Data Structure and Algorithm

Laboratory Activity No. 8

Stacks

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| LastName, FirstName MI. | Engr. Maria Rizette H. Sayo |

OCTOBER 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?
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)

# Results

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

* The abstract data type used in the program is a stack. It is implemented by initializing an empty stack first, followed by the functions *is\_empty()*, *push()*, and *pop()*. After the stack was created, the numbers 1, 2, 3, 4, and 5 were pushed inside the stack.

1. What is the output of the codes?

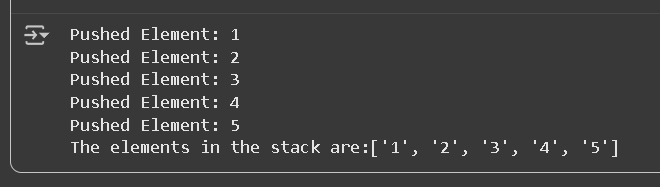
* The output of the code displays the numbers pushed into the stack and its current elements which are [‘1’, ‘2’, ‘3’, ‘4’, ‘5’]

Figure 1 Output of the Program

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

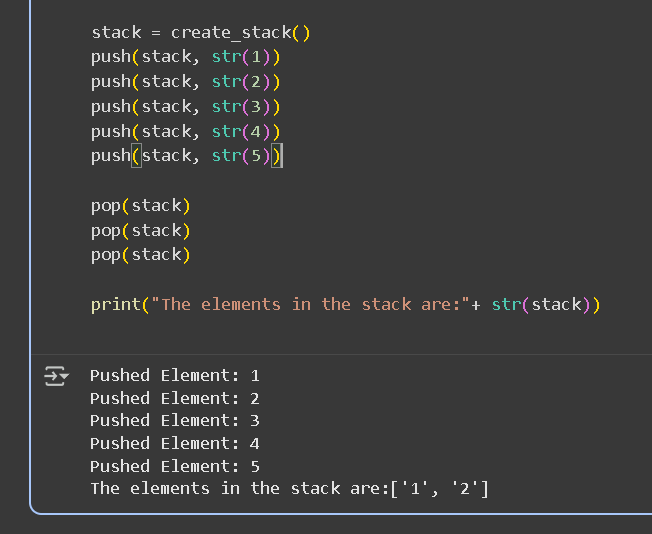
* We would call the *pop()* function three times to pop three elements in the stack.

Figure 2 Popping Three Items in the Stack

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)

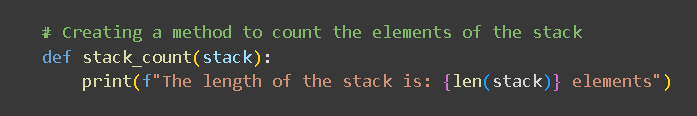
* The method added to the code will be shown below:

Figure 3 New Method for Getting the Length of the Stack

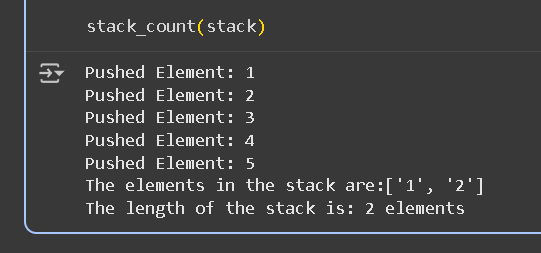


Figure 4 Calling the *stack\_count()* method and Program Output

# Conclusion

This laboratory activity demonstrates the process of implementing the stack abstract data type. It presents the fundamental principles behind stack operations such as push, pop, and getting the stack length, and highlights their practical applications in managing data with a Last-In, First-Out (LIFO) structure.