CSCI 1101
Computer Science II
PRACTICE SET FOR TEST NO.2
SOLUTIONS

Question 1. For each of the following questions, select the most appropriate answer.

1. If the following operations are made in succession to an empty ArrayList of Strings called myList,

```
myList.add(0,"A");
myList.add("B");
myList.remove(1);
myList.set(0,"B");
myList.add("A");
```

myList will have the following values:

   a. A B
   b. B A
   c. A A
   d. B B
   e. none of the above

2. When you make the following declaration:

```
ArrayList<String> myList = new ArrayList<String>(2000);
```

   a. myList will have initially zero locations but will increase to 2000 after one element is added.
   b. myList will have initially 2000 locations and will increase when the limit is reached.
   c. You will be able to store only 2000 elements in myList.
   d. No matter what the initial capacity is, the ArrayList will keep incrementing in sizes of 10.

3. The following is the correct way to declare an arraylist of integer objects.
   a. `ArrayList<int> list = new ArrayList<int>;`
   b. `ArrayList<Integer> list = new ArrayList<Integer>;`
   c. `ArrayList<int> list = new ArrayList<int>();`
   d. `ArrayList<Integer> list = new ArrayList<Integer>();`

4. What is the output of the following code?

```
ArrayList<String> cities = new ArrayList<String>();
cities.add("Halifax");
cities.add("Montreal");
int i = 1;
cities.add(i, "+");
System.out.println(cities);
```

   a. [Halifax, Montreal, +, +]
   b. [Halifax, +, Montreal]
   c. [+, Montreal, Halifax]
   d. [Halifax, Montreal, +]
   e. [Montreal, +, Halifax, +]

5. The following code removes the front node in a linked list (assuming that the list has at least one node):
   a. front = front.getNext(); count--;
   b. front.setNext(null);count--;
   c. front = front.getNext().getNext();count--;

d. front.setNext(front.getNext());count--;

6. Analyze the following code:

```
ArrayList<String> list = new ArrayList<String>();
list.add("Beijing");
list.add("Tokyo");
list.add("Shanghai");
list.set(3, "Hong Kong");
```

This code causes a runtime error.
Which of the following is correct?

a. If you replace the last line by list.add(3, "Hong Kong"), the code will compile and run fine.
b. If you replace the last line by list.add(4, "Hong Kong"), the code will compile and run fine.
c. If you replace the last line by list.add(5, "Hong Kong"), the code will compile and run fine.
d. If you replace the last line by list.set(4, "Hong Kong"), the code will compile and run fine.

7. If an arraylist `a` of Strings contains the following items:
`[emboar, palkia, emboar]`
what will the arraylist look like after the following statements?

```
a.add("pikachu");
a.add("rayquaza");
a.set(3,"dialga");
a.remove("emboar");
```

a. [pikachu, rayquaza, palkia, dialga]
b. [palkia, emboar, pikachu, dialga]
c. [palkia, dialga, pikachu, rayquaza]
d. [palkia, emboar, dialga, rayquaza]

8. A method in the LinkedList class that attaches a Node `n` with String `data` to the front of the linked list has the following statements:

```
a. Node n = new Node(data, front.getNext()); front = n; count++;
b. Node n = new Node(data, front); front = n; count++;
c. Node n = new Node(data, front.getNext()); front.setNext(n);
   count++;
d. Node n = new Node(data, front); front.setNext(n); count++;
```

NOTE: For questions on ArrayLists, a reference sheet with method names and meanings will be provided.

Question 2: Create a class called Course that has the following specification:
name of the course
names of students in the course (an arraylist).
constructor that sets the name of the course and other attributes to default values.
method to add a student to the course
method to print the names of all students enrolled in the course

```
import java.util.ArrayList;
public class Course
{
        private String name;
        private ArrayList<String> students;
        private int num;
```

```
        public Course(String n)
        {
                name = n;
                students = new ArrayList<String>();
        }
        public void add(String s){
                students.add(s);
        }
        public void display(){
                System.out.println(students);
        }
}
```

Question 3: An ArrayList called `names` has been constructed to store Strings. Write java statements to perform the following:

a. Read a number of Strings (one per line, ending with a *) into the arraylist, storing each String at the beginning of the list. (in reverse order).
b. Count the number of times that the String "`Java`" occurs in the arraylist and print the number.
c. Remove all occurrences of the String "`Java`" from the arraylist.

a.

```
System.out.print("Enter a word (* to end): ");
String s = keyboard.nextLine();
while (!s.equals("*"))
{
        names.add(0,s);
        System.out.print("Enter a word(* to end): ");
        s=keyboard.nextLine();
}
```

b.

```
int count = 0;
for(int i = 0; i< names.size(); i++)
        if (names.get(i).equals("Java"))
                count++;
System.out.println(count);
```

c.

```
for(int i = names.size()-1; i>=0; i--)
        if (names.get(i).equals("Java"))
                names.remove(i);
```

Question 4: Write a static method exchange (ArrayList<String> a, int k, int j) that swaps the kth and the jth elements in an ArrayList a.

```
public static void exchange(ArrayList<String> a, int k, int j)
{
        String temp;
        if (k< a.size() && k >=0 && j<a.size() && j >=0)
        {
                temp = a.get(k);
                a.set(k, a.get(j));
                a.set(j, temp);
        }
}
```

