

KillTest

質量更高 服務更好



練習題

<https://www.killtest.net>

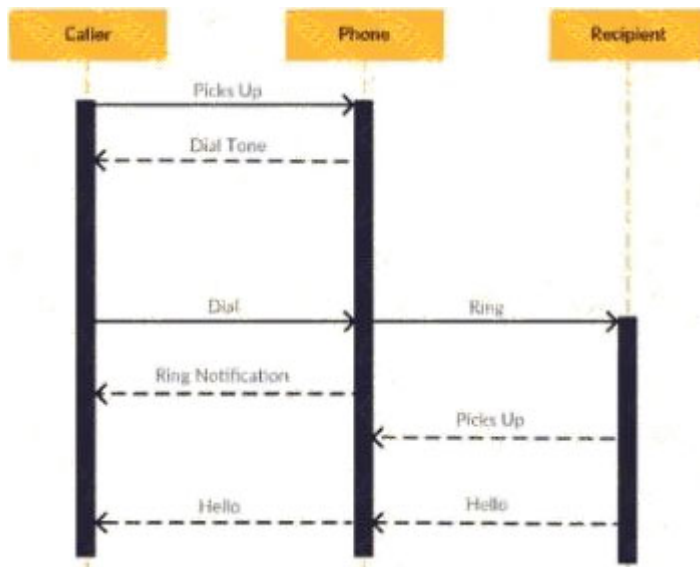
一年免費更新服務

**Exam : Scripting and
Programming Foundations**

**Title : WGU Scripting and
Programming Foundations
Exam**

Version : DEMO

1.Review the following sequence diagram:



What does a sequence diagram do?

- A. Shows interactions and indicates an order of events
- B. Shows interactions but does not specify an order of events
- C. Shows static elements of software
- D. Shows an order of events but does not specify all interactions

Answer: A

Explanation:

A sequence diagram is a type of interaction diagram used in software engineering to model the interactions between objects within a system over time. It shows how objects interact with each other and the order in which those interactions occur. The sequence diagram is organized along two dimensions: horizontally to represent the objects involved, and vertically to represent the time sequence of interactions. This allows the diagram to depict not only the interactions but also the sequence of events as they occur over time.

Option B is incorrect because a sequence diagram does indeed specify an order of events.

Option C is incorrect as sequence diagrams do not show static elements of software but rather the dynamic interactions.

Option D is also incorrect because a sequence diagram does show all interactions along with their order.

2.A software developer creates a list of all objects and functions that will be used in a board game application and then begins to write the code for each object.

- A. Analysis and implementation
- B. Analysis and design
- C. Design and implementation
- D. Design and testing

Answer: C

Explanation:

The process described involves two main phases: first, the developer is designing the application by creating a list of all objects and functions (the design phase), and then they are writing the code for each object (the implementation phase). This aligns with option C, Design and Implementation. Analysis would

involve understanding the requirements or problems the software will address, which is not mentioned in the scenario. Testing is a separate phase that typically occurs after implementation to ensure the code works as intended.

3.A programmer receives requirements from customers and decides to build a first version of a program. Which phase of an Agile approach is being carried out when the programmer starts writing the first version?

- A. Implementation
- B. Testing
- C. Design
- D. Analysis

Answer: A

Explanation:

From Exact Extract:

In Agile software development, the process is iterative, and writing code to create a version of the program occurs during the implementation phase. According to foundational programming principles and Agile methodologies (e.g., Certiport Scripting and Programming Foundations Study Guide, Agile Manifesto), implementation involves coding the software based on requirements and design.

Agile Phases Overview:

Analysis: Gathers and refines requirements (e.g., customer needs as user stories).

Design: Plans the technical solution (e.g., defining functions, classes, or architecture).

Implementation: Writes and integrates code to create a working version.

Testing: Verifies the code meets requirements.

Option A: "Implementation." This is correct. Starting to write the first version of the program involves coding, which is the core activity of the implementation phase. For example, the programmer might write functions or classes to meet customer requirements.

Option B: "Testing." This is incorrect. Testing occurs after coding to verify the program's functionality, not during the writing of the first version.

Option C: "Design." This is incorrect. Design involves planning the program's structure (e.g., specifying functions or objects), not writing the code.

Option D: "Analysis." This is incorrect. Analysis involves receiving and refining requirements, which the programmer has already done before starting to code.

Certiport Scripting and Programming Foundations Study Guide (Section on Agile Implementation).

[Agile](#)

Manifesto: "Working Software" (<http://agilemanifesto.org/>).

Sommerville, I., Software Engineering, 10th Edition (Chapter 4: Agile Software Development).

4.Which two statements describe advantages to using programming libraries?

- A. Using a library minimizes copyright issues in coding
- B. A program that uses libraries is more portable than one that does not.
- C. Using libraries turns procedural code into object-oriented code.
- D. Libraries always make code run faster.
- E. The programmer can improve productivity by using libraries.
- F. Using a library prevents a programmer from having to code common tasks by hand.

Answer: E F

E. The programmer can improve productivity by using libraries.

Why: Libraries offer pre-written, tested code for common tasks. This saves developers time and effort, leading to increased productivity.

F. Using a library prevents a programmer from having to code common tasks by hand.

Why: The core purpose of libraries is to provide reusable code solutions. This eliminates the need to reinvent the wheel for frequently used functions and operations.

5.What is required for all function calls?

A. Parameters

B. Input arguments

C. Output values

D. Function name

Answer: D

Explanation:

When calling a function in Python, you simply give the name of the function followed by parentheses.

Even if the function doesn't take any arguments, you still need to include the parentheses. For example, `print ("Hello!")` is a function call. The function name should describe what it's supposed to do. Function definitions begin with the `def` keyword, followed by the function name and parameters (if any). The statements within the function definition are indented and carry out the task the function is supposed to perform².

References:

[Function Calls and Definitions – Real Python](#)

[Function Calls | Microsoft Learn](#)

[Stack Overflow: Find all function calls by a function](#)