Session 7 / Part 1

Stack

Agenda:

1. Implementing Stack header file
2. Discussing Makefile

1) Implementing Stack header file

Before getting into the code, we need to review some concepts.

It is very easy to get lost in details, and forget the important facts.

In the programming world, we have felt a need for a storage that has LIFO property. Now, with Object oriented mindset, we create a header file, a plan, for creating it.

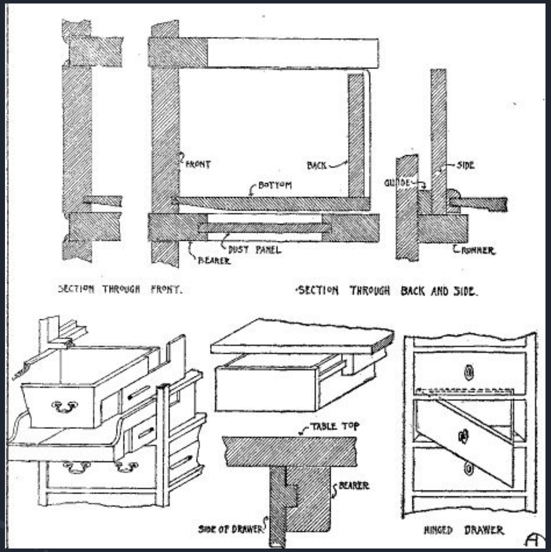
Once the plan or the architecture is ready (the header file), we implement that in .cc file.

Finally, we can create one instance, one singular object, from that design (class Stack), and Start working with it.

1) Implementing Stack header file (one simple analogy)

A need for a storage of clothes

Planning Ready on paper



Woodworking to create a physical drawer

A functional object called "drawer" Ready for use

In the programming world

A need for a storage of data

(let's call it stack)

