

# Queues

Lecturer: Arran Stewart

# Outline

- ▶ What are queues? When do we use them?
- ▶ How can we implement a queue in Java?

# Queues

- ▶ Queues are a useful data structure for storing an ordered collection of elements.
- ▶ A queue is used when the elements need to be removed in the *same order* as they were added.
- ▶ This is referred to as *FIFO* – an acronym that stands for “first in, first out”.

# Examples

- ▶ We will discuss examples of queues in the real world.

# Terminology

- ▶ We refer to the front of the queue as the *head* and the back as the *tail*.

# Implementing a queue

- ▶ We can implement a queue using any collection that stores data in order, such as an array or a linked list.
  - ▶ We will come back to linked lists later.
- ▶ We will use the collection to store the items in the queue

## Queue operations

- ▶ We also need methods to implement *operations* on the queue.
- ▶ We need a method to *add* items at the back of the queue. This is called *enqueueing*.
- ▶ We need a method to remove an item from the *head* of the queue. This is called dequeuing.
- ▶ It is useful to include a method to return the current length of the queue –
- ▶ and possibly a method to check if the queue is empty.

# A queue in Java

- ▶ How can we implement a queue of Objects in Java?

# A queue in Java

- ▶ We will need to create a *class*.
- ▶ We will need variables to hold data.
  - ▶ We will use an array for the collection of elements.
  - ▶ We will need to track track where in the array the front of the queue is.
  - ▶ We will store the length of the queue.

## Using our queue

- ▶ We will discuss how the queue is used.

# Java methods

- ▶ We will need to define some Java methods to implement the queue *operations*.
- ▶ We will define the following methods:
  - ▶ `isEmpty` – this returns true if the queue is empty and false otherwise
  - ▶ `isFull` – this returns true if there is no more space in the queue
  - ▶ `enqueue` – this adds an item to the end of the queue
  - ▶ `dequeue` – this returns the value of the item at the head of the queue; it *also* removes that item from the queue.

# Limitations

- ▶ We have demonstrated one possible implementation of a queue.
- ▶ One limitation of this implementation is that the queue has a fixed maximum capacity. If we try to add more elements than the queue can hold, that results in an error.
- ▶ Later, we will see that using a linked list rather than an array will make it easier to resize the queue as needed.