

Activity No. <4>	
<Hands-on Activity 4.1 Stacks>	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: 04/10/2024
Section:CPE21S4	Date Submitted: 04/10/2024
Name(s): Kenn Jie Valleser	Instructor: Engr. Maria Rizette Sayo

6. Output

```
1 //Tests the push, empty, size, pop, and top methods of the stack library.
2 #include <iostream>
3 #include <stack> // Calling Stack from the STL
4 using namespace std;
5 int main() {
6     stack<int> newStack;
7     newStack.push(3); //Adds 3 to the stack
8     newStack.push(8);
9     newStack.push(15);
10    // returns a boolean response depending on if the stack is empty or not
11    cout << "Stack Empty? " << newStack.empty() << endl;
12    // returns the size of the stack itself
13    cout << "Stack Size: " << newStack.size() << endl;
14    // returns the topmost element of the stack
15    cout << "Top Element of the Stack: " << newStack.top() << endl;
16    // removes the topmost element of the stack
17    newStack.pop();
18    cout << "Top Element of the Stack: " << newStack.top() << endl;
19    cout << "Stack Size: " << newStack.size() << endl;
20    return 0;
21 }
```

```
Stack Empty? 0
Stack Size: 3
Top Element of the Stack: 15
Top Element of the Stack: 8
Stack Size: 2

...Program finished with exit code 0
Press ENTER to exit console.
```

Observation: at Stack Empty?, asking if the stack is empty and the output 0 = false.
at Stack size:, it returns the number of elements inside the stack which is 3
at top element of the stack is showing that the top is 15
after popping the top element it showed the top is 8
and then it shows that after popping the stack size is now 2

Table 4-1. Output of ILO A

```
#include<iostream>
const size_t maxCap= 100;
int stack[maxCap]; //stack with max of 100 elements
```

```

int top = -1, i, newData;
void push();
void pop();
void Top();
bool isEmpty();
void Display();
int main(){
int choice;
std::cout << "Enter number of max elements for new stack: ";
std::cin >> i;
while(true){
std::cout << "Stack Operations: " << std::endl;
std::cout << "1. PUSH, 2. POP, 3. TOP, 4. isEmpty, 5. Display the elements" << std::endl;
std::cin >> choice;
switch(choice){
case 1: push();
break;
case 2: pop();
break;
case 3: Top();
break;
case 4: std::cout << isEmpty() << std::endl;
break;
case 5: Display();
break;
default: std::cout << "Invalid Choice." << std::endl;
break;
}
}
return 0;
}
bool isEmpty(){
if(top== -1) return true;
return false;
}
void push(){
//check if full -> if yes, return error
if(top == i-1){
std::cout << "Stack Overflow." << std::endl;
return;
}
std::cout << "New Value: " << std::endl;
std::cin >> newData;
stack[++top] = newData;
}
void pop(){

```

```
//check if empty -> if yes, return error
if(isEmpty()){
std::cout << "Stack Underflow." << std::endl;
return;
}
//display the top value
std::cout << "Popping: " << stack[top];
//decrement top value from stack
top--;
}
void Top(){
if(isEmpty()) {
std::cout << "Stack is Empty." << std::endl;
return;
}
std::cout << "The element on the top of the stack is " << stack[top] <<
std::endl;
}
void Display(){
    if(isEmpty()) {
        std::cout << "Stack is Empty." << std::endl;
        return;
    }
    std::cout << "Stack elements: ";
    for(int j = top; j >= 0; j--) {
        std::cout << stack[j] << " ";
    }
    std::cout << std::endl;
}
}
```

```

Enter number of max elements for new stack: 2
Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEmpty, 5. Display the elements
1
New Value:
10
Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEmpty, 5. Display the elements
1
New Value:
11
Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEmpty, 5. Display the elements
5
Stack elements: 11 10
Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEmpty, 5. Display the elements
4
0
Stack Operations:
1. PUSH, 2. POP, 3. TOP, 4. isEmpty, 5. Display the elements

```

Descriptions

case switch was used to make the user input an number to operate 5 functions which are push, pop, top, isempty and display the elements. at case 5 (display) the function is checking if the stack is empty and if the answer is no then the else statement was operated which a for loop for each element in the stack.

