UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 8

Stacks

Submitted by:
Ramos, Jan Lawremce M.

Instructor:
Engr. Maria Rizette H. Sayo

October 4, 2025

I. Objectives

Introduction

A stack is a collection of objects that are inserted and removed according to the last-in, first-out (LIFO) principle.

A user may insert objects into a stack at any time, but may only access or remove the most recently inserted object that remains (at the so-called “top” of the stack)

This laboratory activity aims to implement the principles and techniques in:

- Writing Python program using Stack
- Writing a Python program that will implement Stack operations

II. Methods

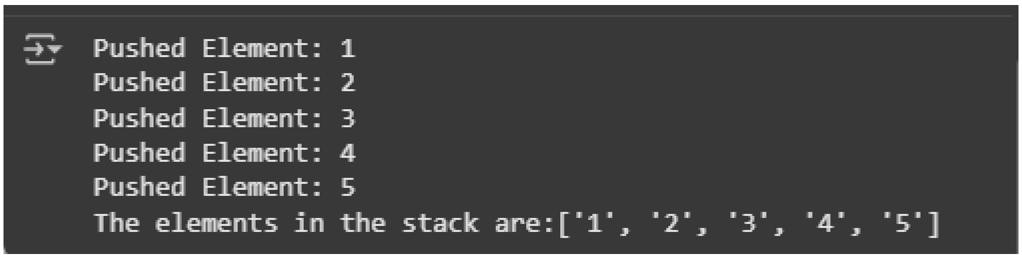
Instruction: Type the python codes below in your Colab. After running your codes, answer the questions below.

```
# Stack implementation in python


# Creating a stack
def create_stack():
    stack = []
    return stack


# Creating an empty stack
def is_empty(stack):
    return len(stack) == 0

# Adding items into the stack
def push(stack, item):
    stack.append(item)
    print("Pushed Element: " + item)

# Removing an element from the stack
def pop(stack):
    if (is_empty(stack)):
        return "The stack is empty"
    return stack.pop()

stack = create_stack()
push(stack, str(1))
push(stack, str(2))
push(stack, str(3))
push(stack, str(4))
push(stack, str(5))

print("The elements in the stack are:"+ str(stack))
```

III. Results

Program Output Analysis

After executing the provided code, the output demonstrates the stack operations:



```
Pushed Element: 1
Pushed Element: 2
Pushed Element: 3
Pushed Element: 4
Pushed Element: 5
The elements in the stack are:['1', '2', '3', '4', '5']
```

Figure 1: this shows the output of the program provided by the lab.

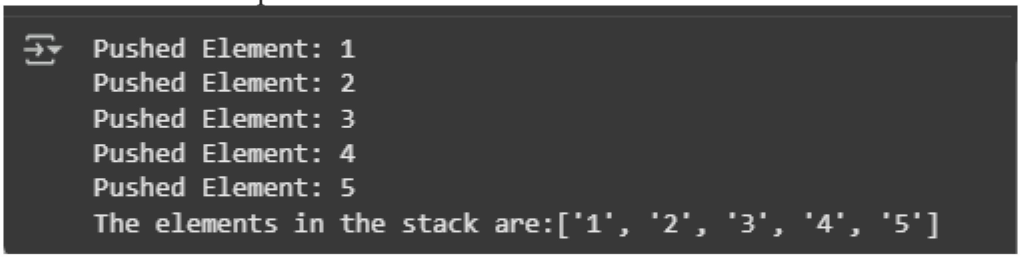
Answer the following questions:

1 Upon typing the codes, what is the name of the abstract data type? How is it implemented?

- The abstract data type implemented is a **Stack**. It is implemented using Python's built-in list data structure, where elements are added to and removed from the same end (the top) using append() and pop() operations, maintaining the LIFO (Last-In, First-Out) principle.
-

2 What is the output of the codes?

- The output shows five elements being pushed onto the stack in sequential order (1 through 5), with the final stack containing all elements in the order ['1', '2', '3', '4', '5'], where '5' is at the top of the stack.

