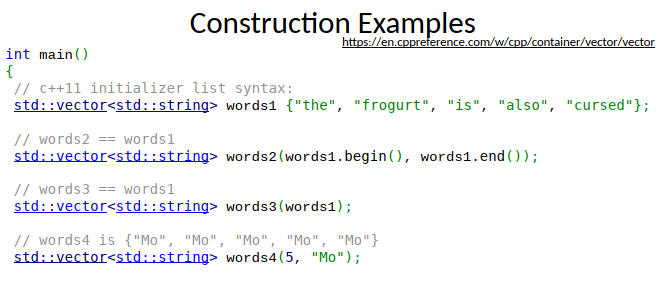
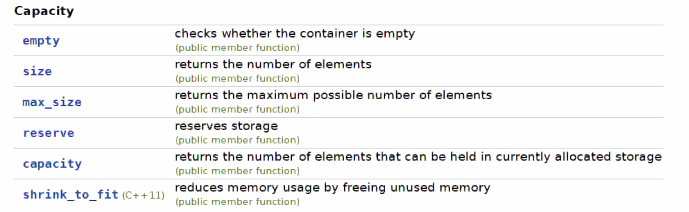
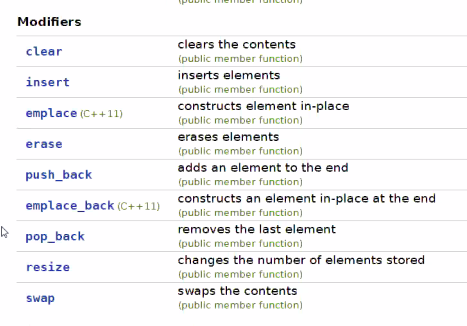
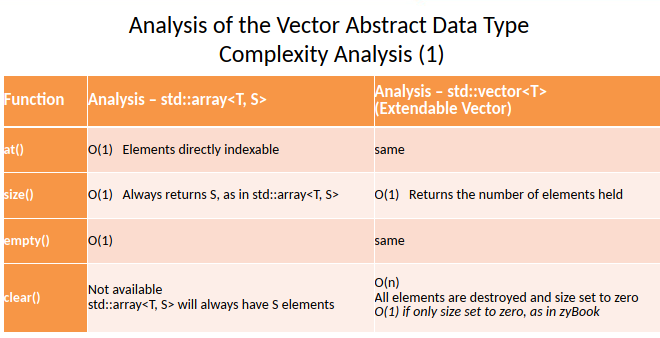
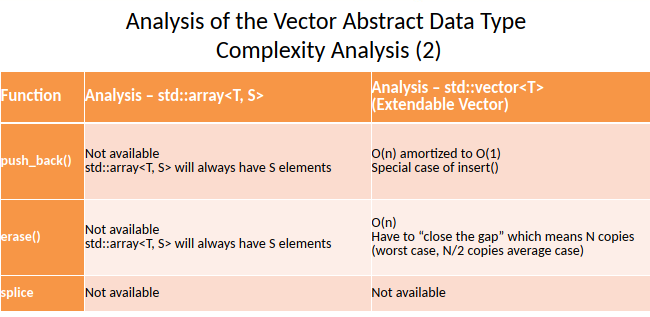
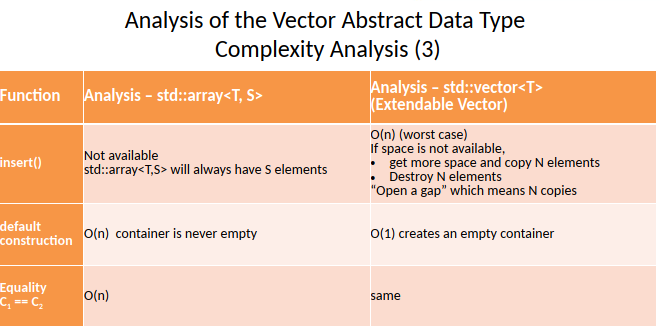
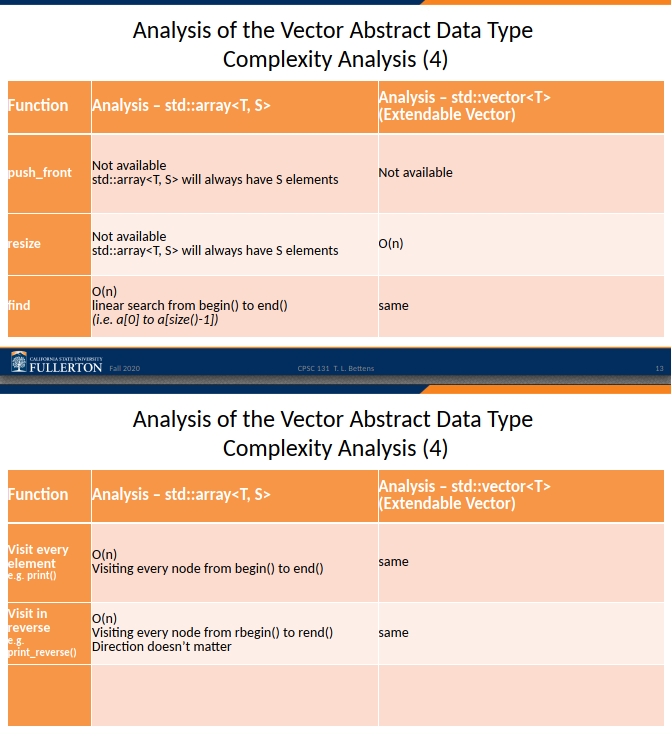
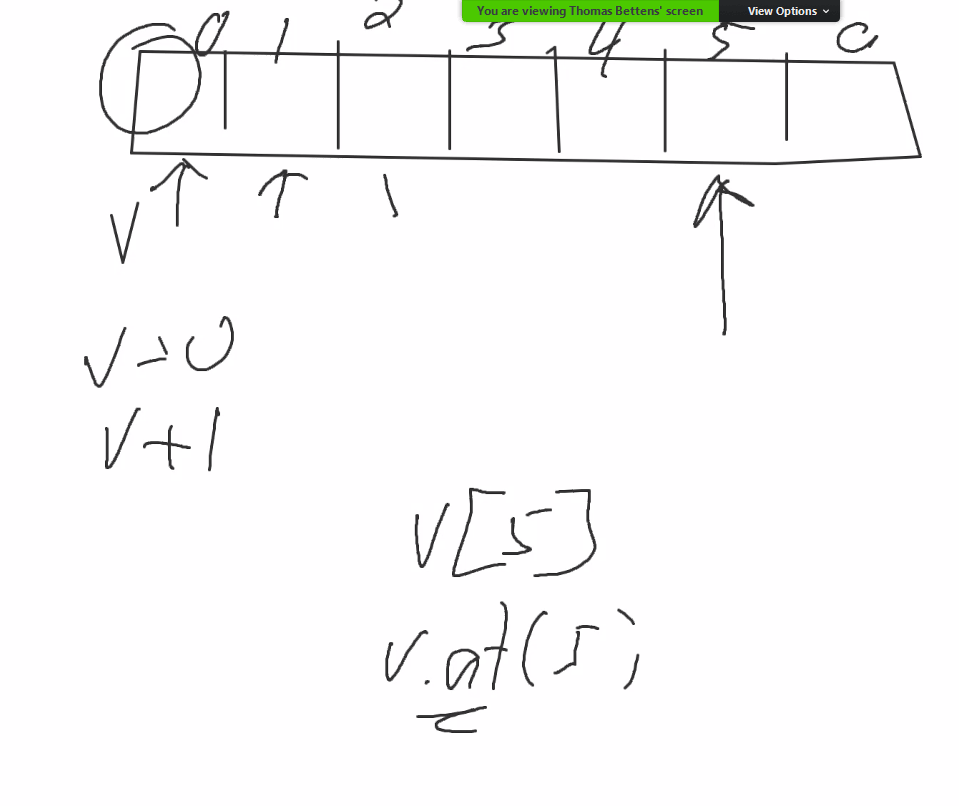
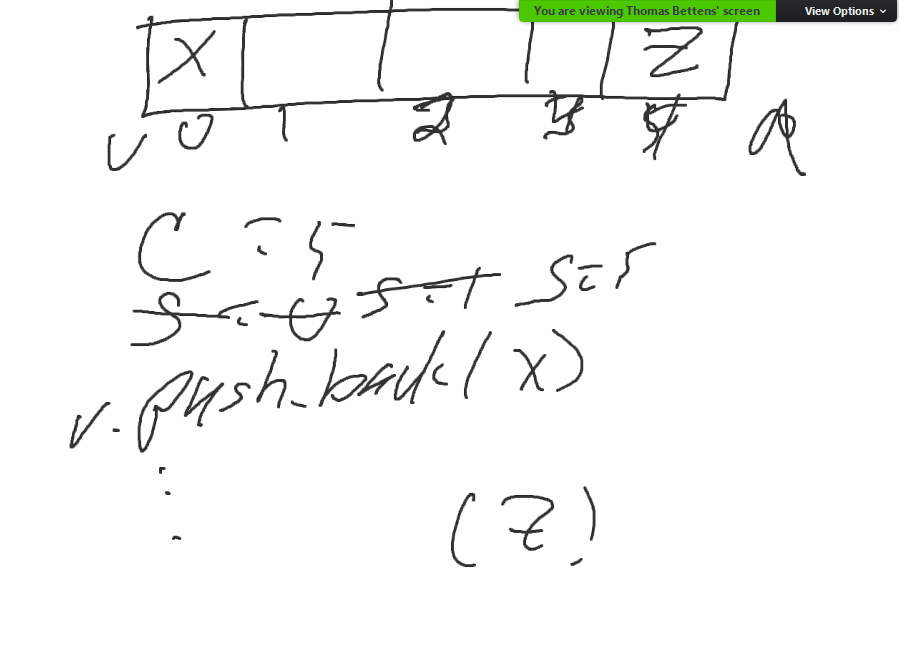
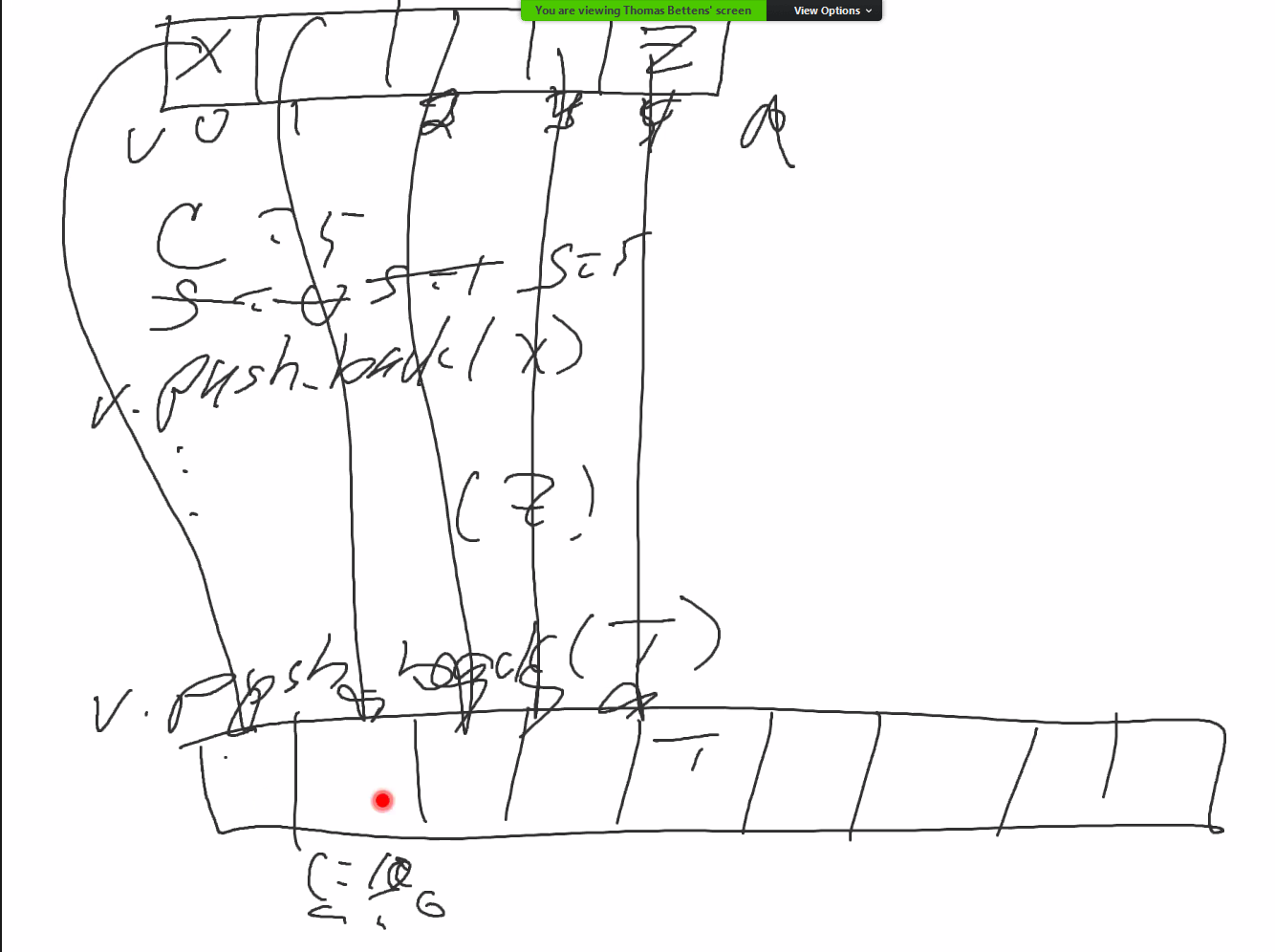
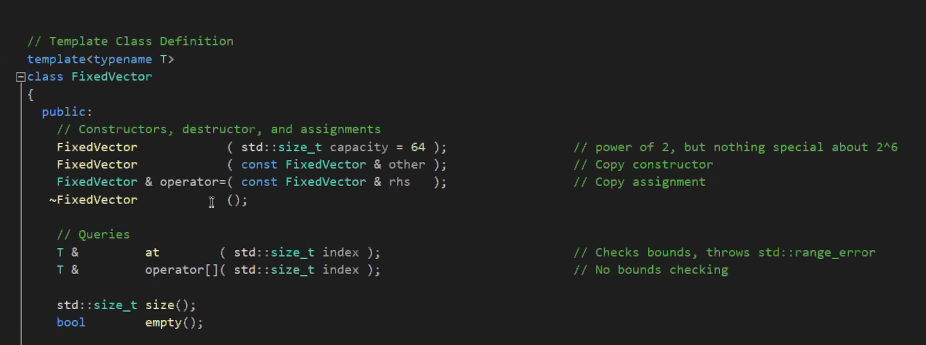
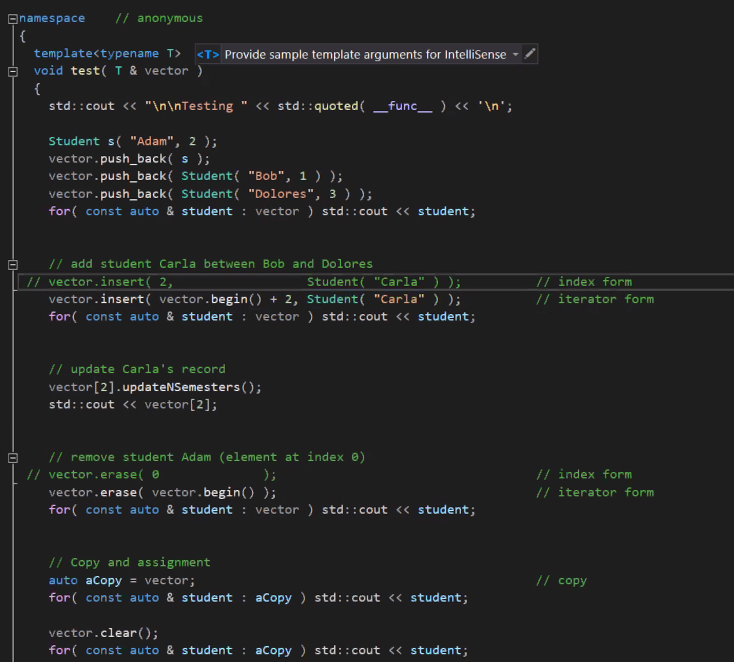
