

Stack

Idea

- Like a stack of books!
 - Make larger: add to **top**
 - Make smaller: remove from **top**



Idea

- Like a stack of books!
 - Make larger: add to **top**
 - Make smaller: remove from **top**

Don't remove this
book first!
;-)



In C++

- In C++: `std::stack<int>`
- Requires: `#include <stack>`

In C++

- In C++: `std::stack<int>`
- Requires: `#include <stack>`

- Commands:
 - `push(val)` Add `val` to top of stack.
 - `pop()` Remove value at top.
 - `top()` Get value at top (without removing).

Example

Example Program

```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```

Example Program

```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```

my_stack

Example Program

```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```

1

my_stack

Example Program

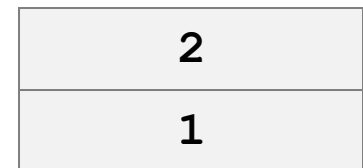
```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```



my_stack

Example Program

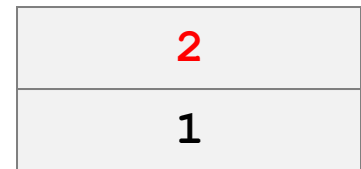
```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```



my_stack

Example Program

```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```

3
2
1

my_stack

Example Program

```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```

3
2
1

my_stack

Example Program

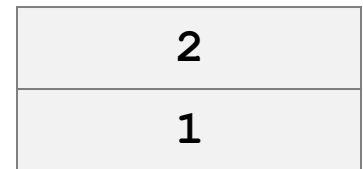
```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```



my_stack

Example Program

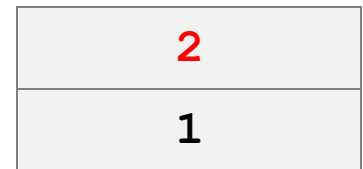
```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```



my_stack

Example Program

```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```

4
2
1

my_stack

Example Program

```
std::stack<int> my_stack; // empty stack

my_stack.push(1);
my_stack.push(2);
std::cout << my_stack.top() << " "; // 2

my_stack.push(3);
std::cout << my_stack.top() << " "; // 3

my_stack.pop();
std::cout << my_stack.top() << " "; // 2

my_stack.push(4);
std::cout << my_stack.top() << " "; // 4
```

4
2
1

my_stack