

Long Quiz Skill Test	
Course Code: 201L DSA	Program: BS CPE
Course Title: Data Structure Analysis	Date Performed: 30/08/25
Section: 2B	Date Submitted: 30/08/25
Name: Junichiro H. Uy	Instructor: Maria Rizette M. Sayo
1. Objectives	
<ul style="list-style-type: none"> • Make a stack data structure using Python. • Push, pop, and traverse each character of Full name starting from the last pushed character. • Insert underscore at the front of stack. 	
2. Discussion	
<p>The stacks are a collection of objects that are first in last out. Meaning if you inserted an object, that object will be the last to be used or pulled out. Then if we pop, the recent inserted object would be the one to be removed. Examples are a can of pringles, a narrow garage, and many more that enables the first in last out principle.</p> <p>Its main components are push() and pop(). Push allows us to insert an object on top of the first object, then pop allows us to remove the top object.</p>	
3. Materials and Equipment	
I used my own laptop, typed the code in google colab.	
4. Procedure	
<ol style="list-style-type: none"> 1. I made a class for stacks, then I added functions such as push, pop, and traverse in it. 2. I initialized the myName as Stack(), for that variable to take the structure of the class. 3. Then I pushed the characters in my name using the push function, then I popped the wrong character using the pop function. After all that, I used the traverse function in myName variable to print it from last to first. 4. I also added an underscore in the push function, so that it adds an underscore in the very front or bottom. 5. I tried adding a while loop to make insertions of characters more lightweight and less time demanding. 	
5. Output	

```

class Stacks:
    def __init__(self):
        self.stack = []

    def push(self, value): #Dito ko na nilagay yung pang insert ng underscore, so it's much cleaner
        if len(self.stack) == 0:
            self.stack.append("_") #Inserts underscore at the bottom
            self.stack.append(value)

    def pop(self):
        if len(self.stack) <= 0:
            print("There are no characters in this stack!")
        else:
            return self.stack.pop()

    def traverse(self):
        if len(self.stack) <= 0:
            print("There are no characters in this stack!")
        else:
            self.stack.reverse()
            for x in self.stack:
                print(x, end=" ")

```

Figure 1 Screenshot of program from colab

I made a class in here called Stacks.

```

myName = Stacks()

myName.push('J')
myName.push('C')
myName.pop()
myName.push('U')
myName.push('N')
myName.push('I')
myName.push('C')
myName.push('H')
myName.push('I')
myName.push('R')
myName.push('O')
myName.push('H')
myName.push('U')
myName.push('Y')

myName.traverse()

Y U H O R I H C I N U J _

```

Figure 2 Screenshot of program from colab

Then I initialized it in the variable then used its functions.

```
|
assistedArray = Stacks()
condition = True

while condition is True:
    question = int(input("\nPush(1) Exit(2): "))
    if question == 1:
        while True:
            print("\nType your letter(1:Pop 2:Exit): ")
            letter = input("")
            if letter == "1":
                assistedArray.pop()
            elif letter == "2":
                break
            for x, i in enumerate(letter):
                assistedArray.push(i)
        else:
            assistedArray.traverse()
            break
```

Figure 3 Screenshot of program from colab

I also tried to make a while loop where I can push a character easily, although this would've been better when we can use buttons instead of typing numbers to command your system, but google colab doesn't offer that much freedom for now.

```
Push(1) Exit(2): 1
Type your letter(1:Pop 2:Exit):
J
Type your letter(1:Pop 2:Exit):
U
Type your letter(1:Pop 2:Exit):
N
Type your letter(1:Pop 2:Exit):
I
Type your letter(1:Pop 2:Exit):
C
Type your letter(1:Pop 2:Exit):
H
Type your letter(1:Pop 2:Exit):
I
Type your letter(1:Pop 2:Exit):
R
Type your letter(1:Pop 2:Exit):
O
Type your letter(1:Pop 2:Exit):
H
Type your letter(1:Pop 2:Exit):
U
Type your letter(1:Pop 2:Exit):
Y
Type your letter(1:Pop 2:Exit):
2
Push(1) Exit(2): 2
Y U H O R I H C I N U J _
```

Figure 4 Screenshot of output from colab

6. Conclusion

Lab Activity Rubric 							
Criteria	Ratings						Pts
 SO 7 PI 1 Student Outcome 7.1 Acquire and apply new knowledge from outside sources. threshold: 4.8 pts	6 pts Excellent Educational interests and pursuits exist and flourish outside classroom requirements, knowledge and/or experiences are pursued independently and applies knowledge learned into practice	5 pts Good Educational interests and pursuits exist and flourish outside classroom requirements, knowledge and/or experiences are pursued independently	4 pts Satisfactory Look beyond classroom requirements, showing interest in pursuing knowledge independently	3 pts Unsatisfactory Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently	2 pts Poor Relies on classroom instruction only	1 pts Very Poor No initiative or interest in acquiring new knowledge	6 pts
 SO 7 PI 2 Student Outcome 7.2 Learn independently threshold: 4.8 pts	6 pts Excellent Completes an assigned task independently and practices continuous improvement	5 pts Good Completes an assigned task without supervision or guidance	4 pts Satisfactory Requires minimal guidance to complete an assigned task	3 pts Unsatisfactory Requires detailed or step-by-step instructions to complete a task	2 pts Poor Shows little interest to complete a task independently	1 pts Very Poor No interest to complete a task independently	6 pts
 SO 7 PI 3 Student Outcome 7.3 Critical thinking in the broadest context of technological change threshold: 4.8 pts	6 pts Excellent Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions	5 pts Good Evaluate information from a variety of sources; formulates a clear and precise perspective.	4 pts Satisfactory Analyze information from a variety of sources; formulates a clear and precise perspective.	3 pts Unsatisfactory Apply the gathered information to formulate the problem	2 pts Poor Gather and summarized the information from a variety of sources but failed to formulate the problem	1 pts Very Poor Gather information from a variety of sources	6 pts
 SO 7 PI 4 Student Outcome 7.4 Creativity and adaptability to new and emerging technologies threshold: 4.8 pts	6 pts Excellent Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.	5 pts Good Ideas are creative and adapt the new knowledge to solve a problem or address an issue	4 pts Satisfactory Ideas are creative in solving a problem, or address an issue	3 pts Unsatisfactory Shows some creative ways to solve the problem	2 pts Poor Shows initiative and attempt to develop creative ideas to solve the problem	1 pts Very Poor Ideas are copied or restated from the sources consulted	6 pts
Total Points: 24							

Overall, I had fun doing this activity as it taught me on how to manipulate a stack. If ever I will create an application or website that requires a stack array to be used, I can do this.