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

**integer.**

**Answer:-**  assert spam >= 0

**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).**

**Answer:-** assert eggs.lower() != bacon.lower() or assert eggs.upper() != bacon.upper()

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

**Answer:-** assert False

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

**Answer:-** The two lines 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?**

**Answer:-** The two lines are:

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?**

**Answer:-** Five levels are logging are as follows:

1. DEBUG
2. INFO
3. WARNING
4. ERROR
5. CRITICAL

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

**Answer:-** The line of code would be:

logging.disable(logging.CRITICAL)

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

**Answer:-** because in logging we get many options like we can disable these messages without removing logging function calls also we can disable them selectively and also it provides timestamp.

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

**Answer:-** 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 ?**

**Answer:-** The debugger will stop when it has reached the end of the program or a line with a breakpoint.

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

**Answer:-** A breakpoint is a point setting on a line of code that causes the debugger to pause when the program execution reaches the line.