Western Governors University (WGU) ITSW 2113 D278 Scripting and Programming Foundations Practice Exam (Sample)

Study Guide

Everything you need from our exam experts!

Questions

1. What is the purpose of the testing phase in the SDLC?

A. To ensure that the program meets the defined goals

B. To design user interfaces

C. To gather requirements from stakeholders

D. To deploy the application

2. Which statement accurately defines a for loop?

A. It contains only a condition to evaluate

B. It consists of initialization, condition, and increment

C. It has no specified number of iterations

D. It cannot contain nested loops

3. Which scripting language is commonly used for web development?

A. Java

B. JavaScript

C. Python

D. C++

4. In Python, what keywords are used for exception handling?

A. try and except

B. catch and throw

C. begin and end

D. error and handle

5. How does the agile approach differ from the waterfall approach?

A. It performs all phases simultaneously

B. It involves large teams working together

C. It repeats small portions of each phase in sequence

D. It eliminates the testing phase entirely

6. What does the term "inheritance" refer to in programming?

A. A method that converts data types

B. A mechanism for one class to inherit properties and methods from another class

C. A process of creating new data structures

D. A method of error handling within the code

7. Which of the following is NOT a characteristic of interpreted languages?

A. Coding is done statement by statement

B. They generally have slower execution times

C. They require a compiler for code generation

D. The code is executed in real-time

8. What are built-in functions in programming?

A. Functions created by the user for specific tasks

B. Predefined functions provided by the language for common tasks

C. Functions that require external libraries to operate

D. Functions that only run in specific environments

9. What does a use case diagram primarily illustrate?

A. The structure of classes in a program

B. The interaction between users and software

C. The sequence of code execution

D. The types of variables in a program

10. In programming, what is a function typically expected to return?

A. Always a string

B. A value or nothing at all

C. Only an integer

D. A boolean value

Answers

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

Explanations

1. What is the purpose of the testing phase in the SDLC?

A. To ensure that the program meets the defined goals

B. To design user interfaces

C. To gather requirements from stakeholders

D. To deploy the application

The testing phase in the Software Development Life Cycle (SDLC) is crucial for verifying that the program meets the defined goals and specifications outlined during the earlier stages of the project. This phase involves systematically checking the software for defects, ensuring that it functions as intended in various scenarios, and validating that it satisfies both the functional and non-functional requirements. During testing, different types of tests, such as unit testing, integration testing, system testing, and acceptance testing, are conducted. Each of these tests focuses on different aspects of the application, helping to identify bugs or weaknesses before the software is released to users. The aim is to ensure that the final product not only works correctly but also meets the expectations set during the requirements gathering phase. In contrast, designing user interfaces focuses more on the layout and user experience rather than testing functionality. Gathering requirements is a preliminary step that occurs before the coding and testing phases. Deploying the application is part of the final stages in the SDLC, following testing, where the software is delivered to users. Thus, the testing phase is essential for confirming that the program aligns with its intended goals and is ready for deployment.

2. Which statement accurately defines a for loop?

A. It contains only a condition to evaluate

B. It consists of initialization, condition, and increment

C. It has no specified number of iterations

D. It cannot contain nested loops

A for loop is a control flow statement that is specifically designed to iterate a set number of times, which is why the choice stating that it consists of initialization, condition, and increment is accurate. In a typical for loop structure, it begins with initialization where a counter variable is defined. Then, a condition is evaluated to determine whether the loop should continue running. Lastly, there is an increment operation that modifies the counter variable after each iteration. This combination allows for precise control of how many times the loop will execute, making it ideal for scenarios where the number of iterations is known before entering the loop. The other options do not correctly represent the characteristics of a for loop. For instance, just having a condition without initialization and increment does not define a for loop's structure. A for loop also typically requires a predetermined number of iterations, which is not implied by a lack of a specified number of iterations. Lastly, while for loops can contain nested loops, the presence of nesting does not restrict their functionality; thus, saying they cannot contain nested loops is inaccurate.

3. Which scripting language is commonly used for web development?

A. Java

B. JavaScript

C. Python

D. C++

JavaScript is the scripting language most commonly used for web development due to its ability to run in web browsers and provide interactivity to websites. It allows developers to create dynamic content, control multimedia, animate images, and much more, enhancing the user experience on the web. JavaScript is specifically designed to work within the browser environment, making it essential for front-end development alongside HTML and CSS. It also plays a crucial role in modern web frameworks and libraries, such as React, Angular, and Vue.js, which simplify the process of building complex, interactive web applications. Its versatility extends to server-side development as well with environments like Node.js, but its primary association remains with client-side scripting in web browsers. In contrast, while Java and Python are powerful programming languages used in various applications, neither is specifically designed for web development in the same way that JavaScript is. C++, although a robust language used for systems programming and game development, is not typically utilized for web development tasks.

4. In Python, what keywords are used for exception handling?

A. try and except

B. catch and throw

C. begin and end

D. error and handle

In Python, the keywords used for exception handling are "try" and "except." The "try" block is used to wrap the code that might potentially raise an exception. If an exception occurs within the "try" block, the execution is transferred to the "except" block, which contains the code for handling the exception. This structure allows programmers to manage errors gracefully, providing a way to respond to unexpected situations without crashing the program. The use of "try" and "except" is fundamental in Python for managing runtime errors, making programs more robust and user-friendly. They enable developers to separate normal operations from error-handling logic, enhancing the readability and maintainability of the code.

