Lab 2:

Linked list implementation of Stack ADT:

In this lab we will implement one part of Project 1, which is the linked list implementation of a stack. Download the stack files from the project 1 files link and copy the stack_linked_list.py into your lab2 folder.

Create a Stack abstract data type (ADT) using the Linked list structure. This requires the use of classes in Python. You will need to create a class for 'Node', which will hold two variables: one is the item being added to the list, and the other is the address of the next node (a variable of type 'Node' containing the next node). If there is no next 'Node', this variable must be set to 'None'.

The stack must have a capacity, i.e. maximum number of elements to hold, a size, i.e. the number of elements currently on the stack, and a head, i.e. the first elements on the stack's list.

Implement the following methods for your stack:

```
def is_empty(self):
    """Returns true if the stack self is empty and false otherwise"""

def is_full(self):
    """Returns true if the stack self is full and false otherwise"""

def push(self, item):
    """Adds item to the stack"""

def pop(self):
    """Returns the current top of the stack"""

def peek(self):
    """Returns the value of the item at the top of the stack without removing it"""

def size(self):
    """Returns the number of elements currently in the stack, not the capacity"""
```

Create enough test cases to test all aspects of your implementation. Make sure you test every function (method) with enough cases.

Demonstrate your work to your instructor during the lab hour on the due date.