**Project #1**

**Due Dates: Saturday, September 24 at 11:59pm**

**Submit: eLearning**

**Late Policy: 24-hour late period, then zero**

**Instructions: This is an individual assignment. Answers should be your own work.**

**Introduction:**

**In this project you will create a generic linked list using**

**Java Generics.**

**Description:**

**Create a generic class called GenLinkedList. GenLinkedList will use nodes**

**that store a value of the generic type to store its contents.**

**It should have the following methods. The methods should**

**all operate on the object making the call (none are static).**

**Perform checking of the parameters and throw exceptions where**

**appropriate.**

**The linked list should be singly-linked.**

**It should not use sentinel nodes (empty header and tail nodes).**

**You should strive for an efficient implementation of each method.**

**7 points each (a-h)**

**a. addFront**

**receives an item to add as a parameter, and adds to the front of the list.**

**b. addEnd**

**receives an item to add as a parameter, and adds to the end of the list.**

**c. removeFront**

**removes a node from the front of the list.**

**d. removeEnd**

**removes a node from the end of the list.**

**e. set**

**receives a position and item as parameters, sets the element at this**

**position, provided it is within the current size**

**f. get**

**receives a position as a parameter, returns the item at this position, provided it is within the current size**

**g. swap**

**receives two index positions as parameters, and swaps the nodes at**

**these positions (not just the values inside the nodes), provided**

**both positions are within the current size**

**h. shift**

**receives an integer as a parameter, and shifts the list forward or**

**backward this number of nodes, provided it is within the current size**

**1,2,3,4,5 shifted +2 3,4,5,1,2**

**1,2,3,4,5 shifted -1 5,1,2,3,4**

**11 points each (i-l)**

**i. removeMatching**

**receives a value of the generic type as a parameter and removes all**

**occurrences of this value from the list.**

**j. erase**

**receives an index position and number of elements as parameters, and**

**removes elements beginning at the index position for the number of**

**elements specified, provided the index position is within the size**

**and together with the number of elements does not exceed the size**

**k. insertList**

**receives a generic List (a Java List) and an index position as parameters, and copies each value of the passed list into the current list starting at the index position, provided the index position does not exceed the size. For example, if list has a,b,c and another list having 1,2,3 is inserted at position 2, the list becomes a,b,1,2,3,c**

**l. main**

**add code to the main method to demonstrate each of your methods**

**Submit to eLearning:**

**GenLinkedList.java**