5️⃣

Stack

A stack is a non-primitive linear data structure. it is an ordered list in which the addition of a new data item and deletion of the already existing data item is done from only one end known as the top of the stack (TOS). The element which is added in last will be first to be removed and the element which is inserted first will be removed in last. As all the deletion and insertion in a stack are don

e from the top of the stack, the last added element will be the first to be removed from the stack. That is the reason why stack is also called **Last-in-First-out (LIFO).**

Stack in STL.

Syntax:

```
stack<object_type> variable_name;
```

Example:

```
stack<int> s;
stack<string> s;
```

Functions in Stack:

push() – to insert an element in the stack.

```
stack<int> s;
s.push(110);
s.push(220);
```

pop() – deletes the last element of the stack.

```
s.pop();
```

top() – returns the element at the top of the stack.

```
s.top();
```

emplace() – to insert an element in the stack.

```
stack<int> s;
s.emplace(1);
s.emplace(2);
```

size() – returns the number of elements on the stack.

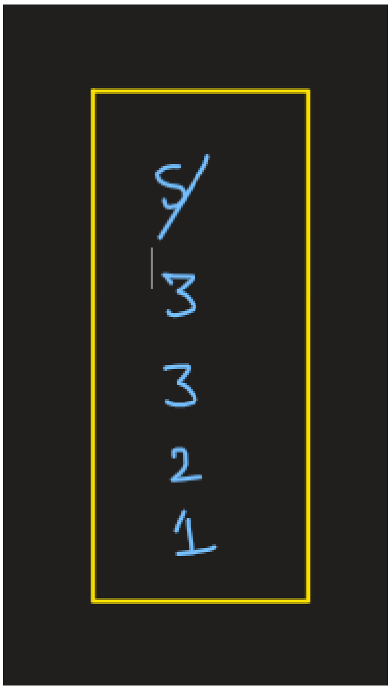
```
s.size();
```

empty() – to check if the stack is empty or not.

```
s.empty();
```

The Whole Code:

```
// Containers -- Stack (Last in First Out (LIFO))
#include <bits/stdc++.h>
using namespace std;
void explainStack()
{
      stack<int> st;
      st.push(1); //{1}
      st.push(2); // {2, 1}
      st.push(3); // {3,2,1}
      st.push(3); // {3,3,2,1}
      st.emplace(5); // {5,3,3,2,1}

      cout << st.top(); // print 5 ** st[2] is invalid**

      st.pop(); // st look like {3,3,2,1}

      cout << st.top(); //3
      cout << st.size(); //4
      cout << st.empty(); // false

      stack<int> str1, str2;
      str1.swap(str2); // swap str1 and str2
}

int main()
{
      explainStack();
      return 0;
}
```



Striver’s code

```
#include<bits/stdc++.h>
using namespace std;
void printstack(stack<int> s1)
{
    stack<int> s2=s1;
    while(!s2.empty())
    {
        cout<<s2.top()<<"\\n";
        s2.pop();
    }
}
int main()
{
    stack<int> s;
    for(int i=1;i<=5;i++)
    s.push(i);

    cout<<"The elements of the stack are:"<<endl;
    printstack(s);

    cout<<"The size of the stack: "<<s.size()<<endl;
    cout<<"The top element of the queue: "<<s.top()<<endl;
    cout<<"Pop the top element: "<<endl;
    s.pop();
    printstack(s);
}

Output:
```

```
The elements of the stack are:
5
4
3
2
1
The size of the stack: 5
The top element of the queue: 5
Pop the top element:
4
3
2
1
```