Chapter 6: Stacks

Concepts

Stacks and Queues can be implemented with a linked list or with an array. The important thing about these structures is *how* they manage the data within them – they restrict access to data except for what's on top (for Stacks), or what is in front (for Queues).

Stacks and Queues are relatively simple data structures, functionality-wise. Their functions include:

Stack	**Last-in-first-out**	**Queue**	**First-in-first-out**
size	get the size of the stack	size	get the size of the queue
top	get the top-most item	front	get the front-most item
push	add a new item to the top	push	add a new item to the back
pop	remove the item on top	pop	remove the item in front

Think of a Stack like a Pringles can, where you can only access the top-most chip (crisp?). If you have an empty Pringles can, the chip that's added first is on the bottom. If you add more chips on top, you cannot access the bottom-most chip until all the chips above it are gone.

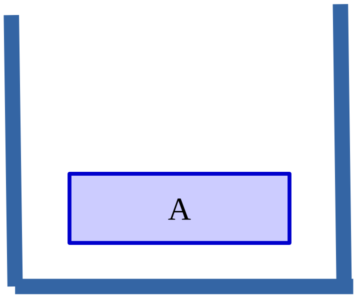
A Queue should be pretty self-explanatory; just like lining up at Micro Center, the first person in line is the first person who gets to leave. As more people enter the line, they line up at the *end* – or the back – of the queue.

Stack functionality

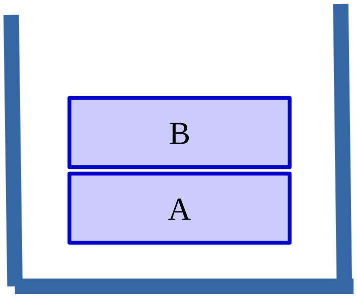
Push

As items are added to the stack, you begin filling it from bottom-to-top.

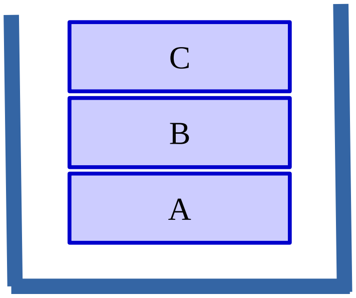
First, Push “A”



Second, Push “B”



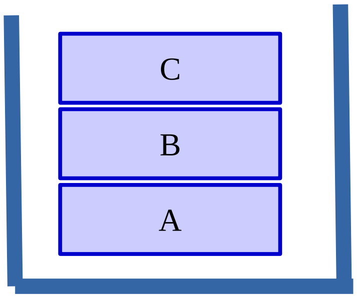
Third, Push “C”



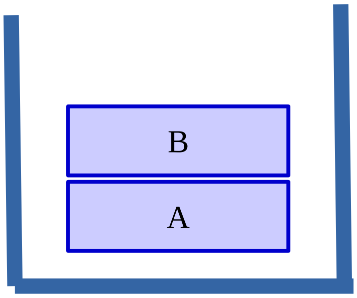
Pop

Pop removes the top-most item.

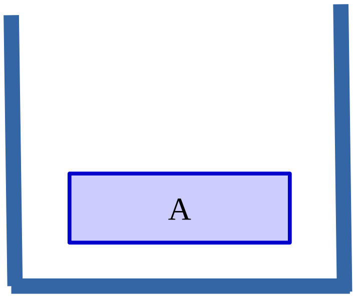
Stack with items in it



Pop removes “C”



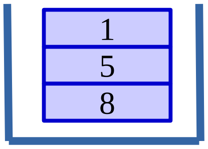
Pop removes “B”



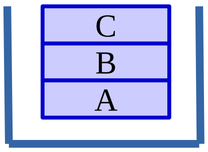
Top

Top returns the data in the top-most item.

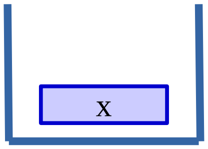
Top returns 1



Top returns “C”



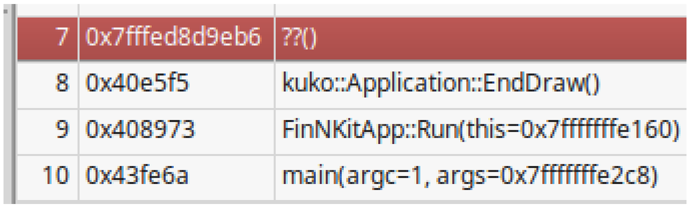
Top returns “x”



Uses of a Stack

Function calls

A stack can be used in various ways in computer science. Remember the Call Stack debugging tool? It will show you the list of functions called up until your current point.

