COMP 2404 -- Tutorial #7

Inheritance

Learning Outcomes

After this tutorial, you will be able to:

- apply the concepts of inheritance

Instructions

1. You will begin with the code you saved from Tutorial #6.

2. We are going to create a small hierarchy of books, with two sub-classes: one for fiction books, and another for non-fiction books. For this tutorial, our book sub-classes will not contain any additional data members or member functions. Our `Control` object will store two library objects: an SCS library object for the non-fiction books, and a lounge library object for the fiction books.

   Update your UML class diagram to represent the new class associations in the program.

3. Create a `FictionBook` class and a `NonFictionBook` class that both derive from the `Book` class. Each will contain a constructor that takes all the necessary parameters and calls the base class constructor.

4. Modify the `View` class by adding a new `readBookType` function that prompts the user for a book type ("Fiction" or "Non-Fiction").

5. Modify the `Control` class as follows:
   - replace the existing library data member with two library data members: an SCS library that will store non-fiction books and a lounge library to hold fiction books
   - update the `launch()` function as follows:
     - using the `View` object, prompt the user to enter the type of book to be created
     - create either a new `FictionBook` or a new `NonFictionBook` object, depending on the user's selection; there will no longer be any `Book` objects created
     - add the new book to the correct library (SCS or lounge)
     - use the `View` object to print both libraries to the screen at the end of the program

   Note: Neither the `Library` nor the `List` classes will change from Tutorial #6. Each `Library` object will still have a `List` object that holds a doubly linked list of `Book` pointers, and the list will still order the books based on the `Book` class's `lessThan()` function, which is inherited by both sub-classes.

6. Modify the `in.txt` file accordingly. Build and run the program. Check that each book is in the correct library, and that the books are ordered correctly, both in the forward direction and the backward direction, when both libraries are printed out at the end of the program.

7. Make sure that all dynamically allocated memory is explicitly deallocated when it is no longer used. Use valgrind to check for memory leaks.

8. Package together the tutorial code into a tar file, and upload it into cuLearn. Save your work to a permanent location, like a memory stick or your Z-drive.