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

Ans-> assert spam >= 0, "spam should be a non-negative integer"

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).

Ans-

eggs = "hello"

bacon = "HELLO"

assert eggs.lower() != bacon.lower(), "eggs and bacon should have different values (case-insensitive)"

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

Ans-> assert False, "This assert statement always triggers an AssertionError"

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

Ans->

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?

Ans->

import logging

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

6. What are the five levels of logging?

Ans->

DEBUG,INFO,WARNING,ERROR,CRITICAL

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

Ans->

logging.disable(logging.CRITICAL)

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

Ans->

Using logging messages offers greater control, flexibility, and maintainability compared to using print() statements, making it the preferred approach for professional software development.

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

Ans->

Step Over executes the current line and moves to the next line, skipping over function calls.

Step In steps into a function call, moving the debugger to the first line of the called function.

Step Out executes the remaining lines of the current function and returns to the caller.

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

Ans->

If the program completes its execution without encountering any breakpoints or exceptions, the debugger will stop when it reaches the end of the program.

11. What is the concept of a breakpoint?

Ans->

Breakpoint is a designated point in your code where the debugger will pause program execution, allowing you to inspect the program's state and variables at that particular moment. It's a helpful tool for interactive debugging and understanding the behavior of your code.