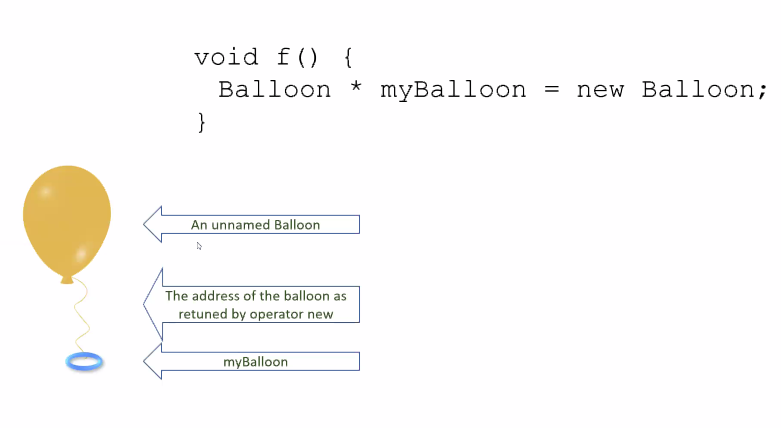
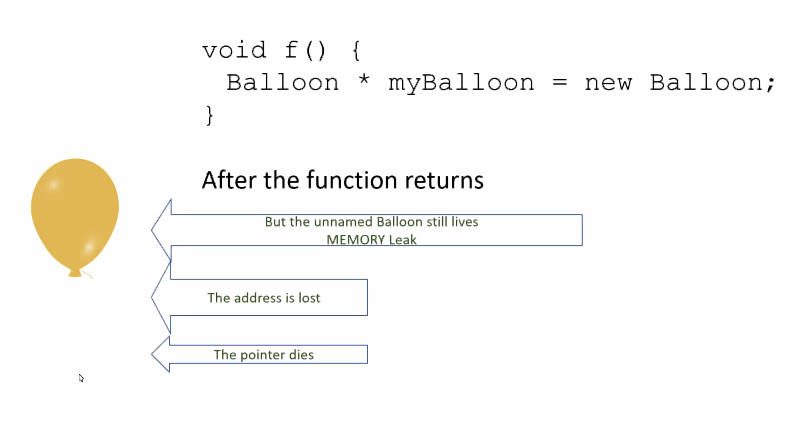
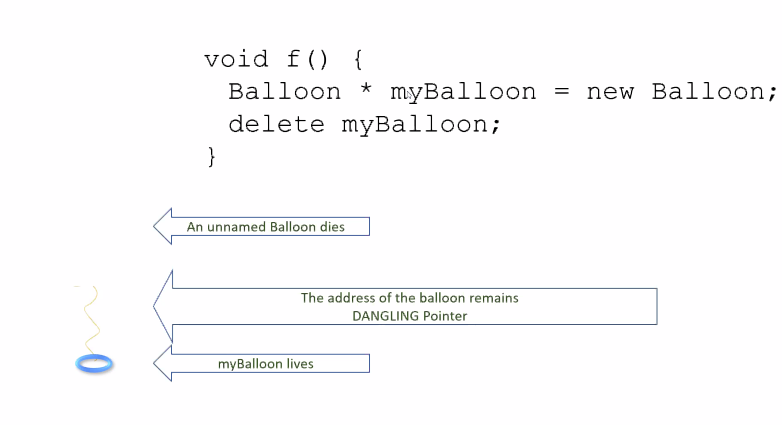
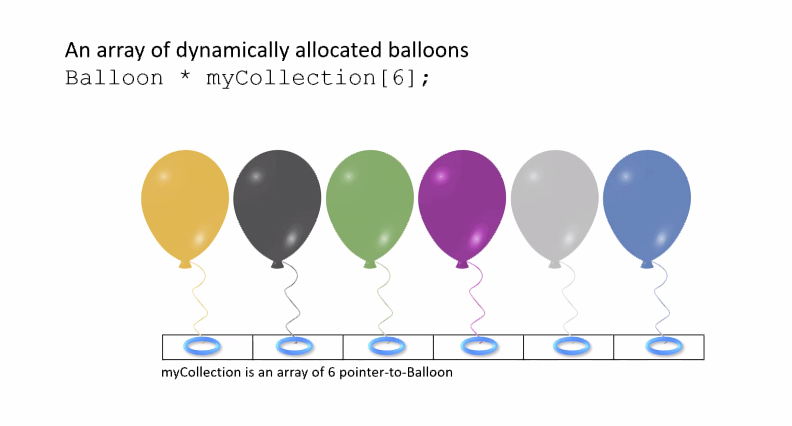
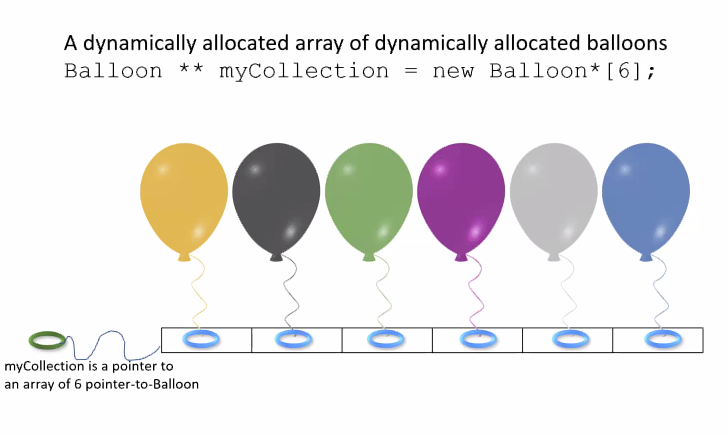
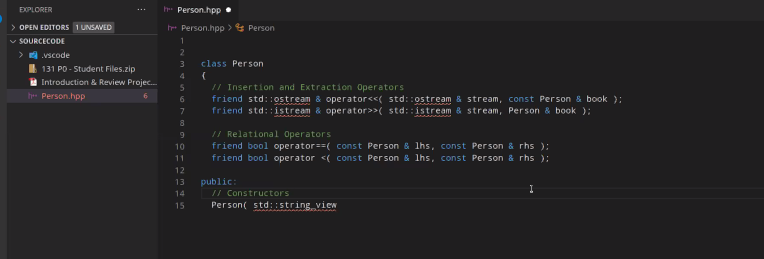
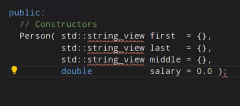
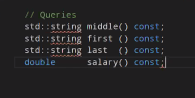
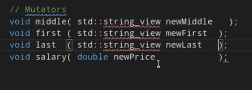
