Lecture 2, 8/26/2020 CPSC 131

1. Projects
   1. Forming Groups will be held in People -> Groups -> Switch To if you want to
      1. In Group 2 with Patrice
      2. Groups change with each new assignment
      3. Will be locked three days before the assignment is due
      4. One submission to the group
      5. Group has own space to work in
      6. Solution is given day after the assignment is due
      7. But you are expected to give a Peer Review
         1. You are assigned an assignment
         2. You will have access to it on the Canvas but you won’t know WHOSE assignment is there
         3. You will leave comments on the Peer Review
            1. Be courteous but honest
   2. The Project Itself
      1. Look at zip file for the assignment for further information
      2. You will write a Class called Book
         1. Attributes
            1. String Title
            2. String Author
            3. String ISBN
            4. Double Price
         2. Constructor
            1. Allow books to be constructed with 0, 1, 2, 3, or 4 args
            2. Initialize each attribute with members
         3. Operations
            1. Set and retrieve each attributes  
               void isbn ( const std::string & isbn); //mutator  
               std::string isbn() const; //query
            2. Remember, objects know how to read and write themselves
            3. Overload the insertion and extraction operations  
               Book book;  
               std:: cin >> book; //read  
               std::cout<<book; //write

Insertion and extraction/read and write shall be symmetrical. So should read what you wrote

* + - * 1. Overload the 6 relational operators

Book interestingNovel, boringNovel;  
if (interestingNovel ==boringNovel)…  
if(interestingNovel < boringNovel)…

If equal, all attributes are equal

But if interestingNovel and boringNovel’s ISBN are equal but interesting author is less than boring author, interesting is less than boring

* + - * 1. Separate interface from implementation suing header (.hpp) and source (.cpp)
      1. Function main() to use Book class
         1. Read book from standard input until end of file

Store book in dynamically allocated obj

Store pointer book to vector

* + - * 1. At end of file, write books to std::out
    1. What to turn in
       1. 4 Things
          1. Book.hpp

Don’t modify but need to include it into the Project Folder because you need a complete deliverable

Book.hpp is part of the solution

* + - * 1. Book.cpp
        2. Main.cpp
        3. Output.txt
      1. How it will be graded
         1. What my Book.hpp is being compared to original Book.hpp

THIS CANNOT BE MODIFIED

What you are given will be turned in

Again, don’t change it

* + - * 1. The rest of your project files depend on implementing the .hpp

1. Review Rational Case Study
   1. A rational number is a precise way of measuring not-irrational numbers
   2. This is a resource you’re going to look at throughout this semester
   3. Download, compile, use it, and know that it contains all the examples we are going to study
2. Pointers, Arrays, Dynamic Memory
   1. Array: A container of elements
   2. Pointers: The address of a variable (but not the value of the variable itself)
   3. He mostly spent the time pointing out a specific part of the Rational Case Study
      1. Main.cpp of Rational   
         RationalArray myArray //a dynamic array
      2. Implementing function bool operator< and lessThan  
         look at return safeMult(a, y/GCD) < safeMult(x, b/GCD);   
         This expression will evaluate whether true/false and then return one of the two
         1. Basically it is supposed to eliminate redundancy
3. Okay so he didn’t actually explain what pointers, arrays, dynamic memory so he’ll do that next week
   1. Today he mostly explained
      1. The Project
         1. But didn’t go over what is the Regression Folder
         2. But he did tell us what to do and what is expected
         3. And that he doesn’t have a Late Policy
      2. Rational Case Study
         1. It’s a reference folder to look at
         2. It’s compile-ready