Activity No. 4	
Hands-on Activity 4.1 Stacks	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed: 10/05/2024**
Section:v CPE21S1	**Date Submitted: 10/05/2024**
Name(s): Jhon Hendricks T. Bautista	**Instructor: Mrs. Maria Rizette Sayo**
6. Output	

```
newStack.push(3);  // Adds 3 to the stack
newStack.push(8);
newStack.push(15);

// Check if the stack is empty
cout << "Stack Empty? " << newStack.empty() << endl;

// Print the size of the stack
cout << "Stack Size: " << newStack.size() << endl;

// Get and display the top element of the stack
cout << "Top Element of the Stack: " << newStack.top() <<
    endl;

// Pop the top element from the stack
newStack.pop();

// Display the new top element after popping
cout << "Top Element of the Stack: " << newStack.top() <<
    endl;

// Print the size of the stack after popping
cout << "Stack Size: " << newStack.size() << endl;

return 0;
```

```
/tmp/4Pg1uua5CA.o
Stack Empty? 0
Stack Size: 3
Top Element of the Stack: 15
Top Element of the Stack: 8
Stack Size: 2


=== Code Execution Successful ===
```

Table 4-1. Output of ILO A

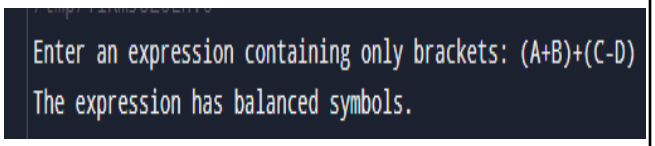
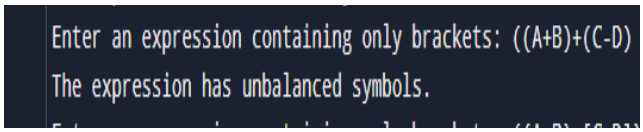
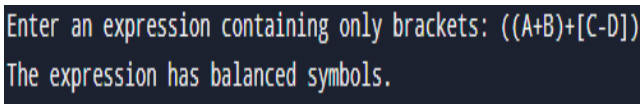
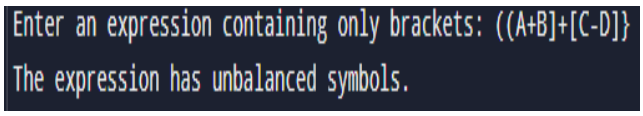
```
Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEMPTY, 5. DISPLAY
Enter your choice: 1
Enter New Value: 1
Value 1 pushed onto the stack.

Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEMPTY, 5. DISPLAY
Enter your choice: 1
Enter New Value: 2
Value 2 pushed onto the stack.

Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEMPTY, 5. DISPLAY
Enter your choice: 1
Enter New Value: 2
Value 2 pushed onto the stack.

Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEMPTY, 5. DISPLAY
Enter your choice: 5
Stack elements are: 2 2 1

Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEMPTY, 5. DISPLAY
Enter your choice:
```

Table 4-2. Output of ILO B

```
/tmp/HjrDnfJenW.o
After the first PUSH, top of stack is: Top of Stack: 1
After the second PUSH, top of stack is: Top of Stack: 5
After the first POP operation, top of stack is: Top of Stack: 1
After the second POP operation, top of stack is: Stack is Empty.
Stack Underflow.
```

Table 4-3. Output of ILO C

7. Supplementary Activity

Expression	Valid? (Y/N)	Output (Screenshot)	Analysis
(A+B)+(C-D)	Yes		In the sample expression it is a valid balanced expression because our function checks that there is a both valid opening and closing symbol which is ().
((A+B)+(C-D)	No		For this sample expression we have an unbalanced symbol because in the first used symbol (, the stack did not receive a corresponding symbol) for the closing. This resulted in a false value making it unbalanced.
((A+B)+[C-D])	Yes		In this sample, we had a result of balanced symbols. This is because even if we used parenthesis and brackets for the two statements, the condition was still fulfilled because the code detected a proper opening and closing of the symbols.
((A+B]+[C-D]}	No		In this sample we have a result of unbalanced symbols. This is because there is an incorrect combination of opening and closing symbols used. There is a mismatch of (,],} symbols for the opening and closing symbols.

8. Conclusion

After doing this activity, I was able to understand stacks in C++. I was able to implement stacks in different variations in array, linked list, and the standard library in C++. I was also able to learn the different methods I can use when dealing with stacks. I can conclude that stacks is an important structure for following LIFO and FILO principles in your program. It was efficient for checking the proper symbols used in an expression. I can use stacks for backtracking and many more other problems.

9. Assessment Rubric