UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 8

Stacks

Submitted by:
Acebedo, Sebastian C.

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.

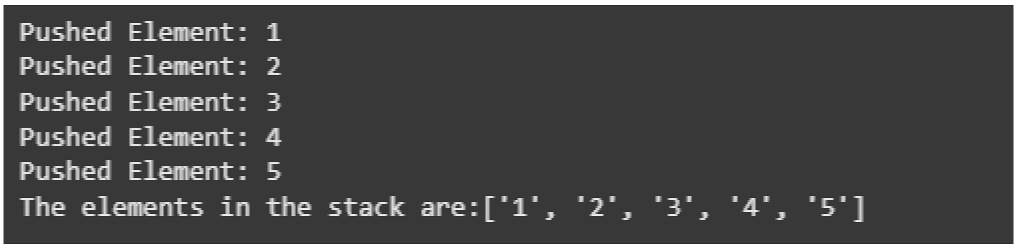
```
# Stack implementation in python


# 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))
```

Answer the following questions:

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

- The data type implemented in the program is Stacks. Stacks operate on the LIFO (last in, first out) principle. This principle means that the last element added to the ask are the first to be removed.

2 What is the output of the codes?



```
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']
```

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

- To pop the 3 element from the top of the stack, I can use the statement;
- print("Popped element:", pop(stack))
- print("Popped element:", pop(stack))
- print("Popped element:", pop(stack))

or using for loop:

- for i in range(3):
        print("Popped element:", pop(stack))

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)

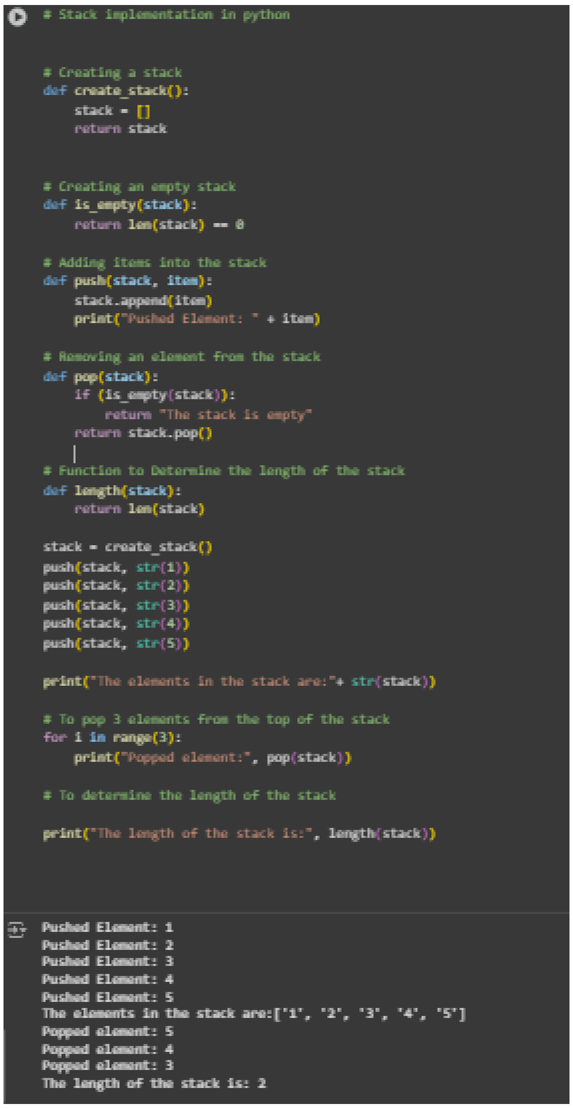
- There are 2 ways to for me to determine the length of the stack. The first one;
- print("Length of the Stack:", len(stack))

or add a helper funtion to determine the length of the stack

- def length(stack):
    return len(stack)

print("The length of the stack is:", length(stack))

III. Results



```
# 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()

# Function to Determine the length of the stack
def length(stack):
    return len(stack)

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))

# To pop 3 elements from the top of the stack
for i in range(3):
    print("Popped element:", pop(stack))

# To determine the length of the stack

print("The length of the stack is:", length(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']
Popped element: 5
Popped element: 4
Popped element: 3
The length of the stack is: 2
```

Figure 1 Output of program

This image shows the whole program with the revised code based on the given questions.

IV. Conclusion

From this activity, I learned that a stack is an abstract data type that follows the Last In, First Out (LIFO) principle. It can be easily implemented in Python using lists with the append() and pop() methods. The output of the code shows the elements pushed into the stack, and by adding extra statements, we can also pop elements from the top or check the number of items in the stack. This exercise helped me understand how stacks work and how to apply basic operations like push, pop, peek, and size in a program.

References

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