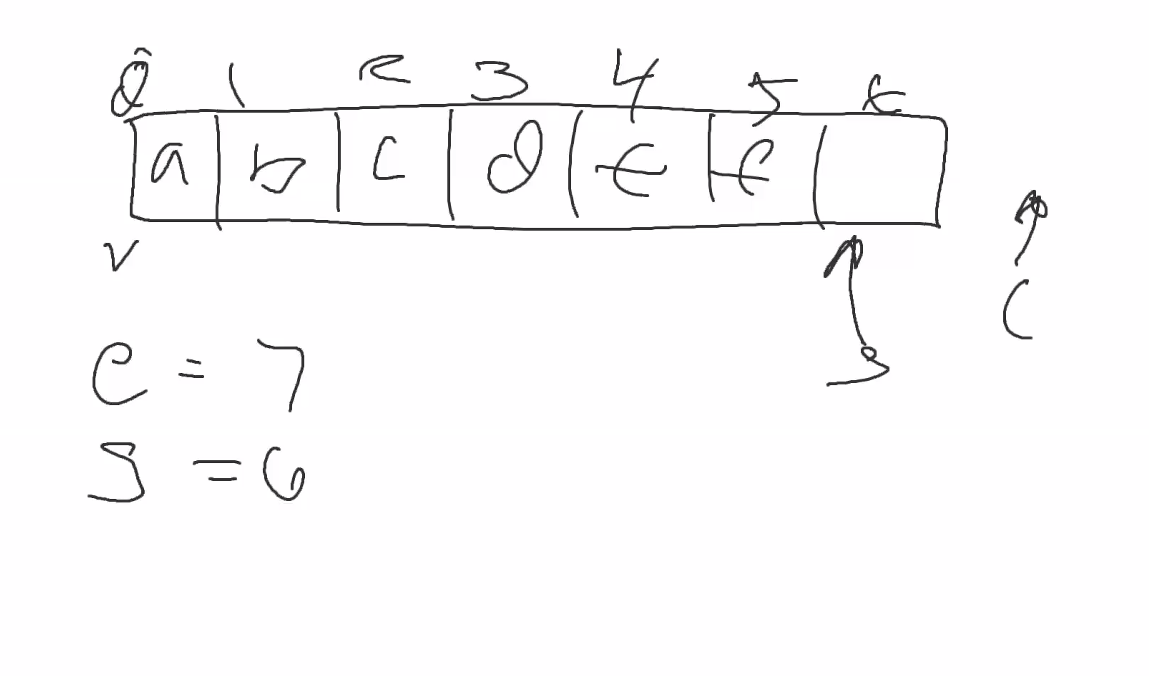
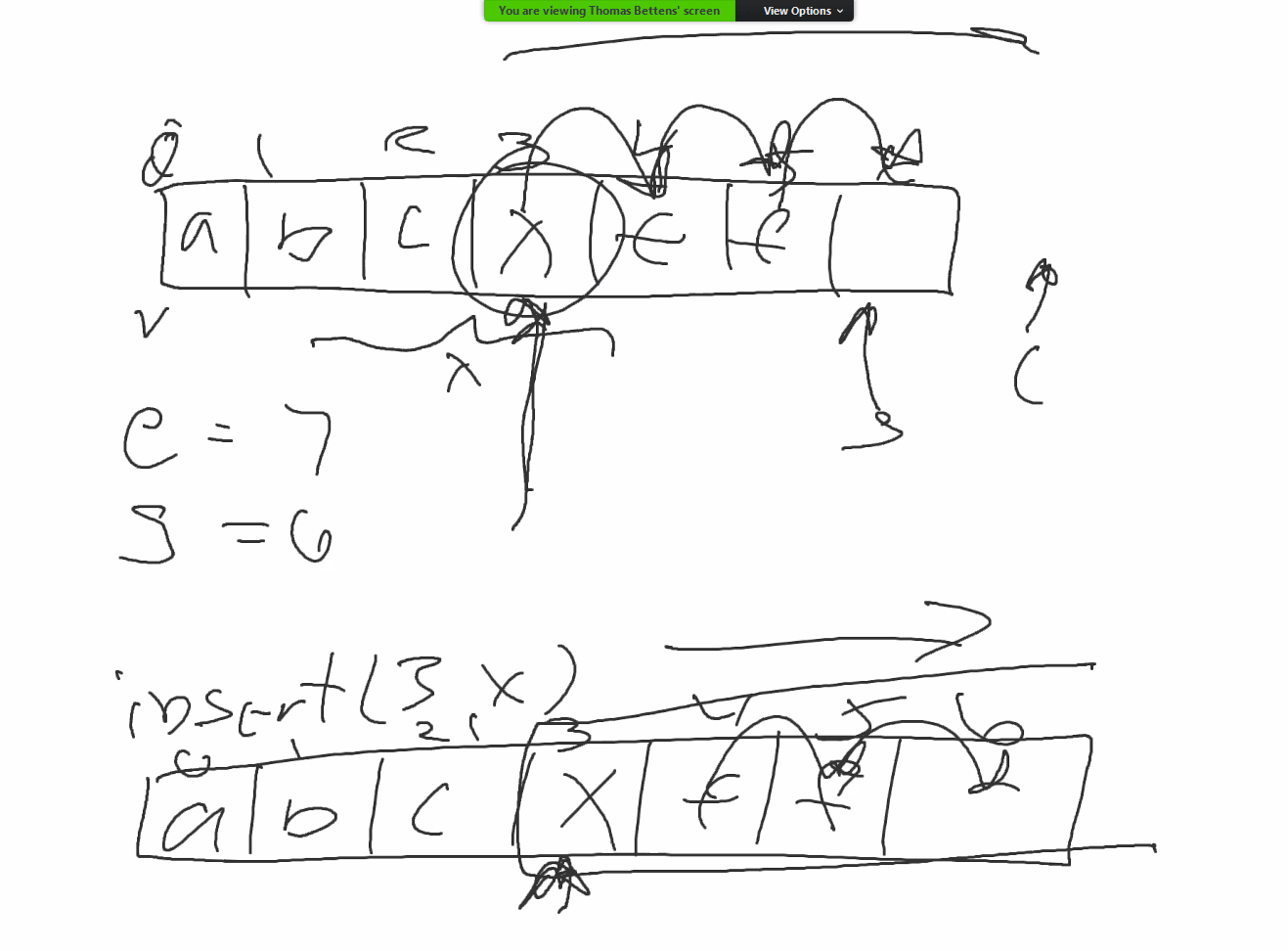
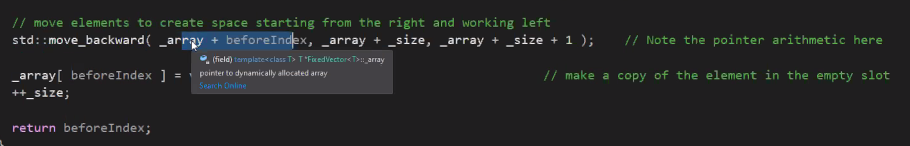
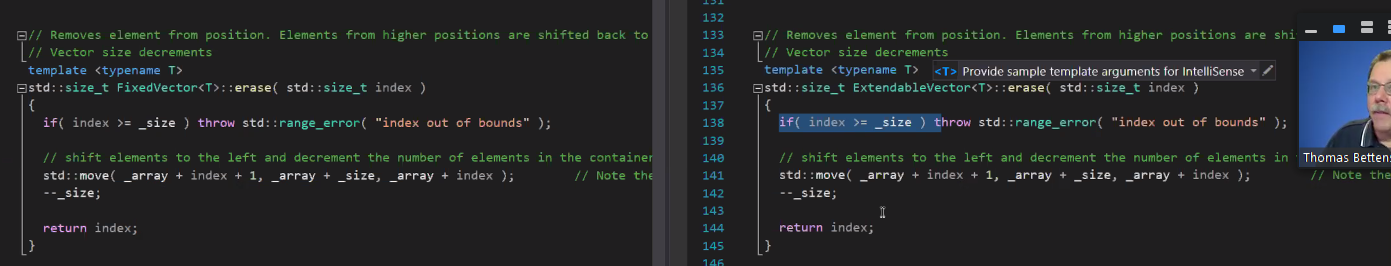
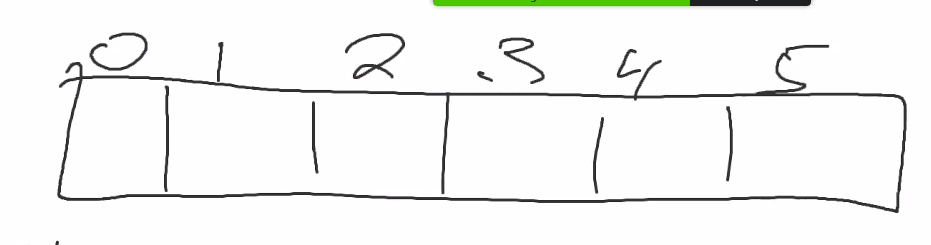
Lecture 6

CPSC 131  
9/14/2020

1. Vector Class
   1. We will get started on Lists on Wednesday
   2. What are they (review)
      1. Containers that is very similar to arrays yet are different by how they are implemented, utilized, and the sheer diversity of what datatypes are acceptable