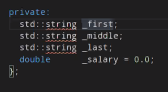
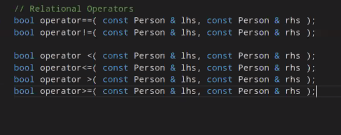
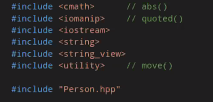
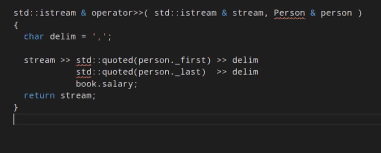
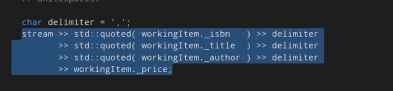
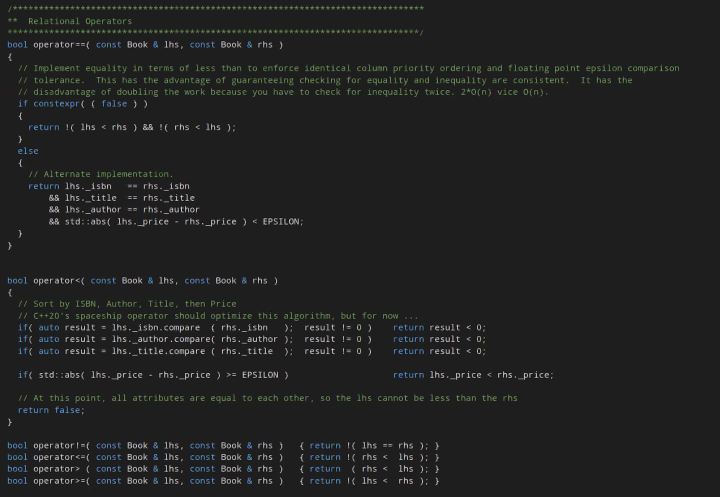
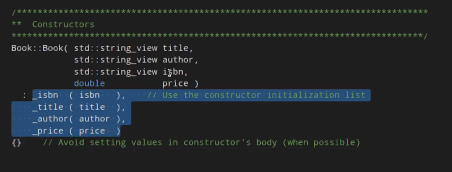
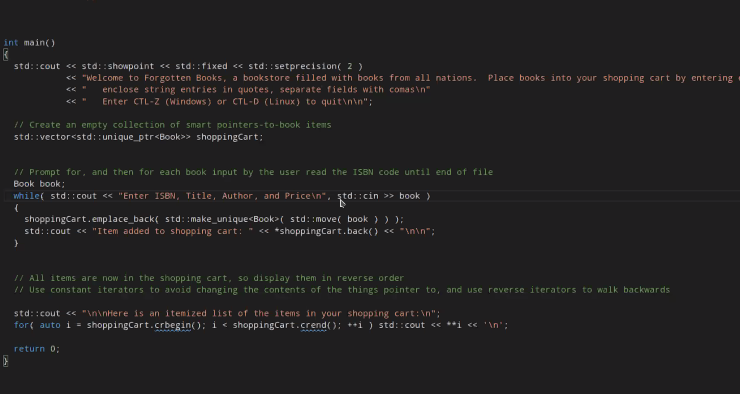
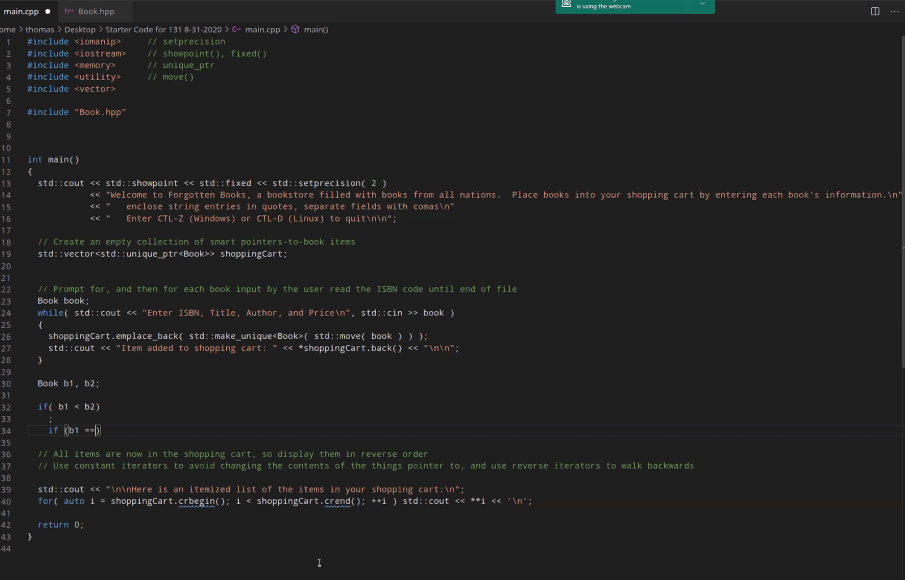
Lecture 3, 8/31/2020 CPSC 131

