

# COMP 2404 -- Tutorial #4

## Dynamically Allocated Objects

### Learning Outcomes

After this tutorial, you will be able to:

- work with dynamically allocate objects
- write a destructor to clean up dynamically allocated objects
- work with an array of pointers
- insert into an array by shifting elements

### Instructions

1. You will begin with the code you saved from Tutorial #3.
2. You will change the `main()` function to dynamically allocate each new `Book` object when the user enters the information for a book.
3. Make the following changes to the `Array` class:
  - books will be stored in an array of `Book` **pointers**, instead of `Book` objects
    - you will need to update the `print()` function to work with this change
  - change the `add()` function as follows:
    - it will take a `Book` pointer as parameter
    - it will insert each new book into the array so that it stays in *ascending order by year*; this will require shifting some books towards the back of the array to make room for a new one
      - terminology: *ascending* means increasing; *descending* means decreasing
    - it will compare book years using a new `lessThan()` function in the `Book` class
  - add a destructor to clean up the dynamically allocated `Book` objects
4. Modify the `Library` class so that the `addBook` function takes a `Book` pointer as parameter.
5. Add a new `lessThan()` function to the `Book` class. This function will take a `Book` pointer as parameter and return a boolean. It will return `true` if the year of the `Book` on which the function is called is less than the year of the `Book` parameter.
6. Build and run the program. Check that the books are ordered correctly when the library is 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, as we saw in class when we covered section 2.3.
8. Package together the tutorial code into a tar file. Start up a browser in the VM, log into cuLearn, and go to the tutorial page. Select the tutorial submission link, and upload your new tar file.
9. Save your work to a permanent location, like a memory stick or your Z-drive.