[**Department of Computer Science**](http://www.cs.umd.edu/)

[**CMSC132 Fall 2016**](http://www.cs.umd.edu/class/fall2016/cmsc132-04XX/)

**Project:** Linked Lists

**Due Date:** Friday Oct 21, 11:00 pm

## Overview

For this project you will implement a basic linked and a sorted linked list. Notice that no public tests are part of this project as all tests are either release or secret. **You must implement this project by yourself.**

## Objectives

This project will allow you practice linked lists and testing.

## Grading

* (20%) Release Tests
* (53%) Secret Tests
* (15%) Student Tests
* (12%) Style

## Code Distribution

The project's code distribution is available by checking out the project named LinkedListsProject. The code distribution provides you with the following:

* **listClasses package** → Where classes implementing linked lists will appear.
* **tests package**  → Where you should place your student tests.

## Specifications

* You are expected to implement two classes: BasicLinkedList and SortedLinkedList.The javadoc describing what you need to implement can be found at [Project Javadoc](http://docs.google.com/doc/index.html).
* The linked list in our project has a head and tail reference. The next link of the last node points to null.
* Methods addToEnd(), getLast(), retrieveLastElement() rely (and must use) the tail reference.
* The head and tail must be set correctly when list operations update the list.
* You can use recursion for the implementation of methods.
* Two methods must be implemented using recursion: getReverseArrayList() and getReverseList(). You are allowed to add an auxiliary function (only one) during the implementation of a recursive method. You MAY NOT add any instance variables nor static variables to support the implementation of a recursive method. **If you do you will lose significant credit**.
* Your code must be efficient. For example, you will lose credit if you perform an unnecessary list traversal(s) during the implementation of a method.
* If you see in the submit server the error "missing\_component", the problem is that you are missing a required class. You can also get this error if you are not defining a generic class correctly. For example, your SortedLinkedList class is not being defined as expected. It is recommended that you try submiting your project immediately (if you have not done so yet) so you can identify this problem (your code may work in Eclipse but not in the submit server).

## Requirements

* Your student tests must include at least a test for each method of each class. Write tests as you develop your code and not at the end; this is the correct approach to develop student tests. Also, think how you can test your code so it will work with any data set. Hint: For a list, you can try inserting and deleting all the permutations for a set of numbers. This kind of hint is what you need to discover for future projects (and for your future job) by yourself.
* If you have a problem with your code and need assistance during office hours, you need to have a student test(s) that illustrates the problem you are experiencing. See [Student Tests](http://www.cs.umd.edu/class/fall2016/cmsc132-04XX/content/resources/StudentTests.html) for information regarding the implementation of student tests.
* You may not shared your student tests with other students.
* The iterator you define may NOT be used to implement other list operations (e.g., add). The iterator is for users that want to iterate over the list.
* You need to implement a singly-linked list that relies on a head and tail reference.
* See [Style Guidelines](http://www.cs.umd.edu/class/fall2016/cmsc132-04XX/content/resources/StyleGuidelines.html) for information regarding style.

## Honor's Section Requirement

The following requirement is for students enrolled in the honor's section.

* Implement the iterator's remove method for the BasicLinkedList class.
* Add a clone method to the BasicLinkedList class.

## Sample Driver

The following driver and associated output provide an idea of the classes you need to implement.

import listClasses.BasicLinkedList;  
import listClasses.SortedLinkedList;  
  
public class SampleDriver {  
  
 public static void main(String[] args) {  
 BasicLinkedList<String> basicList = new BasicLinkedList<String>();  
  
 basicList.addToEnd("Red").addToFront("Yellow").addToFront("Blue");  
 System.out.println("First: " + basicList.getFirst());  
 System.out.println("Last: " + basicList.getLast());  
 System.out.println("Size: " + basicList.getSize());  
 System.out.println("Retrieve First: " + basicList.retrieveFirstElement());  
 System.out.println("Retrieve Last: " + basicList.retrieveLastElement());  
 System.out.println("Removing Red");  
 basicList.remove("Red", String.CASE\_INSENSITIVE\_ORDER);  
 System.out.print("Iteration: ");  
 for (String entry : basicList) {  
 System.out.print(entry + " ");  
 }  
   
 SortedLinkedList<String> sortedList = new SortedLinkedList<String>(String.CASE\_INSENSITIVE\_ORDER);  
 sortedList.add("Yellow").add("Red");  
 System.out.print("\n\nIteration (for sorted list): ");  
 for (String entry : sortedList) {  
 System.out.print(entry + " ");  
 }  
 sortedList.remove("Red");  
 System.out.print("\nAfter remove in sorted list first is: ");  
 System.out.println(sortedList.getFirst());  
 }  
}  
  
Output  
  
First: Blue  
Last: Red  
Size: 3  
Retrieve First: Blue  
Retrieve Last: Red  
Removing Red  
Iteration: Yellow   
  
Iteration (for sorted list): Red Yellow   
After remove in sorted list first is: Yellow

## Submission

* Submit your project using the "Submit Project" option (available by right clicking on the project folder).

## Academic Integrity

Please make sure you read the academic integrity section of the syllabus so you understand what is permissible in our programming projects. We want to remind you that we check your project against other students' projects and any case of academic dishonesty will be referred to the [Office of Student Conduct](http://www.jpo.umd.edu/).