1. Pointers
   1. Pointers have a life, size, state; it’s an object through and through.
   2. Think of it as a balloon whose string is tied to the wrist of a child.  
      Balloon = pointer  
      Child = reference
   3. If that balloon’s string is broken, it becomes a *null pointer* as it references nobody.
   4. Void f(){  
      Balloon \* myBalloon = nullptr; //initializing Balloon  
      }  
      
   5. When you take the ring away  
      
      1. You get a memory leak
   6. Operators
      1. new = returns a new object pointer
      2. delete = destroys the pointer object   
         
         1. The Ring and String are dangling but the Balloon has been popped
         2. We got a dangling pointer
   7. Pointer Arrays  
      
      1. How do we access the thing a string is pointing?
         1. We dereference
            1. Start with the pointer then follow the string to find the balloon
         2. Have new returns the collection (aka green ring)  
            
         3. Remember, you can only tie one string to a string. But you can untie one string from a ring and tie it to another.
2. Project 0
   1. Class Person  
      
   2. Create constructor for Person  
      
      1. Default args allow me to overthrow information
   3. Queries (asking for information)  
      
      1. These functions only work on constant functions/objects.
      2. These cannot be changed.
   4. Mutators
      1. But if you wanted to change these functions  
         
      2. These modify non-constant objects
   5. The private objects  
      
      1. Initialization: The first three are empty strings and the last is set to 0
   6. Relational Operators  
      
      1. Look at all those Booleans!
         1. These signatures are very common
         2. The relational operators are non-member functions because we need symmetry.
         3. Why they aren’t all friends?
            1. All the other relational operators are done in terms of <=
   7. Person.cpp
      1. 
         1. Frist is actually first
      2. This is what the #includes list looks like  
         
      3. How to read Person.cpp
         1. 
            1. What we see is the Person class being read and the person.first, person.last (first name and last name) being std::quoted/read
         2. 
            1. Quoting salary isn’t as elaborately coded as the strings because it is a double.
      4. What we are doing
         1. Creating Person in hpp
         2. Implementing Person in cpp
   8. Back to the project
      1. Book.cpp
         1. We stripped the delimiter  
            
            1. We don’t want to return junk/an error
            2. Please test state of stream before setting a return
            3. This is what they meant by, if anything goes wrong, please don’t change value of the parameter
         2. Relational Operators
            1. 
            2. Comparing and contrasting different objects of the Class
            3. All the other relational operators are written in terms of the first set
            4. So what happens when two Books are not equal?

See how many bool operators there are and that each one refers to different relations (<, >, etc)?

They are the ways you can compare

* + - 1. Remember, initialization is better than assignment  
         
         1. Assignment occurs within the body of the loop
         2. Not as good as initialization
  1. Book Class (main.cpp)  
     
     1. We have a vector of pointers to books
        1. It creates a Book object
        2. Loops until end of loop
        3. Read Books (the whole thing)
           1. Use the operators we just built (the relational ones)
     2. The Relational operators allows you to extend the language to your other types  
        
  2. You need to get the program to Read a fricking Book