

# HOMEWORK #1:

## *First Job!*

**Due Date:** [Monday, 18th of October, 11:59:59pm](#)

For this assignment, you need to submit a file called 'arraylist.hpp'. Remember to put your **name** and **section** at the top of your program file.

### **Problem:**

Good News!! You have been hired by "*Planet Express Softworks*", a new division of "*Planet Express Inc.*" and the newest venture of Prof. Farnsworth. The business model is simple: you write the software, and the delivery crew delivers it. It is not like Prof. Farnsworth is greatly interested in software. He is just interested in using the profits to fund his research.

You have joined just in time. "*Planet Express Softworks*" just found their first ~~victim~~ client.

Emperor Lrrr, from the planet Omicron Persei 8 is requesting your software services. The Omicronian IT services originally outsourced their development to the mud planet Elbonia, and although they developed for a very low price, what they delivered is full of seg-faults and memory leaks. So Emperor Lrrr has decided to take a shot at hiring humans to rewrite one of the software modules.



The Omicronian IT services is giving you a file, `‘arraylist.h’`, an specification of a templated **ArrayList** class.

## Your Job:

Use your coding skills and lead “*Planet Express Softworks*” to create an `‘arraylist.hpp’` file from your knowledge of **Data Structures** and the documentation contained in the `‘arraylist.h’` file.

## Testing:

The Omicronian IT services have also provided you with sample programs that make use of the ArrayList class and their corresponding outputs. You can use them to check if your implementation is working correctly. As long as your ArrayList class behaves just like the original, Emperor Lrrr will be satisfied with your product.

- The program `‘smalltester.cpp’` uses the ArrayList class and is supposed to produce the output `‘smalloutput.txt’`.
- The program `‘largetester.cpp’` uses the ArrayList class and is supposed to produce the output `‘largeoutput.txt’`.

In this assignment, you **will be penalized** for memory leaks. To test for leaks, you can use the "Valgrind" tool available in the UNIX systems:

After compiling with `fg++ tester.cpp -o tester.ex`

run your program with the command `valgrind --leak-check=full ./tester.ex`

## Submission:

Submit your assignment by placing all code in this assignment's git repository in the course's GitLab server, [\[link\]](#) . ( You should have a repository setup sometime next week ).

Your program will be evaluated and graded on the **Computer Science department's Linux machines** so your program needs to be compatible with the current system, compilers and environment.

## Useful Hints:

1. Carefully read the comments of each member function.
2. Write down an algorithm for the function before you start coding it.
3. Develop your member functions one at a time, starting from the simplest ones.  
Move to the next function only after the previous one has been tested.  
Trying to code the whole class and then remove the bugs may prove to be too big a task.
4. Use the provided friend function `'operator<<'` to observe the status of your lists.
5. When a function that needs to return something “panics”, return the `'m_errobj'` member.
6. The default constructor and function `'max()'` are **defined** in the class declaration. You do not need to implement them.