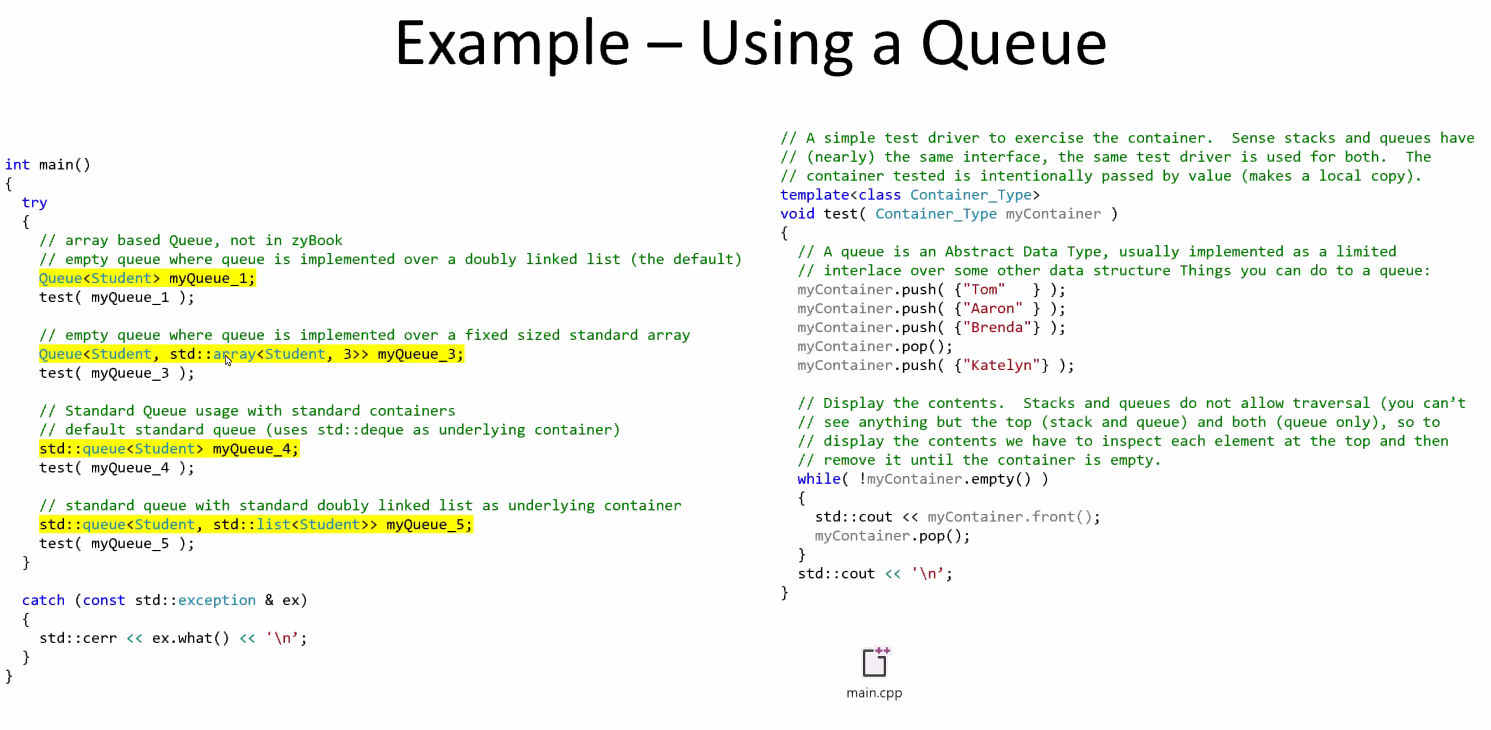
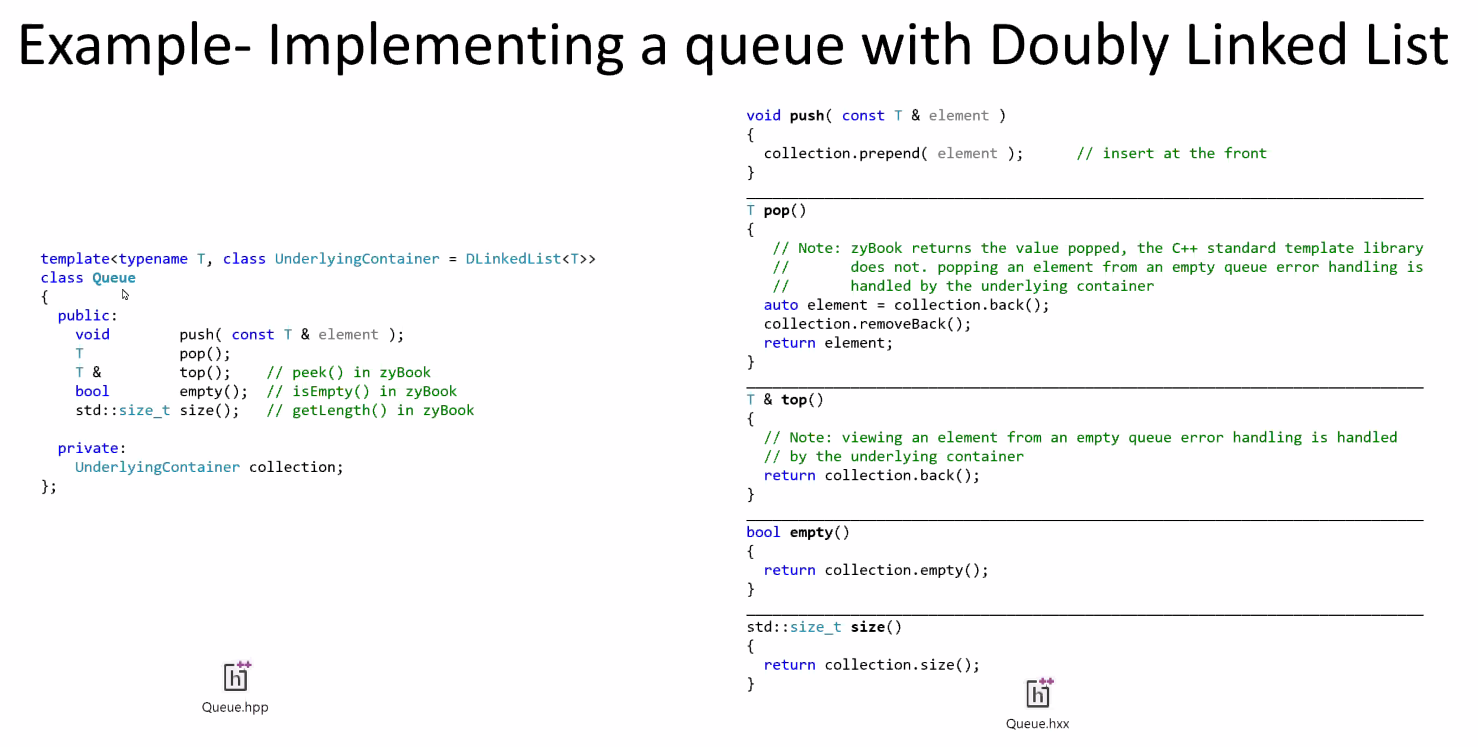
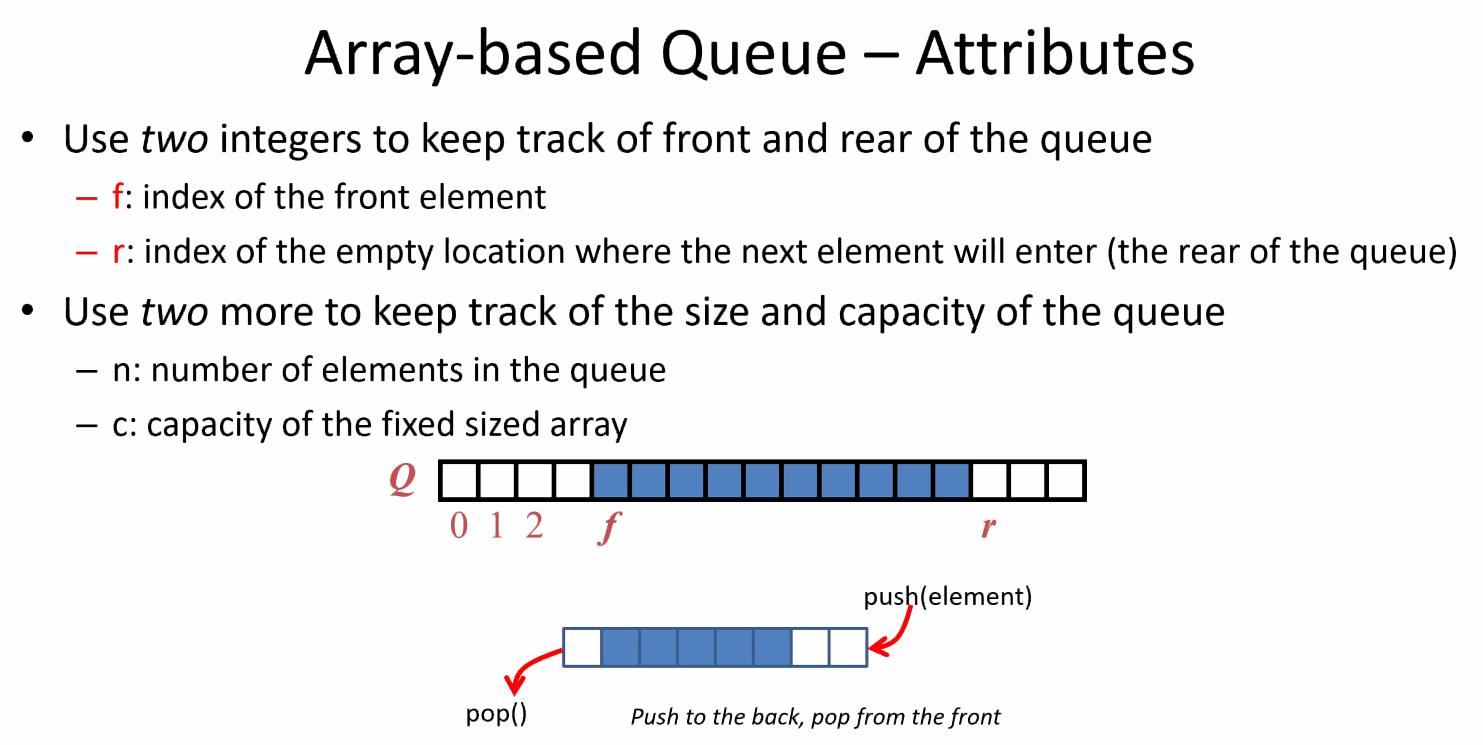
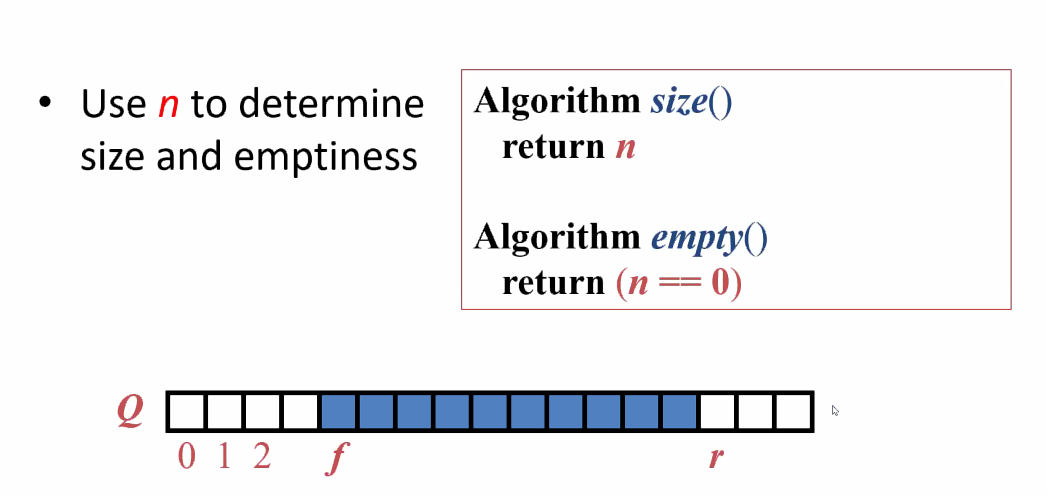
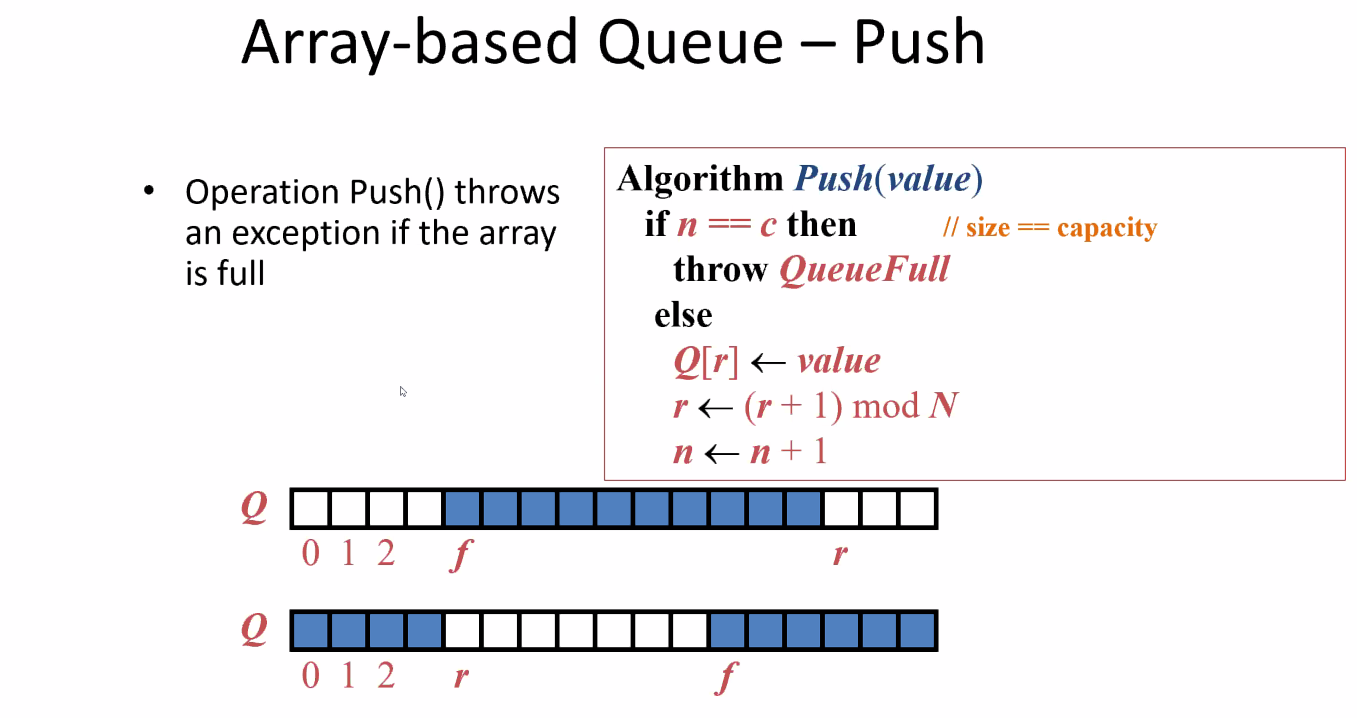
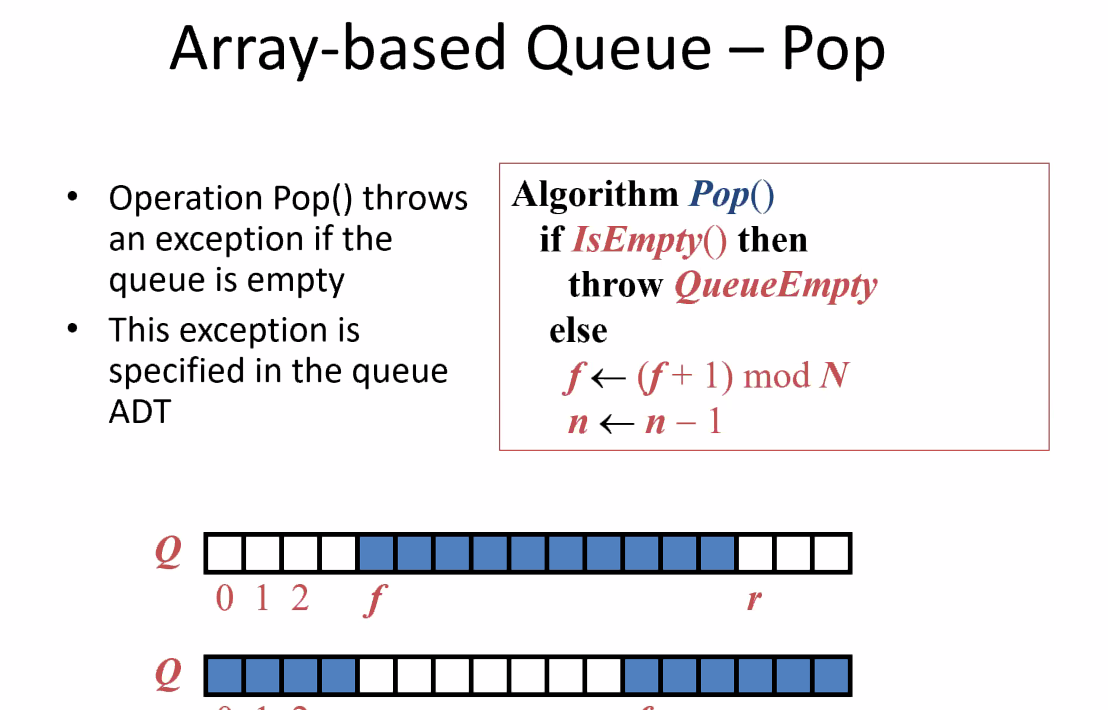
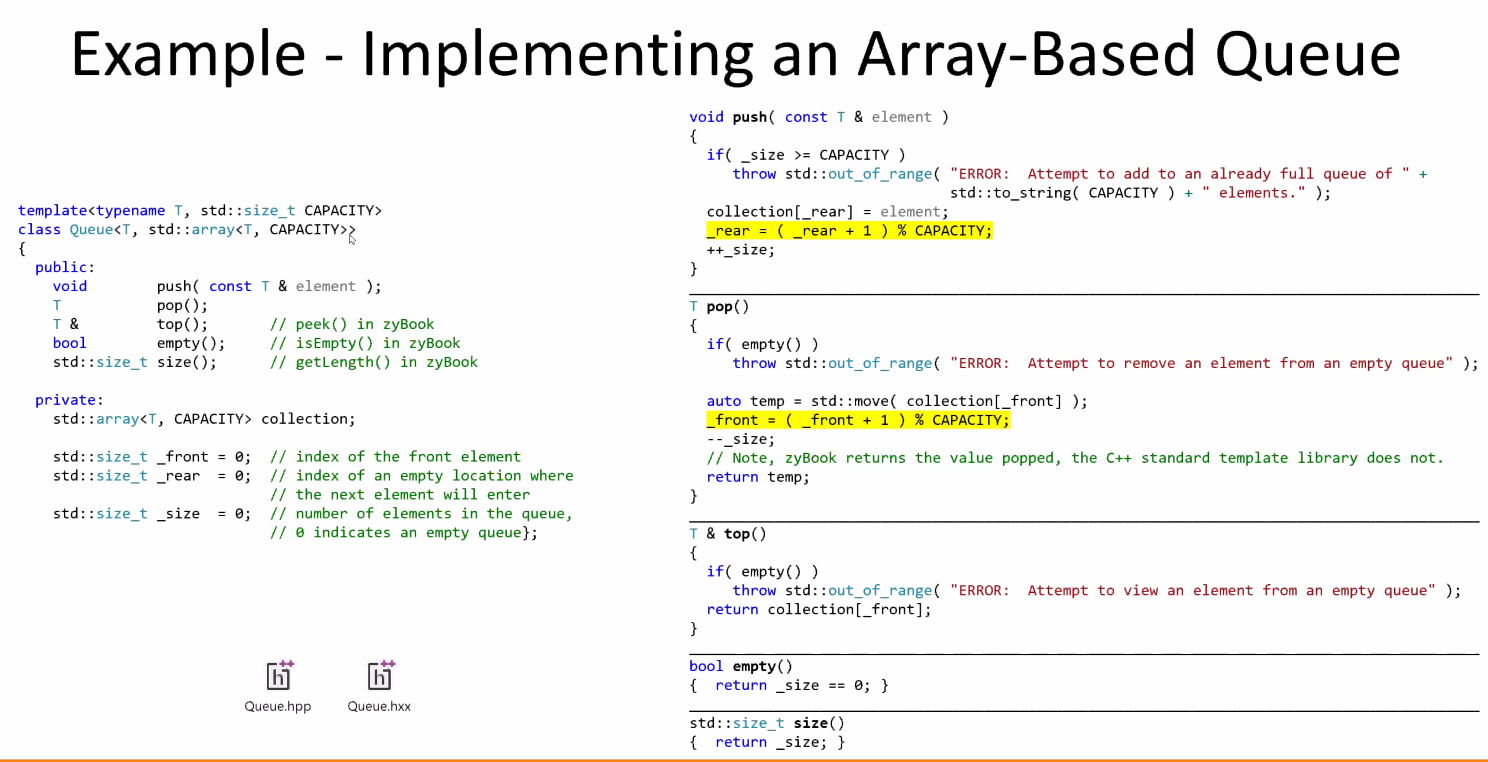
Lecture 13

CPSC 131  
10/12/2020

1. A Queue
   1. 
   2. The singly list does not maintain a pointer to the back of the list. So we don’t have a concept of looking at a singly list and finidng a front and the back.
