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

# assert spam > 0

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).

# assert eggs.lower() != bacon.lower(), 'The eggs and bacon variables are the same!' or assert eggs.upper() != bacon.upper(), 'The eggs and bacon variables are the same!'

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

# assert False, 'This assertion always triggers.'

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

# 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)

1. 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 In will cause the debugger to go into the next function call and break there. Step Over will tell the debugger to execute the next function and break afterwards. Step Out will tell the debugger to finish the current function and break after it

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

# Continue will resume program execution until your program completes normally.

1. What is the concept of a breakpoint?

# The Python breakpoint() built-in function is a tool that allows developers to set points in code at which a debugger is called. By default, this function results in an instantiation of Python's native debugger class