|  |  |
| --- | --- |
| **Long Quiz – Skill Test** | |
| **Course Code:** CPE - 201 | **Program:** Computer Engineering |
| **Course Title:** Data Structure and Algorithms | **Date Performed:** 8/30/2025 |
| **Section:** BSCpE – 2B | **Date Submitted:** 8/30/2025 |
| **Name:** Monoy, Justin Rhey A. | **Instructor:** Engr. Maria Rizette Sayo |
| 1. **Objectives** | |
| * Create a Python program that manipulates a full name by inserting underscores between the words using the stack * By the end of this activity, students will understand the use of .pop() and .push() methods in a stack data structure. | |
| **2. Discussion** | |
| A stack in Python is a linear data structure that adheres to the Last-In-First-Out (LIFO) principle. This means the last element added to the stack is the first one to be removed. To add a new element to end of the stack, use “.push()“, while pop()” is to remove the lastly added element in the stack | |
| **3. Materials and Equipment** | |
| 1. Python software (e.g. Google Colab, PyCharm and visual studio code)  * Use to create python program and displays the output.  1. Desktop computer/Laptop  * Use to run the program need for python, and for typing codes.  1. Operating system (e.g. Windows 10 or 11)  * Most python software requires the latest version of operating system to run. | |
| **4. Procedure** | |
| 1. Initialize the Stack: Create a class called Stack that holds a list (stack) to store items. The full\_name is also stored inside the class as a, which will later be manipulated. 2. Push Items: The push method adds items to the stack. Each time you call push, the data gets added to the end of the list. 3. Pop Items: The pop method removes the last item added to the stack. If the stack is empty, it returns "Stack is empty". Otherwise, it removes and returns the last element in the stack. 4. Check the Stack: The check method checks if the stack is empty. If it is, it prints "Stack is empty". 5. Traverse the Stack: The traverse method goes through each item in the stack and prints them one by one, showing what is currently inside the stack. 6. Display the Stack: The display method prints the entire stack in one line. 7. Underscore Procedure: The underscore method splits the full\_name into words, then pushes each word and an underscore (\_) onto the stack. After pushing all the words and underscores, it removes the last underscore to prevent an extra one at the end. 8. Pop an Item: After calling the underscore method to fill the stack with words and underscores, you can pop the last item from the stack. The program then asks the user to type 1 if they want to pop the last element again. If the user types 1, the item is popped. 9. Check and Display: Finally, the program checks if the stack is empty, and then it traverses and displays the current content of the stack, showing the remaining elements after popping. | |
| **5. Output** | |
|  | |
| **6. Conclusion** | |
| This skill test taught me how to insert underscores in my full name in specific words, helping me understand the practical applications of stack operations in Python. By using the ".push()" and ".pop()" methods, I was able to manipulate my full name and practice key concepts in stack data structures. Through this exercise, I gained a deeper understanding of how elements are added and removed in a stack, and how these operations can be used to perform tasks such as modifying strings efficiently. This experience has improved my problem-solving skills and solidified my knowledge of data structures in Python. | |

****