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

try:

assert spam<0, “spam shouldn’t be negative”

Print(f’No error found ,so spam is correct’)

except AssertionError as msg:

Print(msg)

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).

Try:

assert eggs.lower()!=bacon.lower(), “Both contain same string”

print(“both doesn’t contain same string”)

except AssertionError as msg:

print(msg)

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

Try:

Assert False, “a is always 0”

Print(“value is correct”)

Except AssertionError as msg:

Print(msg)

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

To be able to call logging.debug(), you must have these two lines at the start of your program: import logging logging.basicConfig(level=logging.DEBUG, format=' %(asctime)s - %(levelname)s - %(message)s')

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?

import logging logging.basicConfig(filename='programLog.txt', level=logging.DEBUG, format=' %(asctime)s - %(levelname)s - %(message)s')

1. What are the five levels of logging?

DEBUG, INFO, WARNING, ERROR, and CRITICAL

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

logging.disable(logging.CRITICAL)

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

You can disable logging messages without removing the logging function calls. You can selectively disable lower-level logging messages. You can create logging messages. Logging messages provides a timestamp.

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

Step Over command:

This command lets the application execute the next action. If the action involves a call to an operation, it does not step into its implementation (it steps over it instead).

Step Into command:

This command lets the application execute the next action. If the action involves a call to an operation, it steps into its implementation and breaks the execution on the first action of that implementation.

Step Out command:

This command lets the application execute until the currently executed operation implementation is returned.

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

Continue execution until the current function returns. Continue execution, only stop when a breakpoint is encountered.

11. What is the concept of a breakpoint?

Breakpoint is a setting on a line of code that causes the debugger to pause when the program execution reaches the line