# CSCI 132: Basic Data Structures and Algorithms

Queues (Linked List & Array implementation)

Reese Pearsall & Iliana Castillon Fall 2024

https://www.cs.montana.edu/pearsall/classes/fall2024/132/main.html



#### toString() : Java method used to provide a string representation of an object

By default, it returns a string that consists of: ClassName@hascode (memory address)





Person@762efe5d

#### toString() : Java method used to provide a string representation of an object

By default, it returns a string that consists of: ClassName@hascode (memory address)







Once again, we need a data structure to hold the data of the queue

- Linked List (today)
- Array (today & Wednesday)

Elements get added to the **Back** of the Queue.

Elements get removed from the **Front** of the queue





The Queue ADT has the following methods:

Enqueue- Add new element to the queue

Dequeue- Remove element from the queue

\*\* Always remove the front-most element

**Peek()**- Return the element that is at the front of the queue

IsEmpty() - Returns true if queue is empty, returns false is queue is not empty



Front

Back

Enqueue



Linked List Implementation

When we enqueue, we add the element to the end of the linked list

When we dequeue, we remove the element from the beginning of the linked list





As we use our queue, we might need to keep track of a few things

- The size of the queue
- The front of the queue (not when we use LLs)
- The **back** of the queue (no

(not when we use LLs)







Suppose that we have a queue that can hold 6 elements



We need to keep track of a few things:

1. The index of the **front** of the queue

2. The index of the rear of the queue

- 3. The size of the queue
- 4. The capacity of the queue



Suppose that we have a queue that can hold 6 elements



We need to keep track of a few things:

1. The index of the **front** of the queue

2. The index of the rear of the queue

- 3. The size of the queue
- 4. The capacity of the queue



Suppose that we have a queue that can hold 6 elements



capacity = 6 front = 0size = 0 rear = 0



Suppose that we have a queue that can hold 6 elements



capacity = 6 front = 0 size = 2 rear = 1



Suppose that we have a queue that can hold 6 elements





capacity = 6 front = 0 size = 2 rear = 1





Suppose that we have a queue that can hold 6 elements



15

Suppose that we have a queue that can hold 6 elements



size = 2 rear = 2

Suppose that we have a queue that can hold 6 elements



capacity = 6 front = 0 size = 3 rear = 2









Suppose that we have a queue that can hold 6 elements



capacity = 6 front = 0 size = 3 rear = 3

Suppose that we have a queue that can hold 6 elements





Suppose that we have a queue that can hold 6 elements



$$capacity = 6 \quad front = 0$$
  
size = 6 \quad rear = 5

**Issues with this?** 

**V**I

20

MONTANA STATE UNIVERSITY











$$capacity = 6 \quad front = 0$$
  
size = 6 \quad rear = 5

Suppose that we have a queue that can hold 6 elements



Remove the front element, move front pointer forward one spot

capacity = 6 front = 0size = 6 rear = 5



Suppose that we have a queue that can hold 6 elements



Remove the front element, move front pointer forward one spot





Suppose that we have a queue that can hold 6 elements



capacity = 6 front = 0size = 6 rear = 5







Suppose that we have a queue that can hold 6 elements





- 1. Remove the front element
- 2. Make some room in the back



MONTANA STATE UNIVERSITY





- 1. Remove the front element
- 2. Make some room in the back



Suppose that we have a queue that can hold 6 elements





- 1. Remove the front element
- 2. Make some room in the back

Shift all of our data over one spot



Suppose that we have a queue that can hold 6 elements



$$capacity = 6 \quad front = 0$$
  
size = 5 \quad rear = 4



- 1. Remove the front element
- 2. Make some room in the back

Shift all of our data over one spot

The front of our queue will always stay at zero

Suppose that we have a queue that can hold 6 elements







$$capacity = 6 \quad front = 0$$
  
size = 5 
$$rear = 4$$

Suppose that we have a queue that can hold 6 elements



$$capacity = 6 \quad front = 0$$
  
size = 5 
$$rear = 4$$

## Shift everything over one spot



Suppose that we have a queue that can hold 6 elements





Suppose that we have a queue that can hold 6 elements





Suppose that we have a queue that can hold 6 elements





$$capacity = 6 \quad front = 0$$
  
size = 5 
$$rear = 5$$



$$capacity = 6 \quad front = 0$$
  
size = 5 
$$rear = 4$$

Suppose that we have a queue that can hold 6 elements





Suppose that we have a queue that can hold 6 elements







$$capacity = 6 \quad front = 0$$
  
size = 5 
$$rear = 4$$

Suppose that we have a queue that can hold 6 elements





Suppose that we have a queue that can hold 6 elements





Suppose that we have a queue that can hold 6 elements



Suppose that we have a queue that can hold 6 elements





Suppose that we have a queue that can hold 6 elements





Suppose that we have a queue that can hold 6 elements







$$capacity = 6 \quad front = 0$$
  
size = 4 \quad rear = 3

Suppose that we have a queue that can hold 6 elements



capacity = 6 front = 0 size = 3 rear = 2