5. How does the agile approach differ from the waterfall approach?

A. It performs all phases simultaneously

B. It involves large teams working together

C. It repeats small portions of each phase in sequence

D. It eliminates the testing phase entirely

The agile approach is characterized by its iterative and incremental nature, which involves repeating small portions of each phase in sequence. This means that instead of completing all aspects of the project at once, agile methods break the project into smaller segments or iterations. Each iteration involves planning, execution, and testing, allowing teams to gather feedback and make adjustments continuously. This flexibility makes it easier to adapt to changes and improves collaboration among team members. In contrast, the waterfall approach is a linear process that completes each project phase before moving on to the next one, and it typically does not revisit earlier phases once they are finished. This fundamental difference allows agile to respond to evolving requirements and stakeholder feedback more effectively compared to the more rigid structure of waterfall. Understanding this key distinction is vital for recognizing the strengths and applications of the agile methodology within project management and software development.

6. What does the term "inheritance" refer to in programming?

A. A method that converts data types

B. A mechanism for one class to inherit properties and methods from another class

C. A process of creating new data structures

D. A method of error handling within the code

The term "inheritance" in programming specifically refers to a mechanism in object-oriented programming (OOP) where one class (often called a derived or child class) can inherit properties and methods from another class (often referred to as a base or parent class). This allows the child class to reuse code efficiently without having to rewrite it, promoting code reusability and establishing a hierarchical relationship between classes. In this context, inheritance facilitates the creation of a new class based on an existing one, enhancing modularity and making it easier to manage and scale applications. Additionally, it enables polymorphism, allowing methods to be overridden in the child class for specific behavior, while still maintaining a connection to the original functionality defined in the parent class. This creates a more structured approach to coding and helps in organizing code logically. Other options do not adequately encompass the core concept of inheritance in the context of programming: - A method converting data types does relate to type casting and conversion but lacks any mention of class relationships. - Creating new data structures pertains more to data management and organization rather than to inheritance principles. - Error handling within the code deals with managing errors and exceptions rather than the structural relationships between classes. Thus, the focus of inheritance on sharing and extending code effectively

7. Which of the following is NOT a characteristic of interpreted languages?

A. Coding is done statement by statement

B. They generally have slower execution times

C. They require a compiler for code generation

D. The code is executed in real-time

Interpreted languages operate by reading and executing code line by line or statement by statement, which is characteristic of their execution model. This means that coding in these languages usually involves writing out instructions that are processed immediately. The nature of interpreted execution naturally results in slower performance compared to compiled languages, where the entire code is translated into machine language before execution. Another notable characteristic is that interpreted languages execute code at runtime, often referred to as executing in real-time. This means any errors can be spotted as they occur within the source code during execution, allowing developers to receive immediate feedback on their code. The assertion that interpreted languages require a compiler for code generation does not align with their fundamental behavior. Compilers are used primarily in compiled languages, which convert the entire source code into machine code before execution, rather than interpreting it at runtime. Thus, the presence of a compiler is not a characteristic of interpreted languages, making this choice the correct response to the question posed.

8. What are built-in functions in programming?

A. Functions created by the user for specific tasks

B. Predefined functions provided by the language for common tasks

C. Functions that require external libraries to operate

D. Functions that only run in specific environments

Built-in functions in programming are predefined functions provided by the language itself to facilitate common tasks, such as performing mathematical calculations, manipulating strings, or working with data structures. These functions are readily available for use without the need for the programmer to define them from scratch. This characteristic saves time and effort, enabling developers to write more efficient and readable code by leveraging functionalities that have already been optimized and tested within the language. For instance, languages like Python offer built-in functions such as `len()`, which returns the length of an object, or `print()`, which outputs data to the console. These functions illustrate how built-in functions act as foundational components of programming languages, equipping developers with powerful tools that are easy to use.

9. What does a use case diagram primarily illustrate?

A. The structure of classes in a program

B. The interaction between users and software

C. The sequence of code execution

D. The types of variables in a program

A use case diagram primarily illustrates the interaction between users and software. It serves as a visual representation of how users, often referred to as "actors," engage with a system to achieve specific goals or tasks. This diagram highlights the functions of the system from a user's perspective and helps to identify the requirements for system functionality based on user needs. In a use case diagram, you'll typically find use cases that represent various functionalities or services the software provides, along with the actors that interact with those functionalities. This clear representation allows stakeholders, such as developers and clients, to grasp the essential interactions and workflows within the software, aiding in both understanding and communication. The primary focus on user interactions distinguishes use case diagrams from other design and modeling tools, which may address different aspects of system architecture or code execution. By emphasizing user needs and system responses, use case diagrams play a crucial role in ensuring that the developed software aligns with user expectations and requirements.

10. In programming, what is a function typically expected to return?

A. Always a string

B. A value or nothing at all

C. Only an integer

D. A boolean value

In programming, a function is typically expected to return a value or nothing at all, making this the correct choice. Functions are designed to perform specific tasks and, upon completion, can provide a result that can be used elsewhere in the program. This result can take various forms, including numbers, strings, lists, objects, and more, depending on the programming language and the function's purpose. Additionally, functions can also be defined to return no value at all, which is commonly represented by returning `None` in Python or simply not including a return statement in other languages. This characteristic allows functions to be flexible and serve a variety of roles in managing program flow and logic, such as performing an action without necessarily needing to return data. Understanding that functions can return different types of values or nothing provides a foundation for effective programming, as it highlights the importance of function design and how results from functions are utilized within the wider context of a program.