1. Create an assert statement that throws an AssertionError if the variable spam is a negative integer.
2. ANS: spam=-3  
   assert spam > 0,"Negative number"

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).

bacon="Hello"  
eggs="hello"  
assert bacon.lower()!=eggs.lower(),"Strings are same"

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

assert(False,"this statement raises assertion every time")

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

ANS: 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')

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='myProgramLog.txt', level=logging.DEBUG, format='

%(asctime)s - %(levelname)s - %(message)s')

6. What are the five levels of logging?

ANS:The five levels of logging are info,debug ,error , warning and critical

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

ANS: logging.disable()

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

ANS: Once you’re done debugging, you’ll end up spending a lot of time removing print() calls from your code for each log message. You might even accidentally remove some print() calls that were being used for nonlog messages..

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

ANS: Step Over

Clicking the Step Over button will execute the next line of code, similar to the Step In button. However, if the next line of code is a function call, the Step Over button will “step over” the code in the function. The function’s code will be executed at full speed, and the debugger will pause as soon as the function call returns

Step In

Clicking the Step In button will cause the debugger to execute the next line of code and then pause again. If the next line of code is a function call, the debugger will “step into” that function and jump to the first line of code of that function.

Step Out

Clicking the Step Out button will cause the debugger to execute lines of code at full speed until it returns from the current function

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

ANS: Clicking the Continue button will cause the program to execute normally until it terminates or reaches a breakpoint.

11. What is the concept of a breakpoint?

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