**CSCI 3302 Pair Programming Assignment 03 (100 Points)**

**Due: Oct 6, 8:00 AM**

**GitHub Link:** [**Pair Program 03**](https://classroom.github.com/a/gKrkHAtk)

**Objectives:**

* Demonstrate understanding of generics.
* Demonstrate how to implement Linked List.

**Assignment Assistance:**

* This homework assignment is due before the date and time specified above.
* You and your partner must work on this TOGETHER. One person is the DRIVER, the other is the NAVIGATOR. If you are unfamiliar with paired programming, please google and get acquainted. Here is a quick video with an overview: <https://youtu.be/q7d_JtyCq1A>
* This assignment is restricted to you and your partner. You may not receive help from any other person except the instructor or the AARC (help from the AARC must be well documented!).
* Any resource used (other than Dr. Becnel or the course text) must be documented in the code (as comments) detailing the source and describing exactly what was learned and how that information was used. Submissions will be severely penalized if copied in part or in whole from any source.
* If you need help, visit your instructor during his posted office hours. If your schedule cannot accommodate any of these times, then email your instructor to schedule a different time.

**Problem Description:**

1. After getting access to the repository, you will notice there are a lot of files. This is common for a junior developer starting with a company. But don’t worry….you only need to create one file: GenericList.java.
   1. **GenericList** – a class that implements a linked list for a specified data type. This is the class you will implement.
   2. ImageManipulator – the class you implemented in your first paired programming assignment to manipulate an image
   3. ImageDisplay – this file contains the main method which runs the entire programming. Use this when you are ready to see the result of your work. *You should not edit this file (outside of possibly adding printlns to help with testing).*
   4. TestGenericList – this is a simple test program that you can use to test methods in the GenericList file. Use (and change if you like) this file when testing.
2. You should start by understanding the code you were given. In particular, notice a few things in ImageManipulator:
   1. Node.java – this is the Node class that uses Generics. You should be familiar with this file from our work in class.
   2. ListADT.java – this is the abstract class containing the methods (see below) you will implement in GenericList.java.
3. In this assignment, you will implement a class called GenericList from the abstract class, ListADT<E>. The interface is defined as follows:

public abstract class ListADT<E> {

    public abstract boolean isEmpty();

    public abstract int size();

    public abstract void removeAll();

    public abstract void add(int index, E item) throws ListException;

    public abstract E get(int index) throws ListException;

    public abstract void remove(int index) throws ListException;

}

From our work in class, you should have a good understanding of how these methods should be implemented. In particular, consult your notes on FriendsList.java as this should be very similar.

You should also override toString() for testing purposes.

1. Add appropriate comments to all your methods. You may consider using /\*\* for the Java doc comment. You should also add an appropriate file heading to the file which includes both your name and your partner's name.
2. Your submission should compile and run with errors.
3. You may write any private helper methods if needed. In class, we used a private method getNodeAt(int index).

**Hints:**

* To start testing your methods use TestGenericList.java file. As you implement methods, uncomment the corresponding portions of the file to test. Note: This file is not meant to be a complete test. After this file is working well, try ImageDisplay.java to see if the buttons along the top work.

**Submission:**

* Review the Evaluation below to ensure you have met all the requirements.
* Only one person needs to turn in/commit the assignment. However, both members of the team are expected to understand the solutions and be able to answer questions about the solution.
* Commit all files to GitHub. Upload a backup copy to D2L. You may include testing files in your repository; however, these will not be considered when grading. If you wish to include non-working code for insight into your thought process, make sure to contain it within comment blocks and ensure that submission successfully compiles/runs.

**Evaluation**

|  |  |
| --- | --- |
| **Automatic Deductions:** |  |
| Late/Not Submitted | -100 |
| Code not submitted to GitHub | -30 |
| Code does not run/compile | -50 |
| GenericList does not use LinkedList methodology / Node (e.g. uses Arrays or ArrayList instead of Node) | -40 |
|  |  |
| **Earn Points for the following:** |  |
| Code has a comment header with name, section, date | 5 pts |
| Code organization, structure, and indention is appropriate (SHFT + ALT + F in VS Code) | 5 pts |
| Code is well and meaningfully commented. | 5 pts |
| Appropriate variable and method names that follow Java conventions | 5 pts |
| GenericList uses a generic data type and can hold any Object in the list. (e.g. TestGenericList and ImageDisplay should operate correctly with no need to change the GenericList class). | 15 pts |
|  |  |
| Constructor correctly sets fields | 5 pts |
| isEmpty correctly implemented | 5 pts |
| size correctly implemented | 5 pts |
| removeAll correctly implemented | 5 pts |
| add method correctly implemented with correct exception handling | 15 pts |
| remove method correctly implemented with correct exception handling | 15 pts |
| get method correctly implemented with correct exception handling | 15 pts |