

Activity No. 4.1

Seatwork 4.1 : Stacks

Course Code: CPE010

Program: Computer Engineering

Course Title: Data Structures and Algorithms

Date Performed: 8/12/25

Section:CPE21S4

Date Submitted: 8/12/25

Name(s): Crishen Luper S. Pulgado

Instructor: Jimlord Quejado

6. Output

The screenshot shows a terminal window titled 'C:\Users\TIPQC\Downloads\stl' with the following output:

```
The stack is: 1 5 20 30 10
s.size() : 5
s.top() : 1
s.pop() :      5      20      30      10
-----
Process exited after 0.006951 seconds with return value 0
Press any key to continue . . . |
```

On the left, there is a code editor window showing the source code 'stack.cpp' with line numbers from 1 to 33. The code defines a stack and demonstrates its usage with push and pop operations.

```
[*] stack.cpp
1 #include <iostream>
2 #include <stack>
3
4 void showstack(stack <int> s)
5 {
6     while (!s.empty())
7     {
8         std::cout << '\t' << s.top();
9         s.pop();
10    }
11    std::cout << '\n';
12 }
13
14 int main ()
15 {
16     stack <int> s;
17     s.push(10);
18     s.push(30);
19     s.push(20);
20     s.push(5);
21     s.push(1);
22
23     std::cout << "The stack is: ";
24     showstack(s);
25
26     std::cout << "\ns.size() : " << s.size();
27     std::cout << "\ns.top() : " << s.top();
28
29     cout << "\ns.pop() : ";
30     s.pop();
31     showstack(s);
32     return 0;
33 }
```

8. Conclusion

This activity has taught me how to create a stack in c++. It is used for storing and managing elements in a specific order. It follows the principle of first-in first-out.