Table 4-2. Output of ILO B.1.

```

#include<iostream>
class Node {
public:
    int data;
    Node *next;
};
Node *head=NULL,*tail=NULL;
void push(int newData) {
    Node *newNode = new Node;
    newNode->data = newData;
    newNode->next = head;
    if(head==NULL) {
        head = tail = newNode;
    } else {

```

```

        newNode->next = head;
        head = newNode;
    }
}
int pop() {
    int tempVal;
    Node *temp;

    if(head == NULL) {
        head = tail = NULL;
        std::cout << "Stack Underflow." << std::endl;
        return -1;
    } else {
        temp = head;
        tempVal = temp->data;
        head = head->next;
        delete(temp);
        return tempVal;
    }
}
void Top() {
    if(head==NULL) {
        std::cout << "Stack is Empty." << std::endl;
        return;
    } else {
        std::cout << "Top of Stack: " << head->data << std::endl;
    }
}
void Display() {
    if (head == NULL) {
        std::cout << "Stack is Empty." << std::endl;
        return;
    }
    Node *temp = head;
    std::cout << "Stack elements are: ";
    while (temp != NULL) {
        std::cout << temp->data << " ";
        temp = temp->next;
    }
    std::cout << std::endl;
}
int main() {
    push(1);
    std::cout<<"After the first PUSH top of stack is :";
    Top();
    push(5);
    std::cout<<"After the second PUSH top of stack is :";
    Top();
    pop();
    std::cout<<"After the first POP operation, top of stack is:";
    Top();
}

```

```

pop();
std::cout<<"After the second POP operation, top of stack :";
Top();
pop();
std::cout<<"The elements inside the stack:";
Display();
return 0;
}

```

```

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 :Stack is Empty.
Stack Underflow.
The elements inside the stack:Stack is Empty.

...Program finished with exit code 0
Press ENTER to exit console.

```

Descriptions

after the first push which is 1 then it will display the element 1 in function Push() the new value is initialized a new node and new head for its value.

then checking with the function Top() is checking if the top of the stack is empty if not the then top element will display the output shows stack underflow because the code used pop when the stack is empty

and then the Display() function is used when the front of the stack is not NULL then the front stack is display however in the code the stack is empty.

Table 4-3. Output of ILO B.2.

7. Supplementary Activity

Expression	Valid?(Y/N)	Output(Console Screenshot)	Analysis
(A+B)+(C-D)	Y	Expression: (A+B)+(C-D) Valid	The expression has balanced symbols. The opening delimiters (and are matched with their corresponding closing delimiters).
((A+B)+(C-D)	N	Expression: ((A+B)+(C-D) Error: Unbalanced expression. Invalid	The expression is missing a closing delimiter) at the end. The stack is not empty at the end of the input,

			indicating an unbalanced expression.
((A+B)+[C-D])	Y	Expression: ((A+B)+[C-D]) Valid	The expression has balanced symbols. The opening delimiters (and [are matched with their corresponding closing delimiters) and].
((A+B)+[C-D])	N	Expression: ((A+B)+[C-D]) Error: Mismatched delimiters. Invalid	The expression has mismatched delimiters. The opening delimiter [is matched with the closing delimiter), which is incorrect. The correct matching should be [with].

8. Conclusion

- Summary of lessons learned
I learned how to use a for loop to display the elements in a stack and how to create a stack.
- Analysis of the procedure
The procedure was easy to understand and it implemented the operator in easy way
- Analysis of the supplementary activity
The activity problem is hard to create and new keywords were used and is difficult to understand.
- Concluding statement / Feedback: How well did you think you did in this activity? What are your areas for improvement?
I hope in the future I'm able to recreate the supplementary activity and not be confused on what to use.

9. Assessment Rubric