**A7: List of Employees with Random Access**

In this project, you will implement a data structure called List whose behavior is random access, i.e., insertion to a position and removal from a position as long as the position is valid. You will use a singly-linked list to implement List and the position of the first node is 0.

Objectives:

1. **List.cpp: Based on the provided List.h, implement each member function declared in the header file.**
2. In main()

* insert all the employees read from employee.txt at a random position as long as the position is valid. Use a random number generator to set the position of the new data.
* During the debugging phase, display the list per each insertion.
* remove all the employees read from employee.txt at a random position as long as the position is valid. Use a random number generator to set the position of the new data. Test invalid positions as well.
* During the debugging phase, display the list per each remove

**Note**

* Adopt an incremental approach when developing your program. Add one block or function at a time and rigorously test it before proceeding further.
* Begin the project as soon as possible, allowing ample time to complete it well before the deadline. This method ensures systematic development and thorough testing, leading to a robust and error-free final product.

**Submission**

Zip the following and name the zip file in the format of A7\_Lastname\_Firstname.zip

* .cpp, .h, makefile
* Your report must present the following
* 1. A screenshot of a successful run of your program that shows all the cases.
* 2. Explain the order of the constructor calls when an object of List is created and the meaning of the keyword, this, in each constructor. Crop the portion of a screenshot that matches your explanation
* 3. Explain the order of the functions calls when an object of List is out of scope. Crop the portion of a screenshot that matches your explanation