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

**assert spam >= 0, "AssertionError: Variable 'spam' should not be a negative integer"**

1. 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).

**This assert statement converts both eggs and bacon to lowercase and compares them. If they are the same, an AssertionError will be raised with the specified message**.

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

**This assert statement will always evaluate to False, which means it will always raise an AssertionError with the specified message**.

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

**import logging**

1. 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?

**With** **these lines in place, logging.debug() will write its messages to programLog.txt**

1. What are the five levels of logging?

**Debug, Info, warning, error, critical**

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

**logging.disable(logging.CRITICAL)**

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

**The logging module provides a more structured, configurable, and maintainable way to handle logging messages compared to using print()**

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

**When debugging code, the Step Over, Step In, and Step Out buttons help you control the flow of execution to better understand what's happening**

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

**Breakpoint**: The debugger will stop at the next breakpoint that is hit in the code. Breakpoints are set by the user at specific lines of code.

**Exception**: If an unhandled exception is raised during the execution, the debugger will stop at the line where the exception occurred.

**Program Exit**: The debugger will stop when the program reaches its end or exits normally.

1. What is the concept of a breakpoint?

**Breakpoints make debugging more efficient and effective by letting you pause and examine the program's execution in a controlled manner.**