   3. Cannot look into the middle of the queue just like stack
   4. Cannot traverse for the elements in the queue
   5. To know what’s in the cube, take things out until queue is empty
2. IMplemeneitng queue with Doubly Linked list
   1. 
3. Implementing with a fixed sized array
   1. Array-based queue - Attributes  
      
      1. How to implement queue over array and still maintain consistant push/pop?
      2. N = next available slot
      3. Queue let front() move to keep track
4. Array-based Queue – Size and Empty
   1. 
   2. If there is no more capacity, there would be an error
   3. Going to have to wrap around and start over at the beginning to add more
5. Array-based queue – Push
   1. 
   2. If size = capavity = no more room
   3. But as long as not filled to capacity, could insert next value
   4. Then comes incrementing r, r wants to point to the next available slot. Add one, modulo arithmetic, then that’s your next available slot.
6. Array – based queue – Pop
   1. 
   2. You got to have something to pop for it to work
   3. You pop, move to front, then decrement by one. Still need to use modulo arthimetic to move around
7. Code
   1. 
   2. Rear points to actual element at the end of the queue
      1. Why this way?
      2. If you have rear = next available, then you lose one slot in your capacity.
   3. How to know if you have an empty or full queue?
      1. You have to look at size.
      2. Need to maintain a separate attribute for size because Prof. won’t forgive you if you forget.
   4. Push and pop
      1. When you try to add, verify you have capacity
      2. Let’s take an element and put it at the rear of the queue and increment the rear of the queue.
      3. Have to add 1 and modulo capacity to wrap around.
      4. Ex. If capacity = 5, populate element = 4. \_rear+1%5 = \_rear+0
      5. Array-based queues require modulo-based math.
      6. Stop when size = capacity
8. Sketching queues – Doubly Linked List
   1. Doubly Linked List  
      A close up of a whiteboard

      Description automatically generated
   2. Fixed array  
      Text, whiteboard

      Description automatically generated
   3. Stack vs Queue  
      Diagram

      Description automatically generated