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

Ans. spam = -5 # Assuming spam is a variable

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' # Assuming eggs is a variable

bacon = 'hello' # Assuming bacon is a variable

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

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?

DEBUG: The DEBUG level is used for detailed debugging information. It is typically used for low-level information that can be useful during development or when troubleshooting specific issues.

INFO: The INFO level is used to provide general informational messages about the program's execution. It is used to convey important events or milestones during the normal operation of the program.

WARNING: The WARNING level is used to indicate potential issues or situations that could lead to errors or unexpected behavior. It highlights conditions that may require attention but do not necessarily cause the program to fail.

ERROR: The ERROR level is used to indicate errors that occur during the program's execution. It represents more severe issues that may prevent the program from functioning as intended.

CRITICAL: The CRITICAL level is the highest severity level. It is used to indicate critical errors or failures that may lead to the termination of the program or cause significant issues in the system.

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. While print() statements can be useful for quick and simple debugging during development, logging messages provide a more robust and scalable approach for handling application output in a professional software development context.

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 the details of function calls.

Step In enters into a function call, allowing you to step through the function's execution line by line.

Step Out completes the execution of the current function and returns to the calling function.

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

Breakpoint: If there is an active breakpoint set at a specific line of code, the debugger will halt the execution when it reaches that line. Breakpoints are useful for pausing the program at specific points to inspect variables, step through code, or analyze the program state.

Exception: If an exception occurs during the program's execution and it is not caught by an exception handler, the debugger will stop and display an error message. This allows you to examine the exception details and understand why it was raised.

Program completion: If the program successfully executes all the statements and reaches its end without encountering any breakpoints or exceptions, the debugger will stop. This indicates that the program has completed its execution.

11. What is the concept of a breakpoint?

Breakpoints are useful for pausing the program at specific points to inspect variables, step through code, or analyze the program state.