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

integer.

ANS assert spam >= 10, 'The spam variable is less than 10.'

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, &#39;hello&#39; and &#39;hello&#39; are

considered the same, and &#39;goodbye&#39; and &#39;GOODbye&#39; are also considered the same).

ANS spam = 100

# Remember, if you want it to throw error if it is less than 10

# then you have to test if it is greater than 10.

# Less than 10 is the error condition, not the expression for assert.

assert spam >= 10, 'Your spam is less than 10!'

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

1. ANS 1.4.1 Enable Assertions.
2. 1.4.2 Review the AssertionError Message.
3. 1.4.3 Identify the Failing Assertion.
4. 1.4.4 Print Variable Values.
5. 1.4.5 Evaluate Expressions Separately.

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

ANS The two lines that your program must have in order to be able to call logging.debug() are: ``` 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 If you want to send a**logging message** to a file named programLog.txt, your program must have two lines: *importing the logging module and configuring it to write messages to the file.*

**Explanation:**

In **Python programming**, the option to send a logging message to a specific file like programLog.txt is made possible through the logging module. To send a logging message to a file named **programLog.tx**t, your program must have two lines.

* The first line is to import the logging module using the import keyword.
* The second line is to configure the logging module to write messages to the file using the **basicConfig() function.**

1. What are the five levels of logging?

* ANS Emergency. Emergency logs are given the numerical value "0". ...
* Alert. ...
* Critical. ...
* Error. ...
* Warning. ...
* Notice. ...
* Informational. ...
* Debug.

1. What line of code would you add to your software to disable all logging messages?
2. AN Temporarily change the log level during testing. This approach sets the log level to CRITICAL , which disables all logging with levels lower than or equal to CRITICAL . ...
3. Temporarily remove the log handlers during testing.

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

ANS The logging module provides a flexible way to log different messages in various output destinations such as on the console, in files, and on networks. Logging is a best practice for production code. The logging module provides features such as log levels and filtering.

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

* ANS Step Into: Step Into is used for debugging the test steps line by line. ...
* Step Over: Step Over will enable, only after the debugging is started with Step Into / Run From Step / Run to Step. ...
* Step Out: Step Out will enable, only after the debugging cursor gets inside a procedure.

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

ANS Click the Stop button on the toolbar of the Debug tool window. Alternatively, press Ctrl F2 and select the process to terminate (if there are two or more of them).

11. What is the concept of a breakpoint?

ANS In software development, 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.

FINISH\*\*\*\*\*