Western Governors University (WGU) C859 Introduction to Programming in Python Practice Test (Sample)

Study Guide

BY EXAMZIFY

Everything you need from our exam experts!

Table of Contents

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

1. **Which keyword is used to define a lambda function in Python?**

   A. def

   B. lambda

   C. function

   D. method

2. **What is the first argument of the split method in Python when splitting a string?**

   A. A character to split on

   B. The string to be split

   C. A list of delimiters

   D. A number indicating max splits

3. **What is the outcome of using from collections import defaultdict?**

   A. Imports the entire collections module

   B. Imports specifically the defaultdict function from collections

   C. Exports the defaultdict function to another module

   D. Deletes the defaultdict function from the collections module

4. **What is the result of accessing an index that does not exist in a list?**

   A. It returns None

   B. It raises an IndexError

   C. It returns the last element

   D. It creates a new list

5. **What data type is created by the expression myPhone = (877, 435, 7948)?**

   A. A tuple

   B. A list

   C. A dictionary

   D. A set

6. When you use write() on an open file in 'w' mode, what happens to existing data within the file?

A. It appends new data at the end

B. It clears the existing data before writing

C. It leaves the existing data intact

D. It locks the file for secure writing

7. Which method would you use to remove characters from both ends of a string?

A. trim()

B. strip()

C. clean()

D. remove()

8. Which of the following is NOT a built-in data type in Python?

A. List

B. Set

C. Buffer

D. Tuple

9. Which function is used to convert a string to uppercase in Python?

A. upper()

B. to_upper()

C. uppercase()

D. capitalize()

10. In which scenario would you use the add() method?

A. To check for the existence of a key

B. To remove an item from a list

C. To append an item to a set

D. To insert an item into a dictionary

Answers

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

Explanations

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

A. def

B. lambda

C. function

D. method

In Python, the keyword used to define a lambda function is 'lambda'. A lambda function is a small, anonymous function that is defined using the lambda keyword rather than the traditional def keyword used for regular functions.   Lambda functions can take any number of arguments but can only have one expression. The expression is evaluated and returned. This allows for quick and concise function definitions that are especially useful for short, throwaway functions that don't require a formal definition. For example, the syntax for a lambda function is: ```python lambda arguments: expression ``` This makes lambda a powerful tool when you need simple functionality without the overhead of defining a full function using def. The simplicity and expressiveness of lambda functions allow them to be integrated seamlessly with functions like map(), filter(), and sorted(), among others.   Hence, the correct choice is 'lambda', as it directly relates to the creation of such compact functions in Python.

2. What is the first argument of the split method in Python when splitting a string?

A. A character to split on

B. The string to be split

C. A list of delimiters

D. A number indicating max splits

The first argument of the split method in Python is indeed a character or string that specifies the delimiter on which the original string should be split. When you use the split method, you can provide this delimiter to indicate where the splits should occur within the string. For example, if you have a string like "apple,banana,cherry" and you call the split method with a comma as the argument (using string.split(',')), it will separate the string into a list of substrings: ["apple", "banana", "cherry"]. The delimiter you provide tells the split method where to break the original string into different parts based on that specific character or string. While the split method can also take additional arguments, such as the maximum number of splits or a different behavior if the argument is omitted (which defaults to whitespace), the primary function relies heavily on the first argument being the character or string that dictates where the splits happen.

3. What is the outcome of using from collections import defaultdict?

A. Imports the entire collections module

B. Imports specifically the defaultdict function from collections

C. Exports the defaultdict function to another module

D. Deletes the defaultdict function from the collections module

Using "from collections import defaultdict" specifically imports the defaultdict class from the collections module into the current namespace. This allows you to use defaultdict directly without needing to prefix it with "collections." The defaultdict class is a subclass of the built-in dict class, and it overrides one method to provide a default value for a nonexistent key. This is particularly useful for creating dictionaries that need to handle missing keys gracefully. When you perform the import in this way, you are not importing the entire collections module, which would include all of its components. Nor are you exporting or deleting any functions; the import statement merely makes defaultdict available in your code for immediate use. This focused import is efficient and helps keep the namespace clean.

4. What is the result of accessing an index that does not exist in a list?

A. It returns None

B. It raises an IndexError

C. It returns the last element

D. It creates a new list

When trying to access an index that does not exist in a list, Python raises an IndexError. This type of error indicates that the index specified is outside the range of valid indexes for that list. For example, if you try to access the element at index 5 in a list that has only three elements (indexes 0, 1, and 2), Python cannot find the index and will raise an exception. This behavior is a fundamental part of Python's error handling, designed to prevent unexpected behavior in programs. It ensures that programmers are made aware of their access attempts to out-of-bounds indexes, prompting them to either check the length of the list or handle the situation gracefully with error handling techniques. Other options don't represent the actual behavior of Python lists when an invalid index is accessed.