      2. Construction Vector Abstraction Example  
         
      3. The commands you can use to tell vectors what to do (a sample)  
           
         
      4. The interface of a vector
      5. The Big-O Analysis of Vectors
         1. 
            1. Example: the difference between [] and at(), they both do the same thing but at() is O(1) which is constant and [] are O(n) which depends on the size.   
               Retrieving the element is irrelevant to the size.
         2. 
         3. 
         4. 
         5. In other words, it ultimately depends on the type of vector, the situation it is in, how it is implemented to determine the Big-O
      6. Vector abstraction is contiguous memory  
         
      7. Vector push\_back() in memory
         1. You push something in there and then size changes, capacity changes
         2. Copying and then push\_back()  
            
      8. Fixed Vectors (in code)  
         
         1. When you look at size\_t capacity, you will note that it goes by power of 2
         2. Bool empty() = return true/false if empty
         3. Size\_t size() = return size number
         4. You need to maintain size, capacity, memory
   3. Vectors in past project example
      1. 
      2. What we want: insert Carla between Bob and Dolores
         1. Vector.insert() = inserts an element between 2 elements. It pushes over the elements to the right.
            1. The value of the vector[4] = vector[3], vector[3] = vector[2]
            2. Illustrated  
                 
               
         2. Vector.erase() = point to the element to be removed and shift everything past that spot to the left (close the gap)
            1. Illustrated
         3. Move\_backward() = array.size = end of the contents of the vector, helps with the erase process by moving elements one spot  
            
         4. FixedVector vs ExtendableVector
            1. Capacity is const, capacity is not
            2. More errors to throw in Fixed
            3. 
   4. Contiguous Memory Exercise
      1. Sketch a vector  
         
         1. Sketch a fixed size vector with capacity = 6 [size =6]
         2. Sketch an extendable vector with cap of 6 [size =0]
         3. Now sketch both with these commands:  
            push\_back(5)  
            push\_back(10)  
              
            s = 2, c = 6  
            c^| [c=6 would point to 2]
2. How to Practice
   1. Study for quiz
   2. Look at the starter code for our projects
   3. Look at the sample codes provided on canvas; especially look at the sample codes