What is a Stack?

• A stack is a data structure, we mean something that is meant to hold data and provides certain operations on that data.

• An array is an example of simple data structure. Its ‘operations’ are using offsets with brackets.

  • ex. someArray[2];

• “Hold data and provide certain operations”. Does this sound familiar?
What is a Stack?

- A stack is a bit more sophisticated than an array.
- The two important things are:
  - Its primary operations are “push” and “pop”.
  - Its LIFO (last-in, first-out).
- New items can be added to the top of the stack.
- Items can be removed only from the top of the stack.
What is a Stack?

• Imagine a stack of plates…
  • You can “push” a new plate onto the stack of plates by sticking a plate on top
  • You can “pop” a plate off the stack by taking the top one
Motivation

• There are many data structures. Trees, Graphs, Sets and so on.

• In fact, if you continue in CS you will take a whole course on them.

• Why are there so many?

• They are generalized solutions to common problem types in programming.

• Each data structure has characteristics that make it suited to certain types of problems.
Motivation

• A stack is suited to problems where that have some element of LIFO behavior.

• We saw this in the call stack memory region.

• Another example, suppose we want to reverse a String?

• We could push each character onto a stack, then pop them all off, creating a new string.

• We would need a stack of characters.
Operations

• **push(value: char): void**
  
  • Puts the character value at the top of the stack.

• **pop(): char**
  
  • Removes the character from the top of the stack and returns it. If stack is empty, returns null.

• **peek()**
  
  • Returns the character from the top of the stack (without removing it). If stack is empty, returns null.

• **empty(): boolean**
  
  • Returns true if the stack is empty, otherwise it returns false.

• **getSize(): int**
  
  • Returns the number of elements in the stack.

• **getCapacity(): int**
  
  • Returns the maximum number of characters allowed in the stack.
Let’s look at some code