Data Structure and Algorithm

Laboratory Activity No. 8

Stacks

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Caasi, Karl Benedict D. | Engr. Maria Rizette H. Sayo |

October , 4, 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

Present the visualized procedures done. Also present the results with corresponding data visualizations such as graphs, charts, tables, or image . Please provide insights, commentaries, or explanations regarding the data. If an explanation requires the support of literature such as academic journals, books, magazines, reports, or web articles please cite and reference them using the IEEE format.

Please take note of the styles on the style ribbon as these would serve as the style format of this laboratory report. The body style is Times New Roman size 12, line spacing: 1.5. Body text should be in Justified alignment, while captions should be center-aligned. Images should be readable and include captions. Please refer to the sample below:

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

Answer: stack ADT  append () is used for the push operation, and

 pop () is used for the pop operation.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 1 Screenshot of program

2. What is the output of the codes?

A screenshot of a video

AI-generated content may be incorrect.

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)

**A black screen with white text

AI-generated content may be incorrect.**

**Conclusion**In this exercise, we used a list to effectively create a Stack Abstract Data Type (ADT) in Python. The application showed how to carry out fundamental stack operations, including checking to see if the stack is empty and pushing and popping pieces. This taught us that a stack operates on the Last In, First Out (LIFO) principle, which states that the last element added is the first one deleted. This idea is fundamental to computer science, particularly when it comes to handling recursive algorithms, managing function calls, and reversing data.

**Reference**Python Software Foundation. *“Data Structures — Python Lists.”* https://docs.python.org/3/tutorial/datastructures.html

W3Schools. *“Python Lists.”* https://www.w3schools.com/python/python\_lists.asp

GeeksforGeeks. *“Stack in Python.”* https://www.geeksforgeeks.org/stack-in-python/