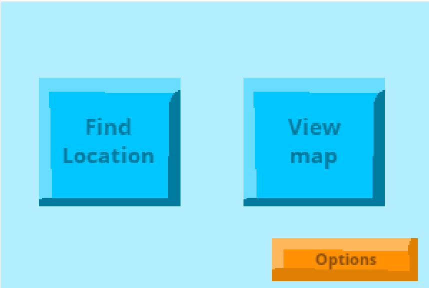


7	0x7fffed8d9eb6	??()
8	0x40e5f5	kuko::Application::EndDraw()
9	0x408973	FinNKitApp::Run(this=0x7fffffffe160)
10	0x43fe6a	main(argc=1, args=0x7fffffffe2c8)

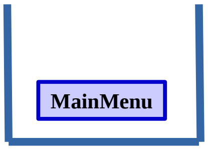
Whenever a function is called, it is pushed onto the stack. To return from the function and go to the last function, you Pop the current function off the stack to easily return to the previous function.

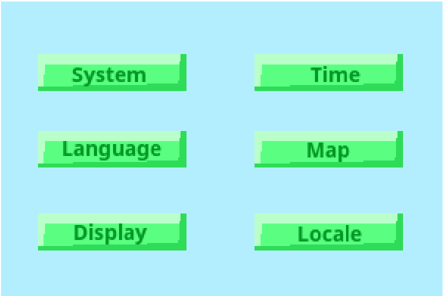
Screen states

Different screens in a program are sometimes designed to be pushed and popped on stacks. For example, let's say you're using your GPS.

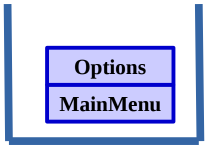


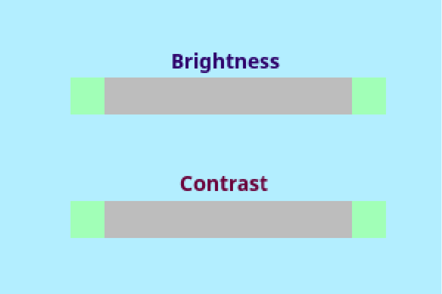
When it boots up, it starts at the main menu. This gets pushed on the stack.





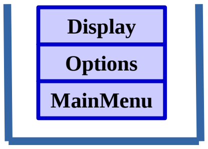
You click on "Options". The Options menu gets pushed onto the stack next.

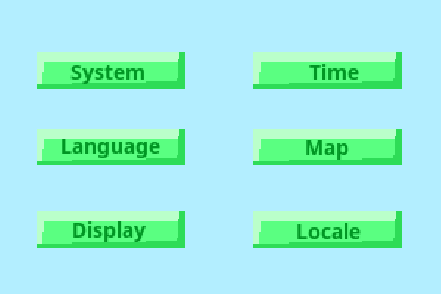




You click on “Display”, and the Display menu gets pushed onto the stack.

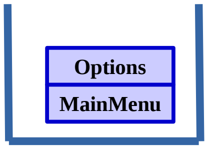
Once you’ve adjusted the display, you hit the Back button. This pops Display off the top.

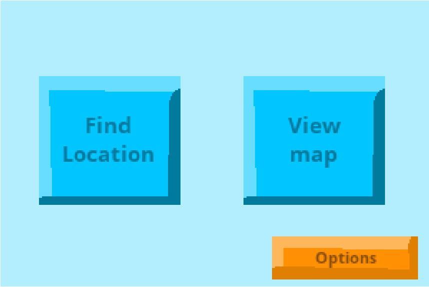




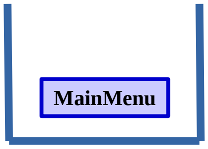
Since Display was popped, the top-most menu is now Options again.

You hit Back again, and it pops Options off the top.





Options was removed, so the next menu is the main menu.



Syntax checking – verifying braces

String recognition

Postfix expressions

(Work in progress)