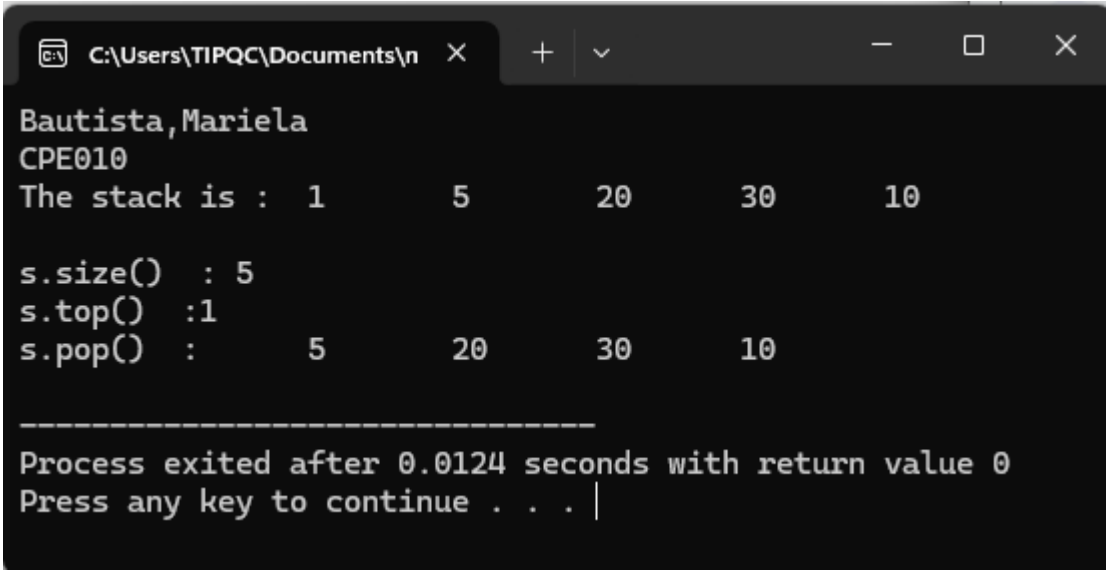


Seatwork 4.1	
Stacks	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: Aug 12, 2025
Section: CPE21S4	Date Submitted: Aug 12, 2025
Name(s): Bautista,Mariela S.	Instructor: Engr. Quejado

6. Output

The Output:



```
C:\Users\TIPQC\Documents\n  X + - □ X
Bautista,Mariela
CPE010
The stack is : 1      5      20      30      10

s.size()  : 5
s.top()   :1
s.pop()   :      5      20      30      10

-----
Process exited after 0.0124 seconds with return value 0
Press any key to continue . . . |
```

7. Supplementary Activity

Instructions: Create a code that demonstrates the basic functions of stack ADT. You may use the C++ STL in this activity. Submissions must include a pdf file containing the screenshot of the source code and the output. Use the activity template to submit your work.

mbautista.cpp

```
1 //CPP program to demonstrate working of STL stack
2 //Bautista,Mariela
3 //CPE2154
4
5 #include <iostream>
6 #include <stack>
7 using namespace std;
8
9 void showstack(stack <int> s)
10 {
11     while (!s.empty())
12     {
13         cout << '\t' << s.top();
14         s.pop();
15     }
16     cout << '\n';
17 }
18
19 int main ()
20 {
21     stack <int> s;
22     s.push(10);
23     s.push(30);
24     s.push(20);
25     s.push(5);
26     s.push(1);
27
28     cout << "Bautista,Mariela \n" ;
29     cout << "CPE010\n " ;
30
31     cout << "The stack is : ";
32     showstack(s);
33
34     cout << "\ns.size() : " << s.size();
35     cout << "\ns.top () : " << s.top();
36     cout << "\ns.pop () : ";
37
38
39     s.pop();
40     showstack(s);
41
42     return 0;
43
44
45
46 }
```

Compile Log Debug Find Results Close

Compilation results...

```
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\TIPQC\Documents\mbautista.exe
- Output Size: 1.81035995483398 MiB
- Compilation Time: 0.41s
```

8. Conclusion

What I've learned so far in this lesson is how a stack works in C++ using the STL Library. I have learned that the stack follows this rule "that the last to go in, it's the first to go out" its where the last item added is the first one removed. By using functions like push(), pop(), top(), IsEmpty(), IsFull(); we can add, remove, and check elements in the stack and to check the current value of the stack without changing anything. The code also helped me more, to understand how each stack changes after each operation.

9. Assessment Rubric