EE208 Laboratory Session 9

12th April 2022

1 Lab objectives

In this lab you will build on last weeks solution. You will continue to work with classes in C++ that use some of the inheritance techniques discussed in the lectures. You will also be required to design your solutions using object oriented design techniques and class diagrams. Please quickly revise recent lectures before proceeding. Reread last weeks lab before continuing; you should be completely familiar with that lab before proceeding.

Learning Outcomes

Having completed this lab you will be able to create inheritance hierachies in C++. You should understand the details of object construction and destruction within such hierarchies. You should also know how to create and manage C++ project with MSVisual Studio. A core objective of this lab is that you design your solutions for the first time using OOD and UML techniques. *This means I wont be telling you exactly what to do anymore; you will be expected to figure out your own solution from a basic set of requirements.*

2 Questions:

For each of the problems given below write a C++ program that provides a solution. Unlike last week you will be graded on your work this week. Your assignment is due for Monday 25th of April.

Remember:

- **Comment your code.**

- **Use proper indentation for function and control structures.**

- **Make sure you take the time to create a complete class diagram for your solution.**

- **You can continue to use the command line compiler if you wish. However, this makes life a little difficult for debugging your code. I would encourage you to begin using Microsoft Visual Studio Express, which is available on your lab PCs.**

Exercise 1: Eason's hired someone else before you; they completed some work before they fell out with the Easons CTO and got themselves fired. You've inherited their code, which is placed on moodle for you to download. The class diagram for this solution is shown below. For the first exercise you need to download the various cpp and h files, create an empty Win32 console appllication using Visual Studio, and add all of these files. Make sure it runs and then take some time to go through the class diagram and all of the code to make sure you understand the work of your predecessor. They were a little lazy with commenting so you should feel free to comment the code as you read through it and make sense of it.

The reason your predecessor was fired was because of a difference of opinion with the CTO. The CTO believed that the book class should not contain a data member for the number of copies; the CTO believed that a book object should not be responsible for knowing how many copies of the book were in a shop. Remember from your design lectures that a class should do one thing and do it well, and a book object should have the same properties as a physical book, which would not contain information about how many copies of it exist. Obviously the CTO was correct. The second thing you need to do is to edit the book class such that the data member num_copies is moved into the book_details class. Make sure your code still runs.

The third thing you should do is redraw the class diagram, and this time make sure you thoroughly write in all of the data and functions as you learned how to do in the recent lectures on design.

You should upload all of your commented code and the class diagram in a folder called Exercise1
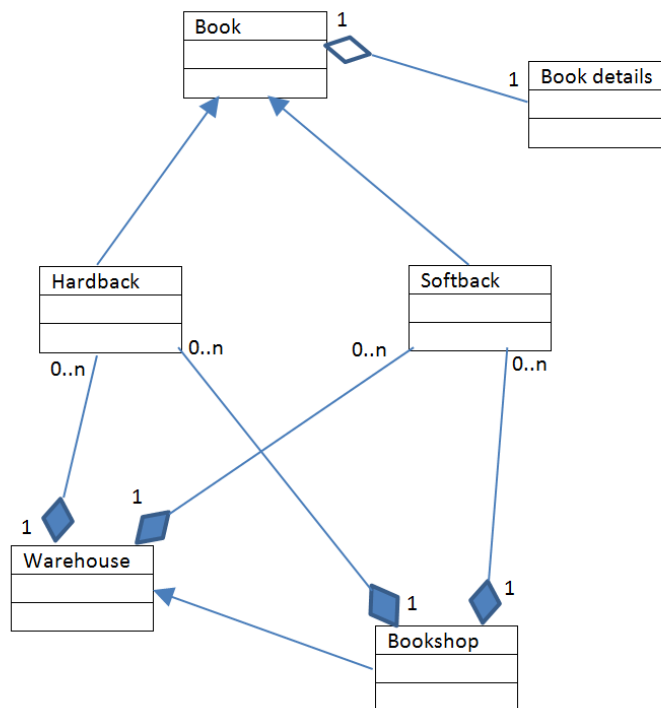
Exercise 2: Your predecessor also forgot to include a magazine class. You must adapt your code to include such a class. You should update your class diagram accordingly. Full marks will require the use of inheritance.

Demonstrate your new class in action in a main function as you see fit. You should upload all of your commented code and the class diagram in a folder called Exercise2

Exercise 3: You will have noticed by now that your warehouse and bookshop classes contain arrays of hardbacks, softbacks, and if you've done a good job on Exercise2, they should also contain an array of magazines. For the third exercise you are required to replace these arrays with vectors, which will be covered in one of the next lectures. For more information on vectors, consult this hyperlink. This should be enough to work on the solution of this exercise. You should add a new member function to the warehouse class called add_book() that allows a new book object to be added to the vector using the push_back function(). Also add a function remove_book() that allows a book object to be removed from the vector using the erase() function. This will require the use of iterators.

Demonstrate your new functionality in action in the main function as you see fit. You should upload all of your code in a folder called Exercise3