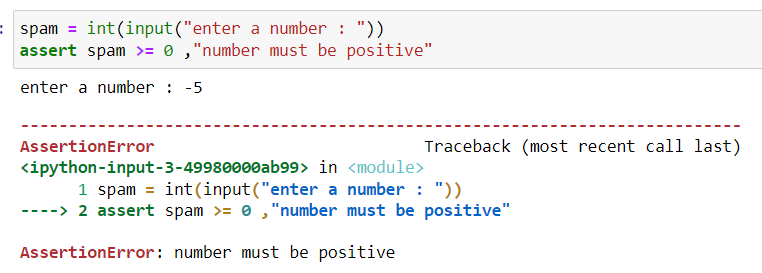
ASSIGNMENT NO 11

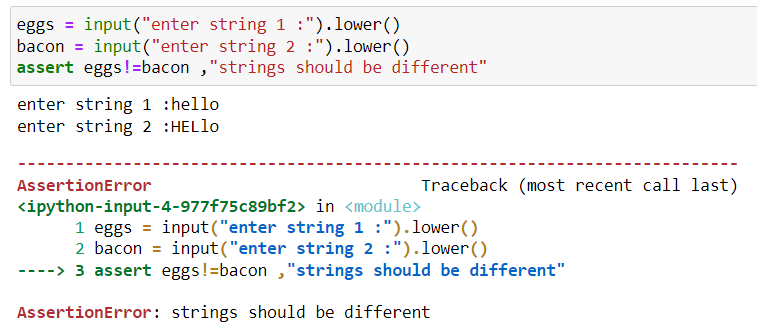
Nirbhay Ahir

[Nirbhay.ahir36@gmail.com](mailto:Nirbhay.ahir36@gmail.com)

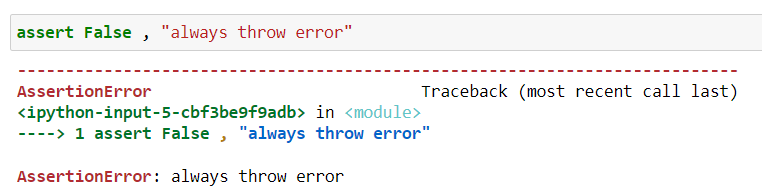
Q1 .Create an assert statement that throws an AssertionError if the variable spam is a negative integer.

Ans: 

Q2. 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: 

Q3. Create an assert statement that throws an AssertionError every time.

Ans: 

Q4. What are the two lines that must be present in your software in order to call logging.debug()?

Ans: The two lines that are always required are :

Import logging

logging.basicConfig(filename = “file.txt” , level=logging.DEBUG , format=' %(asctime)s - %(levelname)s - %(message)s'

Q5.  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: The two lines that are required are :

import logging

logging.basicConfig(filename = 'programLog.txt',level=logging.DEBUG, format=' %(asctime)s - %(levelname)s - %(message)s')

logging.debug("Logging message is added")

Q6.What are the five levels of logging?

Ans: The five levels of logging are :

NOTSET(O) , DEBUG(10), INFO(20) , WARN(30) , ERROR(40) , CRITICAL(50)

Q7. What line of code would you add to your software to disable all logging messages?

Ans: logging.disable = True

Q8. Why is using logging messages better than using print() to display the same message?

Ans: If we want to remove the messages then we can simply disable the logging messages without meddling with the logging function whereas we would need to remove the print statements manually which is tiring activity.

Also, print is used when we want to display any particular message or help whereas logging is used to record all events such as error, info, debug messages, timestamps.

Q9. What are the differences between the Step Over, Step In, and Step Out buttons in the debugger?

Ans: Step Over: It causes the debugger to executes the next line and if there is a function in that line, then debugger will "step over" into the function and execute it until the function returns, and then the debugger pauses.

Step In: It causes the debugger to executes the next line and then pause again.

Step Out: It causes the debugger to execute the lines of code until the current function returns.

Q10. After you click Continue, when will the debugger stop ?

Ans: The debugger will stop either when it reaches the breakpoint or when the whole program is executed.

Q11. What is the concept of a breakpoint?

Ans: Breakpoint is a point on a line of code that causes the debugger to pause when the program execution reaches the line where breakpoint is given.