

# Queue (Data Structure)

Prof. Wade Fagen-Ulmschneider

I ILLINOIS

A queue is a first-in first-out data structure that is similar to waiting in a line or “queue”:



# Abstract Data Type

In data structures, we will always begin our analysis asking “how can this structure be used with data?”

- A structure’s **Abstract Data Type (ADT)** is how data interacts with the structure.
  - An ADT is not an implementation, it is a description.

# Queue ADT

**create** → Creates an empty queue

**push** → Adds data to the back of the queue

**pop** → Removes data from the front of the queue

**empty** → Returns true if the queue is empty

# Example:

Consider four operations:

- Create a queue of integers.
- Push the number 4 to the queue.
- Push the number 2 to the queue.
- Pop a number from the queue.

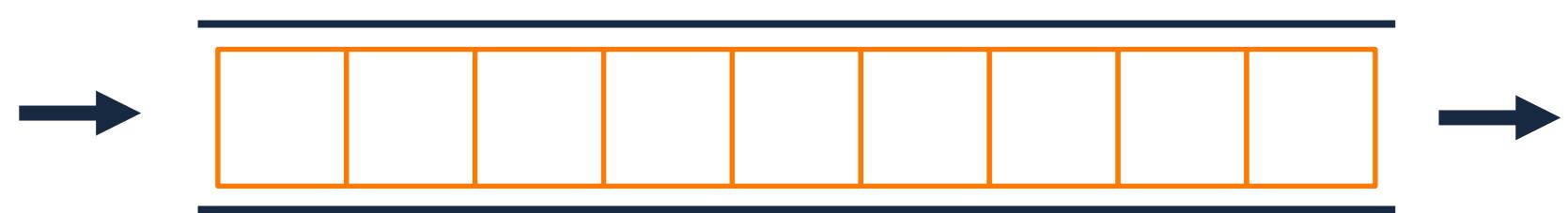


# queue/main.cpp

```
8 #include <iostream>
9 #include <queue>
10
11 int main() {
12     // Create a std::queue:
13     std::queue<std::string> q;
14
15     // Add several strings to the queue:
16-8     q.push( "Orange" );    q.push( "Blue" );    q.push( "Illinois" );
19
20     // Print the front of the queue out and pop it off:
21     std::cout << "First pop(): " << q.front() << std::endl;
22     q.pop();
23
24     // Add another string and then print ouf the front of the queue:
25     q.push( "Illini" );
26     std::cout << "Second pop(): " << q.front() << std::endl;
27
28     return 0;
29 }
```

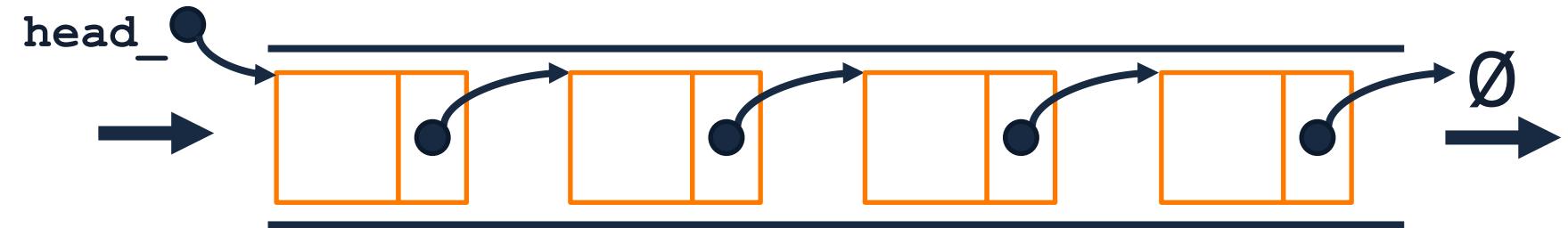
# Implementation

Array-based:



# Implementation

List-based:



# Objective: Create a Queue Data Structure

|        | Array   | Linked List<br><small>(Doubly-linked list required for O(1) runtimes.)</small> |
|--------|---|--|
| Create | O(1)  | O(1)   |
| Push   | O(1)*<br><small>*Amortized runtime; occasional need to double the capacity of the array.</small>                | O(1)   |
| Pop    | O(1)*<br><small>*Amortized runtime; occasional need shrink the size of the array to free unused memory.</small> | O(1)   |
| Empty  | O(1)  | O(1)   |

# Queue

- A queue is a first-in first-out data structure that is similar to waiting in a line.
- A queue may be implemented with an array or a doubly-linked list.
  - Both an array-based and a list-based implementation can be built to run in constant,  $O(1)$  running time.

