CSE 3010 – Data Structures & Algorithms

Lecture #7

What will be covered today

- Algebraic specifications to define ADTs
- Introduction to LIFO (stack) structure
- Implementation of stack operations
- Uses of stack

Algebraic specification of an ADT – Types and Operations

Types:

`ADT, ITEM, BOOLEAN`

where `ITEM` is of some type the `ADT` will contain

`BOOLEAN` is either `True` or `False`

Operations:

```
create:                 -> ADT
push: ADT x ITEM        -> ADT
pop: ADT                -> ADT
top: ADT                -> ITEM
isEmpty: ADT            -> BOOLEAN
```

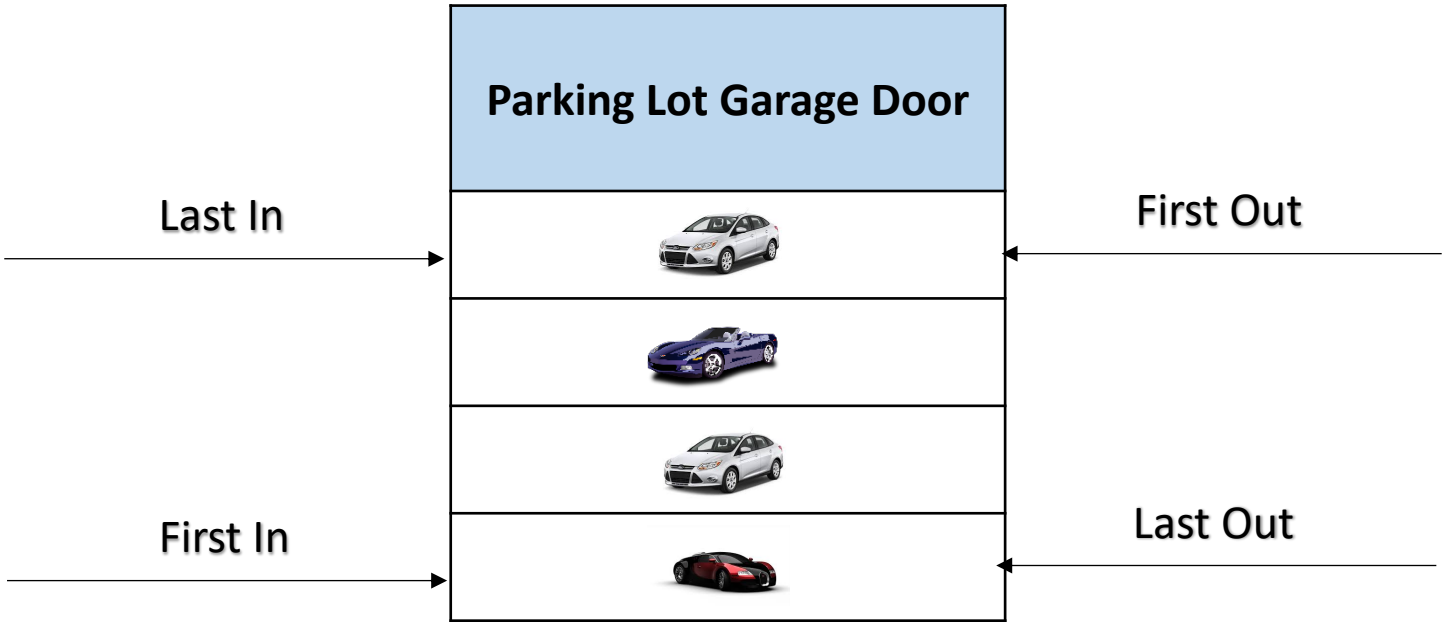
Algebraic specification of the ADT - Axioms

For all S ∈ ADT, **and** k ∈ ITEM

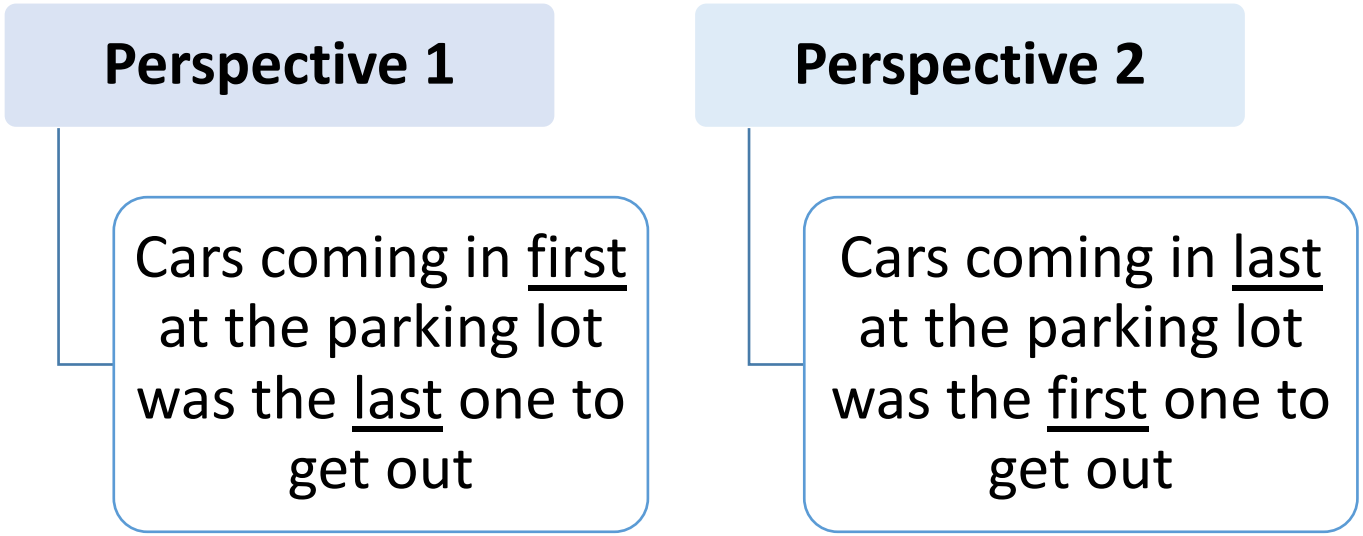
```
pop(create()) = error
pop(push(S,k)) = S

top(create())  = error
top(push(S,k)) = k

isEmpty(create()) = true
isEmpty(push(S,k)) = false
```

Let us visualize

Assume a parking lot in a mall has only one opening for entry and exit



What did we notice through our visualization



Stack is an abstract data type

- Collection of elements
- Element can be added (push)
- Last element added can be removed (pop)

Algebraic specification of Stack ADT – Types and Operations

Types:

`STACK, ITEM, BOOLEAN`

where `ITEM` is of some type the `STACK` will contain

`BOOLEAN` is either `True` or `False`

Operations:

```
create:                 -> STACK
push: STACK x ITEM      -> STACK
pop: STACK              -> STACK
top: STACK              -> ITEM
isEmpty: STACK          -> BOOLEAN
```

Implementation of Stack ADT

- ADTs are realized using data structures
- Stack is a container of elements with last-in first-out (LIFO) data structure
  - What does that mean?
  - How can it be implemented?

More examples

Example 1 : Reverse and print a string

Example 2: Plates waiting to be washed in a restaurant kitchen

Example 3: Checking for matching braces by a compiler

Example 4: Evaluation of an arithmetic expression

Example 5: Function calls in a programming language