WESTERN GOVERNORS UNIVERSITY (WGU) C173 SCRIPTING AND PROGRAMMING PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://wgu-c173.examzify.com

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 15

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.

SAMPLE

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!

SAMPLE

Questions

SAMPLE

1. **What would be the outcome of trying to access an undefined variable in JavaScript?**

   A. Returns a null value

   B. Generates a syntax error

   C. Returns `undefined`

   D. Triggers an exception

2. **What is the purpose of a version control system?**

   A. To compile code into machine language

   B. To track changes in code and manage collaboration

   C. For enhancing graphical representations in code

   D. To secure code from unauthorized access

3. **How do you create a comment in HTML?**

   A. A comment in HTML is created using ""

   B. A comment in HTML is created using "// comment"

   C. A comment in HTML is created using "/* comment */"

   D. A comment in HTML is created using "# comment"

4. **Which UML diagram can illustrate the relationship between different objects in the system?**

   A. Component Diagram

   B. Class Diagram

   C. Deployment Diagram

   D. State Diagram

5. **Which of the following languages is considered object-oriented?**

   A. C

   B. Python

   C. Fortran

   D. BASIC

SAMPLE

6. How does recursion work in programming?

A. It uses loops to repeat commands

B. It divides a problem into smaller sub-problems

C. It requires a global variable to operate

D. It ends execution after one complete run

7. Which operator should be used to determine if a number is evenly divisible by 5?

A. +

B. -

C. %

D. /

8. What differentiates a variable from a constant in programming?

A. A variable can be changed, while a constant cannot

B. A constant can be changed, while a variable cannot

C. Variables are only numbers, constants are text

D. Constants take more memory than variables

9. How do you define a class in Python?

A. A class is defined using the class keyword followed by the class name and a colon.

B. A class is defined using the define keyword followed by the class name.

C. A class is defined using the function keyword followed by the class name.

D. A class is defined using the object keyword followed by the class name.

10. What is a key characteristic of a class in Python?

A. A class cannot inherit properties from another class

B. A class can contain attributes and methods

C. A class must define its properties as global

D. A class cannot have object instances

SAMPLE

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What would be the outcome of trying to access an undefined variable in JavaScript?

A. Returns a null value

B. Generates a syntax error

C. Returns `undefined`

D. Triggers an exception

When attempting to access an undefined variable in JavaScript, the outcome is that it returns `undefined`. In JavaScript, if you try to access a variable that has not been declared or initialized, the language does not throw an error; instead, it returns a special value called `undefined`. This indicates that the variable exists but has not been assigned a value. The behavior of JavaScript allows developers to check whether a variable has been assigned a value by simply inspecting its value. If the variable is indeed undeclared, accessing it will still yield `undefined`, which can be useful in certain contexts, such as checking for the existence of an optional variable. This characteristic is part of JavaScript's flexible and dynamic nature, which allows for variables to be declared and used differently compared to other programming languages that may enforce stricter variable declarations.

2. What is the purpose of a version control system?

A. To compile code into machine language

B. To track changes in code and manage collaboration

C. For enhancing graphical representations in code

D. To secure code from unauthorized access

A version control system is fundamentally designed to track changes in code and manage collaboration among multiple developers. This system allows programmers to keep a historical record of changes made to the codebase, enabling them to revert to previous versions if needed and to understand how the code has evolved over time. It facilitates teamwork by allowing multiple developers to work on the same project simultaneously without overwriting each other's changes. Through features like branching and merging, developers can experiment with new ideas or features in isolation and later integrate them back into the main project when they are ready. While other options address various aspects related to programming and code management, they do not encapsulate the primary function of a version control system, which is to track and manage changes in code effectively. Compiling code relates to transforming source code into executable programs, graphical representations pertain to how code is visually structured and presented, and securing code against unauthorized access is more about application security than version control itself.

SAMPLE

3. How do you create a comment in HTML?

A. A comment in HTML is created using ""

B. A comment in HTML is created using "// comment"

C. A comment in HTML is created using "/* comment */"

D. A comment in HTML is created using "# comment"

In HTML, comments are created using the syntax `<!-- comment -->`. This allows you to include comments in your HTML code that will not be displayed on the webpage when viewed in a browser. Instead of putting the correct syntax for HTML comments, the option referring to using `""` does not actually represent a valid method for creating comments in HTML, and is, therefore, misleading. Understanding how comments function in HTML is vital because they serve as notes within the code. This can help developers explain parts of the code to themselves or to others without affecting the rendered output of the web page. Comments are especially useful for collaboration or when returning to code after some time, as they can clarify the intent and functionality of specific sections. Using other programming syntax, like `//`, `/* */`, or `#`, is relevant in other programming languages (for example, JavaScript, C, or Python), but they do not apply to HTML and would not achieve the purpose of commenting effectively in that environment. By mastering the correct syntax for comments in HTML, developers can keep their code clean and understandable.

4. Which UML diagram can illustrate the relationship between different objects in the system?

A. Component Diagram

B. Class Diagram

C. Deployment Diagram

D. State Diagram

The Class Diagram is the appropriate UML diagram for illustrating the relationships between different objects in a system. It provides a clear visual representation of classes, which are blueprints for objects, and their associations, such as inheritance, aggregation, and composition. This diagram allows developers to understand how objects interact with each other and the properties that define them. Class Diagrams are fundamental in object-oriented design as they help to outline the structure of a system by showing how classes relate to one another. By displaying attributes and methods alongside the relationships between classes, it gives a comprehensive view of the system's design and how various components work together. Other diagrams, while useful for different purposes, do not focus on object relationships in the same way. For instance, Component Diagrams emphasize the organization and dependencies among software components, Deployment Diagrams show the physical deployment of artifacts on nodes, and State Diagrams model the states of an object as it changes over time based on events. Although all these diagrams are valuable in their respective contexts, they do not specifically capture the direct relationships and interactions between objects like Class Diagrams do.

