitively the correct choice for concatenating arrays in MATLAB, highlighting the specific functionality and versatility it offers for array manipulation.

3. What is the output of the 'print' function in Python?

- A. It displays text in a new window
- B. It sends output to the console**
- C. It logs data into a file automatically
- D. It produces a sound alert

The 'print' function in Python is primarily designed to send output to the console, which is the command line or terminal from which the Python program is executed. This function takes one or more arguments, converts them to a string if necessary, and then writes that string to the standard output, which is typically displayed on the screen. This functionality is essential for displaying messages, debugging, and providing feedback to the user regarding the execution of a program. The clarity of output in the console makes it a versatile tool for developers to view results directly without needing additional interfaces or commands. In contrast, other options describe functions or behaviors that do not relate to the primary role of the 'print' function. For instance, displaying text in a new window typically requires a graphical user interface context, logging data into a file would necessitate file handling operations, and producing sound alerts would involve audio libraries or separate functions entirely, which are unrelated to the purpose of 'print'. Thus, the correct understanding of the 'print' function is that it effectively communicates outputs directly to the console for immediate visibility and interaction.

4. What is a 'for loop' used for in programming?

- A. To define a function
- B. To execute a block of code repeatedly**
- C. To create a new list
- D. To handle errors in the program

A 'for loop' is a fundamental control flow structure in programming that is specifically designed to execute a block of code repeatedly for a specified number of iterations or over a collection of items. It allows developers to efficiently run the same set of instructions multiple times, which is particularly useful when working with arrays, lists, or any collection where a repetitive operation is required. For example, if you have a list of numbers and want to calculate the total sum, you could use a for loop to iterate through each number in the list and add it to a cumulative sum variable. The loop continues until all elements have been processed, showcasing the effectiveness of the for loop in automating repetitive tasks. This capacity to manage iterations and streamline code makes it an essential tool in programming. While defining functions, creating new lists, and error handling are important programming concepts, they do not describe the function of a for loop. Thus, the role of a for loop is aptly captured by its ability to execute a block of code repeatedly.

5. What does 'inheritance' signify in object-oriented programming?

- A. A feature that restricts data access**
- B. A method of organizing data in tables**
- C. A mechanism where classes share properties and methods**
- D. A process of combining different classes**

Inheritance in object-oriented programming refers to a mechanism where classes can share properties and methods. This allows a new class, known as a subclass or derived class, to inherit attributes and behaviors (methods) from an existing class, referred to as a superclass or base class. This concept promotes reusability and the establishment of a hierarchical relationship between classes. For instance, if you have a base class called "Animal" with properties like "species" and methods like "makeSound," a subclass "Dog" can inherit these features, allowing it to utilize or override them while also introducing its own unique properties and methods. By using inheritance, developers can create a structured and organized approach to their codebase, minimizing redundancy and facilitating maintenance, as common functionality needs to be written only once in the base class. The other choices do not accurately capture the essence of inheritance in object-oriented programming. The first option relates more to access control, the second refers to data organization typically associated with databases, and the fourth implies a merging of distinct classes, which does not represent the concept of inheritance.

6. How would you check if a number is odd in Python?

- A. Using division: if `number / 2 != 0`**
- B. Using addition: if `number + 1 % 2 == 0`**
- C. Using modulus: if `number % 2 != 0`**
- D. Using multiplication: if `number * 1 % 2 != 0`**

To determine if a number is odd in Python, the most effective method is to utilize the modulus operator. When you perform `number % 2`, you get the remainder of the division of the number by 2. If this remainder is not equal to zero, it indicates that the number cannot be evenly divided by 2, thus confirming that it is odd. For example, if the number is 5, the calculation would be `5 % 2`, which equals 1. Since 1 is not equal to 0, it confirms that 5 is indeed an odd number. This method is both clear and efficient, making it the standard approach for checking whether a number is odd in programming. Other methods listed might lead to confusion or incorrect results. For instance, using division can create a false impression since dividing any number could yield a non-zero result even for even numbers, while the addition and multiplication methods are convoluted and do not directly assess the characteristics of odd and even numbers as clearly as the modulus method does.

7. Which keyword is used to define a function in Python?

- A. function
- B. def**
- C. method
- D. declare

In Python, the keyword used to define a function is "def." This keyword initiates the function definition and indicates to the Python interpreter that a function is being declared. Following "def," the user specifies the function name and any parameters that the function may take in parentheses. For example, the syntax for defining a simple function in Python is: ``python def my_function(param1, param2): # function body return param1 + param2 `` In this case, "def" signals the start of the function definition, followed by the function's name and its input parameters. This structure is essential for creating reusable blocks of code that can be executed when the function is called later in the program. Understanding the use of "def" is crucial for writing and organizing code effectively in Python.

8. What output will the following code produce if temperature is set to 100? if is_hot: print("It's hot")

- A. It's not hot
- B. It's hot**
- C. No output
- D. Error

The output of the code depends on the value of the variable `is_hot`. If the code checks for the condition `if is_hot:` and `is_hot` is set to True (which is usually determined by the logic preceding this code), then the message "It's hot" will be printed when the temperature variable is set to 100. In situations like this, the key point is the evaluation of the boolean condition. If `is_hot` is correctly defined and evaluated as True in association with a temperature of 100 (indicating a hot condition), the print statement within the if block is executed, resulting in the output "It's hot." Thus, the code effectively reflects a specified condition based on the value of `is_hot`. In the absence of proper context on how `is_hot` is defined in relation to the temperature, one can assume the logic aligned with typical conditions where temperature values determine thresholds for "hot" conditions.

9. Which exception is raised when trying to perform an operation that is invalid for a specific data type?

- A. OSError**
- B. TypeError**
- C. ValueError**
- D. IndexError**

The **TypeError** is raised when an operation is applied to an object of an inappropriate type. This typically occurs in scenarios where a function or operation expects a specific data type, but receives a different type instead. For instance, if you attempt to add a string and an integer, Python will raise a **TypeError** because the addition operation cannot be performed between these two different data types. Understanding this exception is crucial for debugging and writing robust code, as it helps to ensure that operations are only performed on compatible data types. In contrast, other exceptions like **OSError** relate to system-related issues, **ValueError** occurs when a function receives a parameter of the right type but an inappropriate value, and **IndexError** is raised when trying to access an index in a sequence that is out of range. Each of these exceptions has its own specific context and use case, which is different from the scenario where a **TypeError** would be applicable.

10. How can elements in a list be accessed in Python?

- A. Using the keys of the dictionary**
- B. Through string manipulation**
- C. Using indexing**
- D. By looping through the list**

Accessing elements in a list in Python is primarily achieved through indexing. Each element in a list is assigned a specific index, starting from zero for the first element. This allows easy retrieval of elements based on their position. For example, to access the first element of a list called `'my_list'`, you would use `'my_list[0]'`. This indexing method is efficient and straightforward, making it the standard practice for accessing individual items within a list. While looping through the list is also a valid method to access elements collectively, it is not the primary way to access a specific element when you want to retrieve just one. String manipulation and dictionary keys do not apply in this context, as they pertain to different data structures. Thus, indexing is the most accurate and direct method for accessing elements in a Python list.

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://tamu-engr102.examzify.com>

We wish you the very best on your exam journey. You've got this!

SAMPLE