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

**integer.**

**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).**

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

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

**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?**

**6. What are the five levels of logging?**

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

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

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

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

**11. What is the concept of a breakpoint?**

**SOLUTIONS**

1. *Here is an assert statement that throws an AssertionError if the variable "spam" is a negative integer:*

***assert spam >= 0, "Spam should not be a negative integer"***

1. *Here is an assert statement that triggers an AssertionError if the variables "eggs" and "bacon" contain strings that are the same, even if their cases are different:*

***assert eggs.lower() != bacon.lower(), "Eggs and Bacon should not be the same string"***

1. *Here is an assert statement that throws an AssertionError every time:*

***assert False, "This assert statement will always trigger an AssertionError"***

1. *To call* ***logging.debug()****, you need to have the following two lines in your software:*

***import logging***

***logging.basicConfig(level=logging.DEBUG)***

1. *To have* ***logging.debug()*** *send a logging message to a file named* ***programLog.txt****, you need to have the following two lines in your program:*

***import logging***

***logging.basicConfig(filename='programLog.txt', level=logging.DEBUG)***

1. *The five levels of logging are, in order of increasing severity: DEBUG, INFO, WARNING, ERROR, and CRITICAL.*
2. *To disable all logging messages, you can set the logging level to NOTSET:*

***import logging***

***logging.basicConfig(level=logging.NOTSET)***

1. *Using logging messages is better than using* ***print()*** *to display the same message because logging provides a more flexible and systematic way to handle the messages. With logging, you can specify the level of severity, decide where the messages are stored (e.g., file, console), and filter the messages based on their severity.*
2. *In a debugger, the "Step Over" button allows you to step over a function or method call, executing it in one step and moving to the next line of code. The "Step In" button allows you to step into a function or method call, so you can see its execution line by line. The "Step Out" button allows you to step out of a function or method and continue executing the code at the next line after the call.*
3. *The debugger will stop when it reaches a breakpoint.*
4. *A breakpoint is a marker in your code that tells the debugger to pause execution at a certain point so you can inspect the state of the program and see what's going on.*