1. Create an assert statement that throws an AssertionError if the variable spam is a negative integer.

ANSWER.

assert spam >= 0

2. Write an assert statement that triggers an AssertionError if the variables eggs and bacon contain strings that are the same as each other, even if their cases are different (that is, 'hello' and 'hello' are considered the same, and 'goodbye' and 'GOODbye' are also considered the same).

ANSWER.

assert eggs.lower() != bacon.lower()

3. Create an assert statement that throws an AssertionError every time.

ANSWER.

assert False

4. What are the two lines that must be present in your software in order to call logging.debug()?

ANSWER.

import logging

logging.basicConfig(level=logging.DEBUG)

5. What are the two lines that your program must have in order to have logging.debug() send a logging message to a file named programLog.txt?

ANSWER.

import logging

logging.basicConfig(filename='programLog.txt', level=logging.DEBUG)

6. What are the five levels of logging?

ANSWER.

DEBUG, INFO, WARNING, ERROR, CRITICAL.

7. What line of code would you add to your software to disable all logging messages?

ANSWER.

logging.disable(logging.CRITICAL + 1)

8.Why is using logging messages better than using print() to display the same message?

ANSWER.

Using logging messages provides a more structured, flexible, and efficient approach to logging compared to print() statements, making it the preferred choice for logging in most software applications.

9. What are the differences between the Step Over, Step In, and Step Out buttons in the debugger?

ANSWER.

Step Over: executes the current line of code and advances to the next line, stepping over function calls.

Step In: executes the current line of code and steps into function calls, allowing you to debug inside called functions.

Step Out: completes the execution of the current function and returns to the caller, allowing you to quickly move to the next line after the function call.

10.After you click Continue, when will the debugger stop ?

ANSWER.

The debugger will stop after clicking "Continue" when it encounters a breakpoint, an unhandled exception, or reaches the end of the program's execution.

11. What is the concept of a breakpoint?

ANSWER.

Breakpoints are an essential tool for debugging software, enabling developers to effectively diagnose and troubleshoot problems in their code by pausing execution and inspecting program state at specific points in the code.