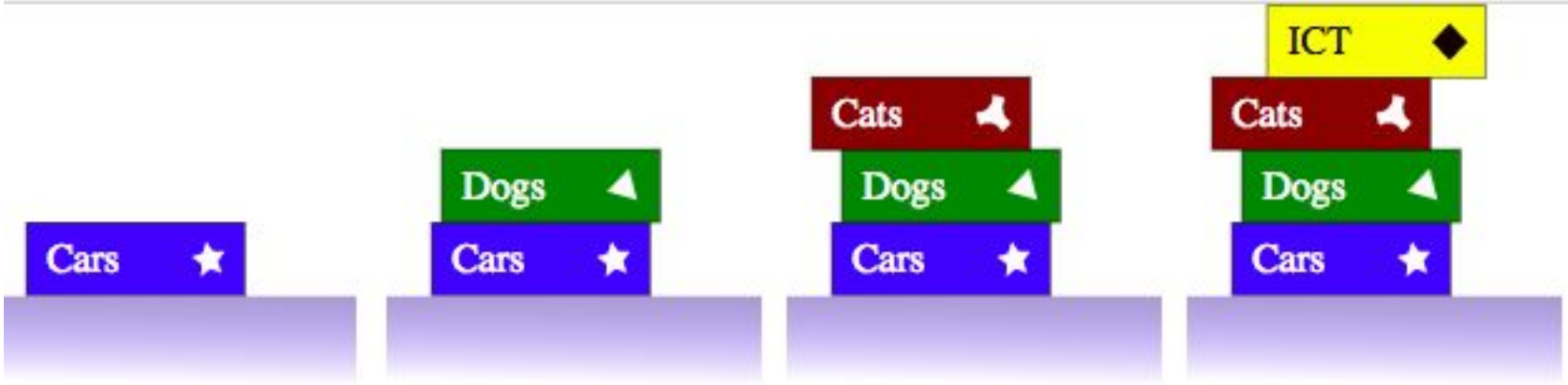
Stacks

describe the functioning of the stack and queue data types correctly using the terms ‘last in last out’ and ‘first in first out’

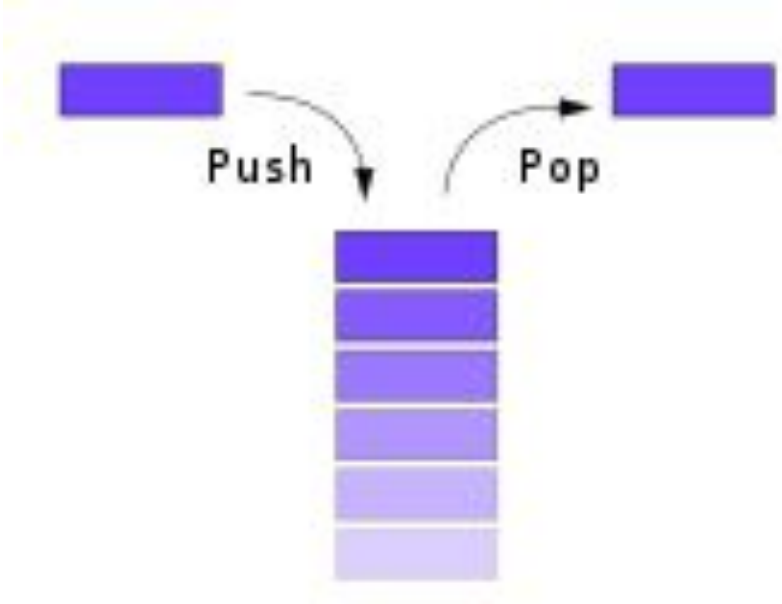
A stack is an ADT that might involve a dynamic or static implementation. A stack is a last-in-first-out (LIFO) or first-in-last-out (FILO) ADT. Implementations should include two operations, pushing and popping, and a pointer to the top of the stack.

A real life example is a stack of books you might have on your desk:



CPT-ADT-stacks-real-life.svg
In this example we keep adding (pushing) books to the stack. If we want to get at the bottom book about Cars, we must first remove (pop) all the books above it. Hence First In Last Out (FILO)

Let's take a look at a computer implementation of a stack:



Pushing: Adds a new specified item to the top of the stack

Popping: Removes the item from the top of the stack.

State 1

Push 'elephant'

memloc	data	TopOfStack
4		
3		
2		
1	elephant	<--

State 2

Push 'hippo'

memloc	data	TopOfStack
4		
3		
2	hippo	<--
1	elephant	

State 3

Push 'zebra'

memloc	data	TopOfStack
4		
3	zebra	<--
2	hippo	
1	elephant	

State 4

Pop

memloc	data	TopOfStack
4		
3	zebra	
2	hippo	<--
1	elephant	

Note the data isn't 'deleted'
The TopOfStack pointer moves
to show it is no longer in
the stack

State 5

Push 'flamingo'

memloc	data	TopOfStack
4		
3	flamingo	<--
2	hippo	
1	elephant	

Stacks have several uses:

- Reversing queues (as seen above with the Alphabetised names)
- Performing Reverse Polish Calculations (see ....)
- Holding return addresses and syste states for recursive function calls

Exercise: Stacks

Draw the stack after each of the following commands, starting with an empty stack. What does the stack achieve:

Push 'Annabelle'
Push 'Chris'
Push 'Hemingway'
Push 'James'
Pop
Pop
Pop
Pop

Exercise: Stacks

Give the set of instructions required to get from State 1 to State 2 as shown below:

State 1

memloc	data	TopOfStack
4		
3	Sharks	<--
2	Chalk	
1	Cheese	

State 2

memloc	data	TopOfStack
4		
3	Witches	<--
2	Sand	
1	Cheese	

Exercise: Stacks

Give the set of instructions required to get from State 1 to State 2 as shown below:

State 1

memloc	data	TopOfStack
4		
3	Sharks	<--
2	Chalk	
1	Cheese	

State 2

memloc	data	TopOfStack
4		
3	Witches	<--
2	Sand	
1	Cheese	

Answer:
Pop
Pop
Push 'Sand'
Push 'Witches'

Exercise: Stacks

https://en.wikibooks.org/wiki/A-level_Computing/AQA/Paper_1/Fundamentals_of_data_structures/Stacks

information from the site:

- https://en.wikibooks.org/wiki/A-level_Computing/AQA/Paper_1/Fundamentals_of_data_structures/Stacks

- https://upload.wikimedia.org/wikipedia/commons/thumb/f/f5/CPT-ADT-stacks-real-life.svg/632px-CPT-ADT-stacks-real-life.svg.png