# Class Exercise

## ArrayList

Q1. Write a program that reads a file and displays the words of that file as a list.

- First display all words. !
- Then display them in reverse order. !
- Then display them with all plurals (ending in "s"). !
- Then display them with all plural words removed and the size of updated list.

## LinkedList

Q2. Suppose the list letters contains elements "A", "B", "C", and "D". Draw the contents of the list and the iterator position for the following operations:

```
ListIterator<String> iter = letters.iterator();
iter.next();
iter.next();
iter.remove();
iter.next();
iter.add("E");
iter.next();
iter.add("F");
```

Q3. Explain what the following code prints. Draw a picture of the linked list and the iterator position after each step.

```
LinkedList<String> staff = new LinkedList<String>();
ListIterator<String> iterator = staff.listIterator();
iterator.add("Tom");
iterator.add("Diana");
iterator.add("Harry");
iterator = staff.listIterator();
iterator.next();
iterator.next();
iterator.add("Romeo");
iterator.next();
iterator.add("Juliet");
iterator = staff.listIterator();
iterator.next();
iterator.remove();
while (iterator.hasNext()) {
        System.out.println(iterator.next());
}
```

## Stack & Queue

4. Use a stack to reverse the words of a sentence. Keep reading words until you have a word that ends in a period, adding them onto a stack. When you have a word with a period, pop the words off and print them. Stop when there are no more words in the input. (20 mins)

**Definition of Done**

i. You should the convert the input as follows:

*Mary had a little lamb. Its fleece was white as snow.*

into

*Lamb little a had mary. Snow as white was fleece its.*

ii. Display the desired resultant stack

5. A homeowner rents out parking spaces in a driveway during special events. The driveway is a "last-in, first-out" stack. Of course, when a car owner retrieves a vehicle that wasn't the last one in, the cars blocking it must temporarily move to the street so that the requested vehicle can leave. Write a program that models this behavior, using one stack for the driveway and one queue for the street. Use integers as license plate numbers to identify an individual car. (30 mins)

**Definition of Done**

i. The program uses one stack for "driveway" and one queue for "street and uses license plate numbers to represent individual cars

ii. The user inputs the car license number to retrieve a car

iii. Display the resultant driveway and street after every car moves out

6. Implement a to do list. Tasks have a priority between 1 and 9, and a description. When the user enters the command add priority description, the program adds a new task. When the user enters next, the program removes and prints the most urgent task. The quit command quits the program. Use a priority queue in your solution. (30 mins)

**Definition of Done**

i. Assign priorities b/w 1 to 9 to all the tasks

ii. Tasks should be added and removed according to the user input

iii. Finally display the resultant queue

**Set**

7. Write a program that implements a simple spell checker using a hash table. Your spell-checker will be reading from two input files.

The first file is a dictionary "dictionary.txt". The program should read the dictionary and insert the words into a hash table. After reading the dictionary, it will read a list of words from a second file. The goal of the spell-checker is to determine the misspelled words in the second file by looking each word up in the dictionary. The program should output each misspelled word.

**Definition of Done:**

i. Program should read two files namely, “dictionary.txt” and “words.txt”
ii. It should check the mis-spelled words given in “words.txt” and display such words
iii. Use the appropriate Collection that implements the hashtable for storing items

**Map**

8. Write a program that keeps a map in which both keys and values are strings—the names of students and their course grades. Prompt the user of the program to add or remove students, to modify grades, or to print all grades. The printout should be sorted by name and formatted like this:

Carl: B+

Joe: C

Sarah: A

**Definition of Done**

i. Define a class that implements Map interface and take String attributes

ii. Take the input from the user to add/remove/modify the records

iii. Display the resultant Map as formatted in the question

9. Write a program that reads a text file and prints a list of all words in the file in any order, together with a count that indicates how often each word occurred in the file.

For example, the following is the beginning of the output that results from processing the book Alice in Wonderland:

| WORD | FREQUENCY |
|---|---|
| abide | 1 |
| wander | 28 |
| absurd | 2 |
| decide | 54 |
| alice | 653 |
| dream | 168 |
| about | 97 |

**Definition of Done**

i. The program should read the text from the file namely “alice_in_wonderland.txt”
ii. Use appropriate Map class to get the word and its count
iii. Should individual word along with its frequency as the final output