**CSC002 Tutorial 2**

Q1) State what ‘FIFO’ and ‘LIFO’ stands for.

Q2) The following series of expressions are executed (with the given Stack class):

1) my\_stack = Stack()

2) my\_stack.push(5)

3) my\_stack.push(10)

4) my\_stack.push(15)

5) my\_stack.peek()

6) my\_stack.pop()

7) my\_stack.pop()

8) my\_stack.pop()

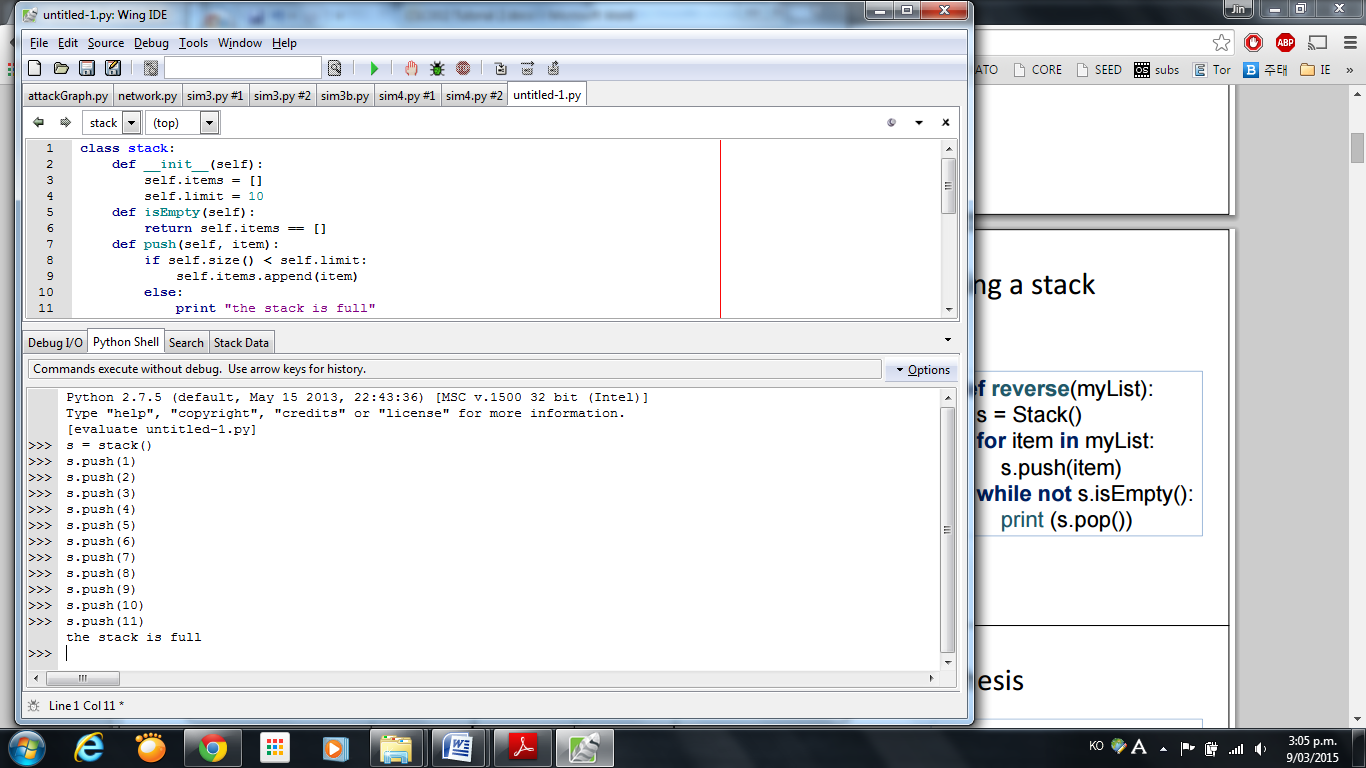
9) my\_stack.peek()

10) my\_stack.pop()

Describe what happens at lines (5), (8), and (10).

Q3) Write a LimitedStack class, which has the same properties as the *Stack* class but the size of the stack is limited by 10 items. If the stack is full, then no more items are added to the stack and prints the error message “the stack is full”.

For example:



Q4) Provide the Big Oh notation for the following:

a) For a Stack that uses index 0 for the top, what is the order of inserting an item?

b) For a Stack that uses index 0 for the top, what is the order of removing an item?

c) For a Queue that uses 0 for the back, what is the order of inserting an item?

d) For a Queue that uses 0 for the back, what is the order of removing an item?

e) For c) or d) what change can you make to the data structure so that both becomes O(1)?

Q5) Show the state of the stack as the following post-fix expression is *evaluated* and state the resulting value. You should use the algorithm covered in lectures.

3, 10, 5, -, 7, \*, +

Q6) Show the state of the stack as the following post-fix expression is *evaluated* and state the resulting value. You should use the algorithm covered in lectures.

3, 10, \*, 5, 7, -, +, 2, \*

Q7) *Convert* the following expression to post-fix using the algorithm outlined in lectures, make sure you show the state of the stack and the output as you go.

3 \* 2 - 1 \* 3

Q8) *Convert* the following expression to post-fix using the algorithm outlined in lectures, make sure you show the state of the stack and the output as you go:

(3 \* 10 – 5 + 7) \* 2

Q9) *Convert* the following expression to post-fix using the algorithm outlined in lectures, make sure you show the state of the stack and the output as you go:

(3 \* (10 – 5) - 2) \* 5