

C++ Review Exercise 1 (C++ OOP and Sorting)

Topics: C++ classes and objects

Goals: Upon successful completion of this tutorial you should be able to:

1. Create and sort your own list class using an array
2. Create and sort your own list class using 1 linked list

Related text sections:

C++ Review Exercise 1 Instructions

Next week we will start working with the object oriented language, Java.

For this tutorial, as a review of C++ OOP, you are to implement a template container class called List. List should use dynamic memory allocation.

Your class should contain the following member or friend methods:

1. A default constructor: List(). The constructor should create an empty list.
2. A destructor: ~List(). The destructor should release all dynamically allocated memory space.
3. A copy constructor: List (const List & other). This constructor should create a new list that is a deep copy of the other list passed as a parameter.
4. An assignment operator: List & operator = List (const List & other). This operator should release space allocated to the current list object the create a list that is a deep copy of the other list passed as a parameter.
5. A sorting function: void Sort (SortType ad). SortType should be defined as an enumerated type containing the values ASCENDING and DESCENDING. When the passed value is ASCENDING, the values in the list should be sorted into ascending order using the simple sort you selected for Tutorial 2 (Exchange, Insertion, or Selection). When the passed value is DESCENDING, the values in the list should be sorted into descending order using the advanced sort you selected for Tutorial 2 (Merge or Quick).
6. An input operator: >>. The input operator should read from a character stream. Values should be placed in the list as they are read. A value of a non-compatible type or the end-of-file marker should terminate the input process. White space should not be included in the list.
7. An output operator: <<. The output operator should write the values in the list to a character stream. Consecutive output values should be separated by a single space character.

A driver program (called main.cpp) for testing your List class can be found in ~tiawatts /cs460pickup.

Your completed template file should be placed in ~tiawatts/cs460drop. The file should be called *your-last-name.cpp*.