This assignment uses objects that require heap memory. Specifically, you will build the following objects.

1. A node struct, for a linked list of integers.
2. A list\_builder class, for building lists of integers.
3. A fixed\_list class, for compact and unchanging lists of integers.

Class list\_builder should contain a linked list of integers (using the node struct). Implement additional helper methods if you like, and any necessary member variables, to provide the following features.

* **(1 points)** An empty constructor that initializes an empty list.
* **(4 points)** A destructor that cleans up all memory used by the object.
* **(5 points)** Method void add\_to\_front(int value), which adds an element to the front of the list.
* **(5 points)** Method void add\_to\_back(int value), which adds an element to the back of the list.
* **(5 points)** Method void reset(), which removes all elements from the list.
* **(2 points)** The appropriate declarations to disable assignments or copies of a list\_builder object.

Class fixed\_list should contain an array of integers and the size of the array, and should provide the following features.

* **(4 points)** A constructor that builds a fixed\_list from a list\_builder object.
* **(2 points)** A destructor that cleans up all memory used by the object.
* **(2 points)** Method Size(), that returns the size of the list.
* **(3 points)** Overloaded operator [] for reading (not modifying) list elements, with bounds checking.
* **(2 points)** The appropriate declarations to disable assignments or copies of a fixed\_list object.
* **(5 points)** Overloaded stream output so that a fixed\_list object can be placed onto an output stream.

To test your implementation, you may use main.cc (see the git repository for class) which includes list.hh and executes an input file with commands to modify and display the list. Example inputs, and their corresponding outputs, are given in the git repository. Submit your source files, list.hh and list.cc, in Canvas.