5. What data type is created by the expression myPhone = (877, 435, 7948)?

A. A tuple

B. A list

C. A dictionary

D. A set

The expression `myPhone = (877, 435, 7948)` creates a tuple because it is defined by the use of parentheses and contains a sequence of values. Tuples in Python are immutable, meaning that once they are created, their contents cannot be modified (i.e., you can't add, remove, or change items). This characteristic makes tuples suitable for storing fixed collections of items. In this case, `myPhone` is intended to store a phone number, which typically consists of distinct components that shouldn't change, such as area code, central office code, and line number. Therefore, using a tuple is appropriate for this scenario. Other data types mentioned do not match the characteristics of the tuple in this example. Lists, for instance, are mutable and defined by square brackets. Dictionaries use key-value pairs and are identified by curly braces containing pairs of keys and values. Sets are also defined by curly braces but do not allow duplicate values and are unordered collections. Hence, given the specific syntax and properties of the expression, it is clear that a tuple is the correct data type being utilized.

6. When you use write() on an open file in 'w' mode, what happens to existing data within the file?

A. It appends new data at the end

B. It clears the existing data before writing

C. It leaves the existing data intact

D. It locks the file for secure writing

When a file is opened in 'w' (write) mode in Python, the behavior is designed to overwrite the contents of the file. Therefore, when the write() method is called on a file opened in this mode, it first clears any existing data within the file before writing the new data. This means that if the file had any content already, it will be entirely removed, and the new data will be written from the beginning of the file, effectively replacing whatever was there before. This feature is particularly useful when you want to start afresh with a file without maintaining the previous content. In contrast, other modes such as 'a' (append) would add data to the end without modifying the current contents, which is not the case here. The 'r' (read) mode only supports reading from an existing file without altering it, reinforcing that 'w' mode is specifically intended for writing new data after clearing the old. Additionally, file locking is not a function of the write mode itself but rather a mechanism that must be implemented separately if needed.

7. Which method would you use to remove characters from both ends of a string?

A. trim()

B. strip()

C. clean()

D. remove()

The method used to remove characters from both ends of a string in Python is the `strip()` method. This function is designed specifically to eliminate whitespace characters (spaces, tabs, newlines) from the beginning and end of a string, but it can also remove specified characters if you provide them as arguments. For example, if you have a string with leading and trailing spaces and you want to clean it up, using `strip()` will effectively give you the modified string without those spaces. Additionally, you can use `strip()` to remove any characters you specify. For instance, if you want to remove specific punctuation or letters, you can pass those as a string to the method. Other options such as `trim()`, `clean()`, and `remove()` do not exist in Python as string methods for this specific purpose. So using the `strip()` method is the appropriate choice for achieving the goal of removing characters from both ends of a string.

8. Which of the following is NOT a built-in data type in Python?

A. List

B. Set

C. Buffer

D. Tuple

The choice of "Buffer" as the answer is correct because it is not considered a built-in data type in Python in the same way that the others are. Python has several built-in data types that are fundamental to programming in the language, including lists, sets, and tuples. A list is an ordered collection that can hold a variety of items, including other lists, and allows for modification of its contents. A set is a collection of unique elements, meaning there can be no duplicates, and it is unordered. A tuple is similar to a list but is immutable, meaning once it is created, its content cannot be changed. While Python does include a `bytes` type that represents binary data, the term "buffer" is not specifically recognized as a standard built-in data type in Python. Therefore, it stands out among the options as the one that does not belong in the same category as list, set, and tuple.

9. Which function is used to convert a string to uppercase in Python?

A. upper()

B. to_upper()

C. uppercase()

D. capitalize()

The function used to convert a string to uppercase in Python is upper(). This built-in method is called on a string object and returns a new string where all the lowercase letters in the original string have been converted to their uppercase counterparts. It's a straightforward and commonly used function when you want to ensure that the text representation is in uppercase form, which can be particularly useful for standardizing input, formatting for display, or making comparisons case-insensitive. For example, using the upper() function like this: ```python text = "hello" uppercase_text = text.upper() ``` will result in `uppercase_text` being "HELLO". This is the correct and most efficient way to achieve the conversion to uppercase in Python. The other options listed do not exist as standard string methods in Python, making them unsuitable for converting strings to uppercase.

10. In which scenario would you use the add() method?

A. To check for the existence of a key

B. To remove an item from a list

C. To append an item to a set

D. To insert an item into a dictionary

The add() method is specifically designed to append or add an element to a set in Python. A set is a collection type that is unordered, does not allow duplicates, and is mutable. The primary function of the add() method is to insert a new element into the set without creating any duplicates. When using add(), if the specified element already exists in the set, it will not be added again, ensuring that the set remains a collection of unique elements. This method is straightforward and efficient for managing membership in collections when duplication is not desired. Alternative scenarios mentioned involve operations that are handled by other methods. For instance, checking for the existence of a key in a dictionary would typically involve using the 'in' keyword or the get() method, and removing items from a list can be done using methods like remove() or pop(). Inserting items into a dictionary is often achieved using direct assignment rather than a method called add(). Thus, using add() in the context of sets aligns perfectly with its primary purpose.

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://wgu-c859.examzify.com

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