Designing the Big picture

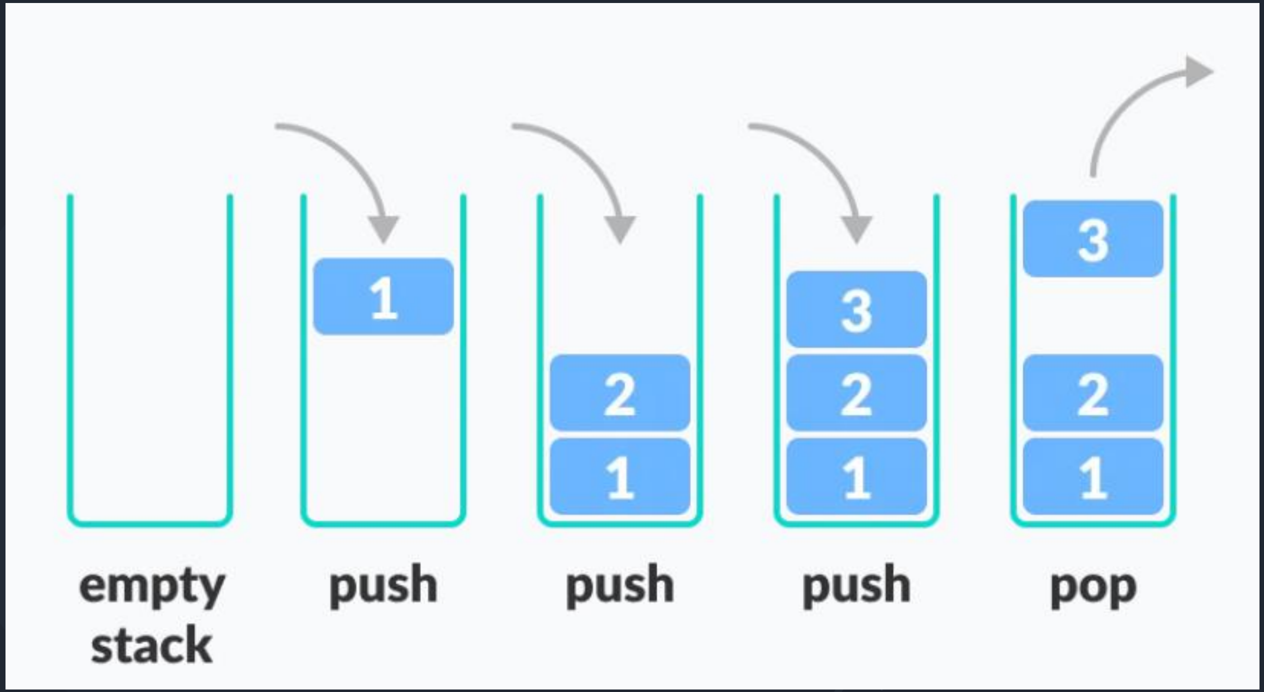
Header file

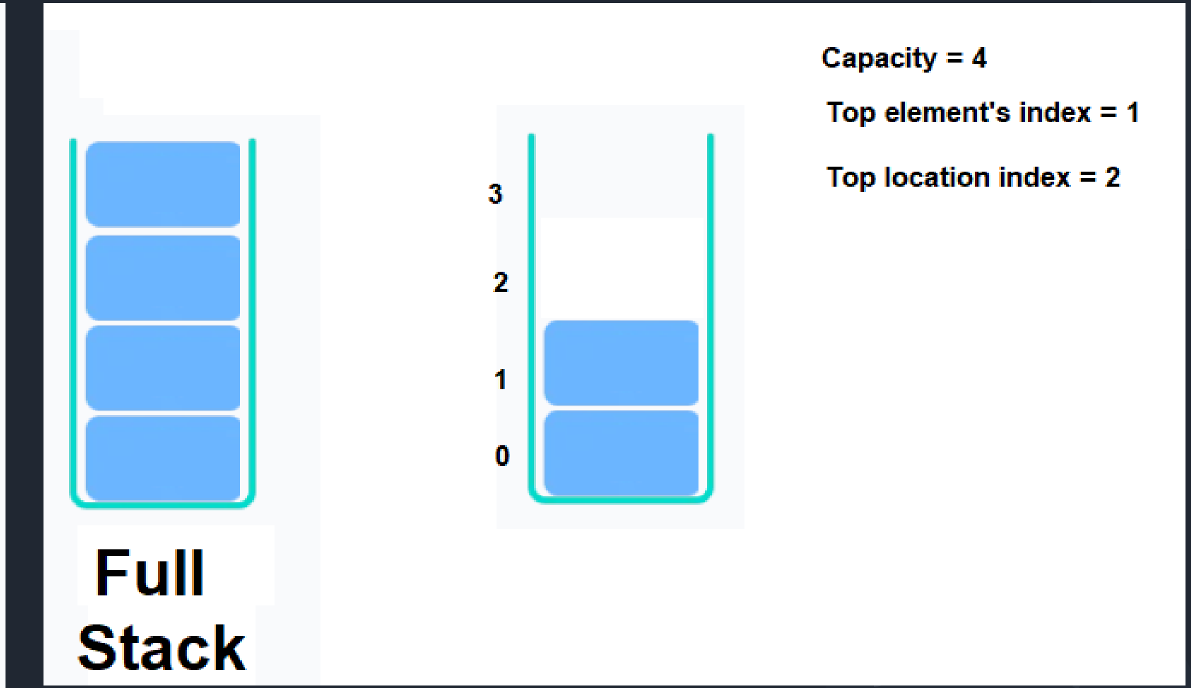
Adding details To the big picture

.cc file

Now, in your program, you can write " myStack = Stack()" to create one functional object and start using it

1) Implementing Stack header file





1) Implementing Stack header file

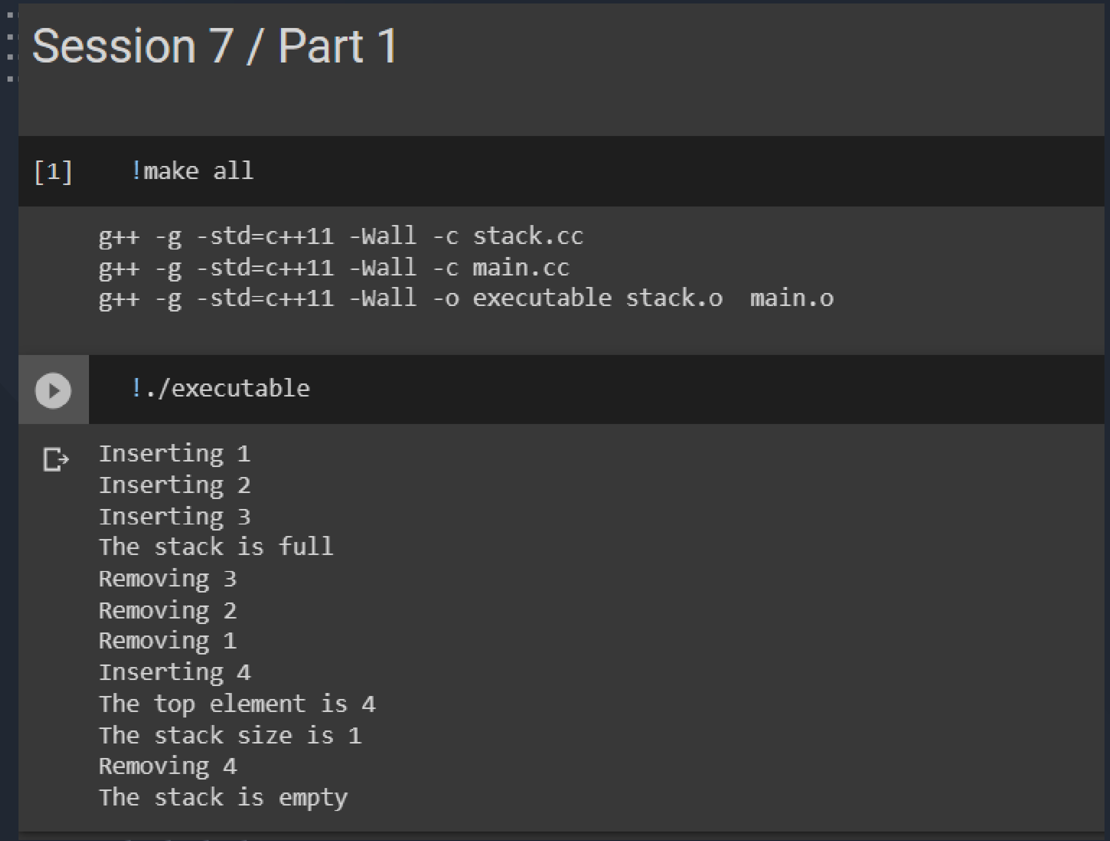
Now that you know the concept of Stack, open the stack.cc file and implement:

+ Constructor()

+ Pop()

+ isEmpty()

Expected output



Session 7 / Part 1

```
!make all
```

```
g++ -g -std=c++11 -Wall -c stack.cc
g++ -g -std=c++11 -Wall -c main.cc
g++ -g -std=c++11 -Wall -o executable stack.o  main.o
```

```
!./executable
```

```
Inserting 1
Inserting 2
Inserting 3
The stack is full
Removing 3
Removing 2
Removing 1
Inserting 4
The top element is 4
The stack size is 1
Removing 4
The stack is empty
```