SAMPLE

5. Which of the following languages is considered object-oriented?

A. C

B. Python

C. Fortran

D. BASIC

Python is considered an object-oriented programming language because it supports the principles and paradigms associated with object-oriented design. This includes encapsulation, inheritance, and polymorphism, which are fundamental concepts in object-oriented programming (OOP). In Python, everything is treated as an object, including functions and data types, which allows developers to create classes and instantiate objects easily. The syntax and structure of Python facilitate the implementation of OOP principles, making it a powerful and flexible language for developers who wish to build applications using an object-oriented approach. Other languages listed, such as C, Fortran, and BASIC, do not fully embrace object-oriented paradigms in the same way Python does. While some of these languages may support certain aspects of OOP through additional libraries or extensions, they were not designed with object-oriented principles as a core focus. This distinction is key in understanding why Python stands out as an object-oriented language in this context.

6. How does recursion work in programming?

A. It uses loops to repeat commands

B. It divides a problem into smaller sub-problems

C. It requires a global variable to operate

D. It ends execution after one complete run

Recursion is a fundamental programming technique where a function calls itself in order to solve a problem. The correct answer highlights that recursion divides a problem into smaller sub-problems, which is central to how recursive functions operate. When using recursion, the function processes a complex problem by breaking it down into simpler, more manageable components that resemble the original issue. This process continues until a base case is reached — a condition under which the function can provide a straightforward answer without further recursion. An example of this can be seen in computing factorials. For instance, the factorial of a number n can be defined in terms of the factorial of (n-1), creating a series of smaller calculations until reaching the base case of 1. When implemented correctly, recursion not only simplifies the structure of the code but can lead to more elegant solutions than iterative approaches in certain scenarios. Understanding this principle is crucial for designing algorithms that effectively utilize recursion, and it underscores its utility in many programming contexts.

7. Which operator should be used to determine if a number is evenly divisible by 5?

A. +

B. -

C. %

D. /

To determine if a number is evenly divisible by 5, the modulus operator (%) is used. This operator calculates the remainder of a division operation. When a number is divided by 5, if the remainder is 0, it confirms that the number is evenly divisible by 5. For example, if you take the number 15 and check if it is divisible by 5 using the operation `15 % 5`, the result is 0, which means 15 is evenly divisible by 5. On the other hand, for the number 14, performing the same operation `14 % 5` yields a remainder of 4, indicating that 14 is not evenly divisible by 5. This method is straightforward and efficient for checking divisibility, making the modulus operator the correct choice for this purpose.

SAMPLE

8. What differentiates a variable from a constant in programming?

A. A variable can be changed, while a constant cannot

B. A constant can be changed, while a variable cannot

C. Variables are only numbers, constants are text

D. Constants take more memory than variables

The distinction between a variable and a constant in programming is primarily centered on mutability—whether the value can be altered during program execution. A variable is a named storage location that can hold different values throughout its lifecycle, allowing for updates and modifications as the program runs. This is essential for operations where data needs to be dynamic, such as in loops or conditionals, allowing the programmer to manage state and respond to varying conditions. On the other hand, a constant is a named storage location that holds a value intended to remain unchanged once it is set. This is useful for defining fixed values that should not be altered, such as mathematical constants (like π), configuration values, or other immutable data that the integrity of the program relies upon. Having constants can enhance code readability and maintainability, as they serve as a clear indication of fixed values being used throughout the program. In summary, the defining feature that separates a variable from a constant is that a variable can be changed during execution, while a constant is defined to remain static, which is why the first choice accurately reflects this fundamental principle in programming.

9. How do you define a class in Python?

A. A class is defined using the class keyword followed by the class name and a colon.

B. A class is defined using the define keyword followed by the class name.

C. A class is defined using the function keyword followed by the class name.

D. A class is defined using the object keyword followed by the class name.

A class in Python is defined using the `class` keyword, which is followed by the name of the class and a colon. This syntax is fundamental in Python and is essential for creating user-defined data types. The name of the class typically follows the convention of using CamelCase, which helps differentiate classes from functions and variables. For example, when you define a class named `MyClass`, the definition would look like this: ```python class MyClass: pass ``` This structure signifies the start of a class definition, and anything indented below the class definition represents the body of the class, where attributes and methods can be defined. Understanding this syntax is crucial for effectively using object-oriented programming in Python, as classes are the building blocks for creating objects that encapsulate data and behaviors. The other options provided do not correctly represent the syntax needed to define a class in Python.

10. What is a key characteristic of a class in Python?

A. A class cannot inherit properties from another class

B. A class can contain attributes and methods

C. A class must define its properties as global

D. A class cannot have object instances

The key characteristic of a class in Python is that a class can contain attributes and methods. In object-oriented programming, a class serves as a blueprint for creating objects (instances). Attributes are the variables that belong to the class, which hold data or state related to the object. Methods are functions defined within the class that can perform operations on the attributes or provide functionality related to the object. This ability to encapsulate both data (attributes) and behavior (methods) is fundamental to the concept of classes in Python, enabling developers to create organized and reusable code. By allowing the definition of attributes and methods, classes facilitate the modeling of real-world entities and their interactions in a structured way.

SAMPLE

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-c173.examzify.com

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

SAMPLE