



Data Structure and Algorithm

Laboratory Activity No. 8

Stacks

Submitted by:
LastName, FirstName MI.

Instructor:
Engr. Maria Rizette H. Sayo

Month, DD, YYYY

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 name of abstract data types implemented in this program is a STACK*
- 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?
 - *print("Popped:", pop(stack))*
 - *print("Popped:", pop(stack))*
 - *print("Popped:", 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)
 - *def size(stack):*
return len(stack)
 - *print("The length of the stack is:", size(stack))*

III. 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:

```
# 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))
```

Figure 1 Program

```
➡ 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']
```

Figure 2 Output

If an image is taken from another literature or intellectual property, please cite them accordingly in the caption. Always keep in mind the Honor Code [1] of our course to prevent failure due to academic dishonesty.

IV. Conclusion

*By doing this activity, I learned that a **stack** is an abstract data type that follows the **Last-In, First-Out (LIFO)** principle. In Python, it can be implemented using a simple list where `append()` works as push and `pop()` works as pop. I also saw how to check if the stack is empty, display its contents, remove multiple elements, and count its size. This helped me understand how stacks work in real programming and how useful they are in storing and organizing data.*

References

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