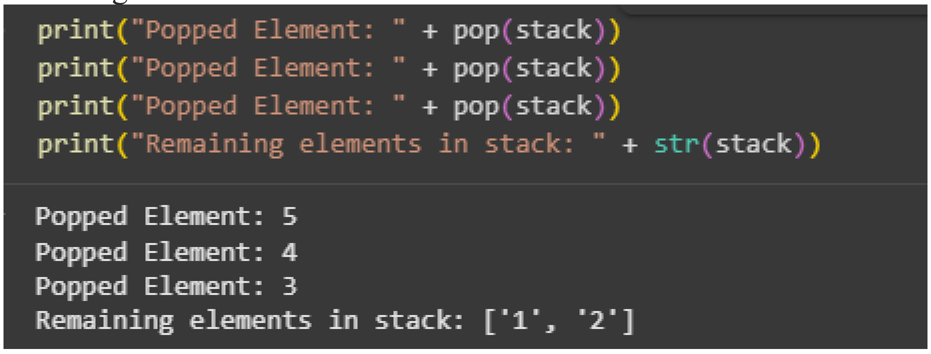


```
Pushed Element: 1
Pushed Element: 2
Pushed Element: 3
Pushed Element: 4
Pushed Element: 5
The elements in the stack are:['1', '2', '3', '4', '5']
```

Figure 2: this shows the output of the program provided by the lab.

3 If you want to type additional codes, what will be the statement to pop 3 elements from the top of the stack?

- Popping Three Elements: To pop three elements from the top of the stack, the following code would be added:



```
print("Popped Element: " + pop(stack))
print("Popped Element: " + pop(stack))
print("Popped Element: " + pop(stack))
print("Remaining elements in stack: " + str(stack))
```

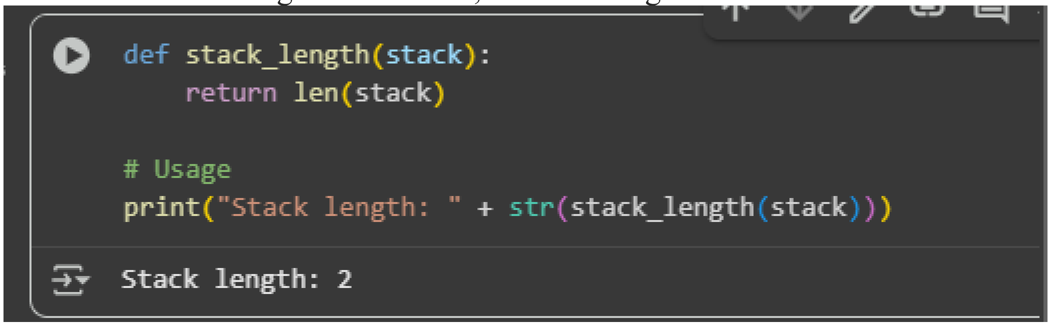
```
Popped Element: 5
Popped Element: 4
Popped Element: 3
Remaining elements in stack: ['1', '2']
```

Figure 3: shows the additional program.

This would remove elements '5', '4', and '3' from the stack, leaving ['1', '2'].

4 **If you will revise the codes, what will be the statement to determine the length of the stack? (Note: You may add additional methods to count the no. of elements in the stack)**

- To determine the length of the stack, the following method can be added:



```python
def stack_length(stack):
    return len(stack)

# Usage
print("Stack length: " + str(stack_length(stack)))
```

```
Stack length: 2
```

Figure 4: this shows the additional revised code.

IV. Conclusion

In conclution, this lab really helped me understand how a stack works by letting me actually build one in Python. I learned that it's all about the last-in, first-out rule, kind of like stacking books where you can only take the top one off. By using simple list commands like append and pop, I could make the stack add and remove items exactly how it's supposed to. This wasn't just about typing code, it showed me how important stacks are for real things like the undo button or going back on a web browser.

References

[1] Co Arthur O.. “University of Caloocan City Computer Engineering Department Honor Code,” UCC-CpE Departmental Policies, 2020.

[2] Python Software Foundation, "Data Structures in Python," Python Documentation, 2023. [Online]. Available: https://docs.python.org/3/tutorial/datastructures.html

[3] GeeksforGeeks, "Stack Data Structure," 2023. [Online]. Available: https://www.geeksforgeeks.org/stack-data-structure/

[4] Programiz, "Stack in Python Programming," 2023. [Online]. Available: https://www.programiz.com/python-programming/stack