Data Structure and Algorithm

Laboratory Activity No. 8

Stacks

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Calica, Ljay L. | Engr. Maria Rizette H. Sayo |

Oct, 04, 2025

# 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

# 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 abstract data type is a Stack. It is implemented in Python using a **list** and its built-in methods (append() for push and pop() for pop).
2. What is the output of the codes?

// The code pushes elements 1, 2, 3, 4, 5 into the stack and then prints them.

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

// You can simply call pop() three times. This will remove '5', '4', and '3' (last-in, first-out order).

1. 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)

// We can add a new function size()

# Results

The program's execution in Google Colab showed how stack operations are implemented in Python. The software correctly displayed the elements stored in the stack once they were placed into it one after the other. The last-in, first-out (LIFO) idea was followed by the push and pop functions.

The output of the program is shown in Figure 1.

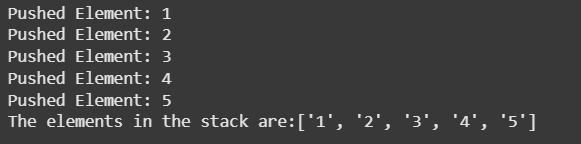


Figure 1 Screenshot of program

When additional statements were added to pop three elements from the stack, the following output was produced:

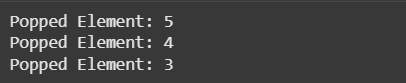


Figure 2 Screenshot of program

After popping, the stack contained the remaining elements:



Figure 3 Screenshot of program

When a method for determining the stack length was added, the program displayed the correct number of elements in the stack.



Figure 4 Screenshot of program

# Conclusion

This lab activity demonstrated how to create a stack abstract data type (ADT) in Python using lists. By performing push, pop, and size determination, the software illustrated the fundamental stack concept of last-in, first-out (LIFO). The experiment also showed how stack operations can be implemented effectively using Python's built-in append() and pop() routines. By providing us with practical experience implementing stack operations, this exercise improved our comprehension of abstract data types and their applications in programming.

**References**

[1] Liang, Y. D. (2020). *Introduction to programming using Python* (2nd ed.). Pearson.

[2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.