



Data Structure and Algorithm

Laboratory Activity No. 8

Stacks

Submitted by:

Mamanao, Kurt Marwin C.

Instructor:

Engr. Maria Rizette H. Sayo

Oct, 04, 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?
- 2 What is the output of the codes?
- 3 If you want to type additional codes, what will be the statement to pop 3 elements from the top of the 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)

III. Results

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

- The abstract data type is a **Stack**. It is made in Python using a **list**, where `append()` adds items (push) and `pop()` removes items (pop).

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']
```

This is the output of the raw code given in the activity, where it only execute pushing the elements from 1 – 5.

```
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']
The length of the stack is: 5
```

And now this is the output codes from the modified raw code base on the answered Guide questions, beside pushing the elements it has now pop that removes the 3 elements and Len that count the elements after popping

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

```
# Question No. 3
# POP 3 elements from the stack
pop(stack)
pop(stack)
pop(stack)
```

What I did is just add `pop(stack)` 3 times in the code to remove the top of the code elements 3 times.

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)

```
# Question No.4
def size(stack):
    return len(stack)
```

So to find how many elements are in the stack, I added the function **size()** that uses **len(stack)**. And this lets me count the number of elements inside the stack.

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

and when I call this function, it will print the length of the stack that shows how many elements are still in the stack.

IV. Conclusion

In this activity, I had a chance to practice what I learned in our previous lessons about stacks by modifying the given code base on the guide question provided, I added **pop(stack)** to remove the top element in the code that follows LIFO or Last in First out, and I determine the length of the code after popping by using **size()** accompanied by **Len(stacks)**.

References

- [1] W3Schools, “Python Data Structures - Stack,” *W3Schools.com*. <https://www.w3schools.com> (accessed Oct. 4, 2025).