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

Ans:

spam = int(input("Enter an integer: "))

try:

assert spam > 0, "negative integer"

print(spam)

except AssertionError as ae:

print(ae)

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 = input("Enter String1: ")

bacon = input("Enter String2: ")

assert eggs.lower() != bacon.lower()

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

Ans: assert True == False

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, format=' %(asctime)s - %(levelname)s - %(message)s')

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, format=' %(asctime)s - %(levelname)s - %(message)s')

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:

i. Track the certain logging time

ii. We can track the logging based on its severity.

iii. Can store the log messages in custom filename which is specified.

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

Ans:

The Step in button will move the debugger into a function call.

The Step Over button will quickly execute the function call without stepping into it.

The Step Out button will quickly execute the rest of the code until it steps out of the function it currently is in.

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

Ans: Continues execution, only stop when a breakpoint is encountered.

11. What is the concept of a breakpoint?

Ans: A breakpoint is an intentional stopping or pausing place in a program, put in place for debugging purposes. It is also sometimes simply referred to as a pause.