Question 5: Write a static method that returns an arraylist that is the concatenation of two arraylists a and b.

```
public static ArrayList<String> concatenate(ArrayList<String> a, ArrayList<String> b)
{
        ArrayList<String> c = new ArrayList<String>();
        for(int i = 0; i < a.size(); i++)
                c.add(a.get(i));
        for(int i = 0; i< b.size(); i++)
                c.add(b.get(i);
        return c;
}
```

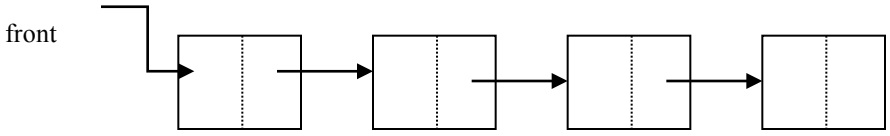
Question 6: Write a static method that returns the index of the smallest integer in an arraylist of integers.

```
//assumes that a has atleast one element
//if the smallest integer is repeated, it returns the index of the first occurrence.
public static int indexOfSmallest(ArrayList<Integer> a)
{
        int smallest = a.get(0);
        int index = 0;
        for(int i = 1; i<a.size(); i++)
        {
                if (a.get(i)< smallest)
                {
                        smallest = a.get(i);
                        index = i;
                }
        }
        return index;
}
```

Question 7: Write a static method that takes in two arrays of integers and returns an arraylist containing the integers of the first array in the same order and the integers of the second array in the reverse order. For example, if the input arrays are {1, 2, 3, 4} and {5, 6, 7, 8} the resulting arraylist must be
[1, 2, 3, 4, 8, 7, 6, 5]

```
public static ArrayList<Integer> convert(int[] a, int[] b)
{
        ArrayList<Integer> result = new ArrayList<Integer>();
        for(int i=0;i<a.length;i++)
                result.add(a[i]);
        for(int i = b.length-1; i >=0; i--)
                result.add(b[i]);
        return result;
}
```

Note: For questions on Linked Lists, a reference sheet with basic method headers will be given.
Question 8: Assume the availability of the Node class and LinkedList class. Let front be the reference to the first node in the linked list. Assume that the LinkedList stores Strings.



Write java code segments to do the following operations:

a) Display the Strings in the first node and the last node.
b) Print the Strings in the even numbered nodes (0, 2, 4, 6, etc.)
c) Count the total number of characters of all the Strings.

a)
```
if (front==null)
        return;
else
{
        System.out.println(front.getData());

        Node curr = front;
        while (curr.getNext()!=null)
                curr= curr.getNext();
        System.out.println(curr.getData());
}
```

b)
```
if (front==null)
        return;
```

```
else
{
        Node curr= front;
        int i = 0;
        while (curr!=null)
        {
                if (i%2==0)
                        System.out.println(curr.getData());
                i++;
                curr= curr.getNext();
        }
}
```

c)

```
Node curr = front;
int count=0;
while (curr!=null)
{
        count+=curr.getData().length();
        curr = curr.getNext();
}
System.out.println("Total number of characters in all the strings is: " + count);
```

Question 9: Add the following methods to the Linked List class:

a) Add a node with a given String s after the front node. If the list is empty, just return.
b) Delete the last node in the linked list.
c) Delete the second last node (the last but one) in the linked list. If the list has zero or one node, just return.

a)

```
public void addAfterFirstNode(String s)
{
        if (front==null)
                return;
        else
        {
                Node n;
                n = new Node(s, front.getNext());
                front.setNext(n);
                count++;
        }
}
```

b)

```
public void removeLast()
{
   if (front==null)
   {
            return;
   }
   else if (front.getNext()==null)
   {
            front = null;
            count--;
   }
   else
   {
            Node curr = front;
            while (curr.getNext().getNext()!=null)
                     curr = curr.getNext();
            curr.setNext(null);
            count--;
   }

}
```

c)

```
public void removeLastButOne()
{
            if (count==0) return;
            else if (count==1) return;
            else if (count==2) front = front.getNext();
            else
            {
               Node curr = front;
               while (curr.getNext().getNext().getNext()!=null)
                        curr = curr.getNext();
               curr.setNext(curr.getNext().getNext());
            }
            count--;
}
```

Question 10: What is the output of the following program?

```
import java.util.ArrayList;

public class Demo1
{
 public static void main(String[] args)
        {

                ArrayList<String> list = new ArrayList<String>();

                list.add("A");
                list.add(0,"B");
                list.add(0,"C");
                list.add(2,"D");
                list.set(3,"E");
                list.remove(0);
                System.out.println(list.indexOf("D"));
                System.out.println(list);

        }
}
```

Output:

1

[B, D, E]

Question 11: What is the output of the following code?

```
public class Q4B
{
        public static void main(String[] args)
        {
                LinkedList names = new LinkedList();
                names.addToFront("Java");
                names.addToFront("C");
                names.addToFront("C++");
                names.addToFront("Python");
                System.out.println(names.size());
                names.enumerate();
                names.clear();
                names.addToFront("Pascal");
                names.addToFront("Visual Basic");
                System.out.println(names.size());
                names.enumerate();

        }
}
```

```
----jGRASP exec: java -ea Q4B
4
Python-->C++-->C-->Java-->
2
Visual Basic-->Pascal-->
----jGRASP: operation complete.
```

Question No. 12: What is the output of the following code?

```
import java.util.ArrayList;
public class Q4C {
   public static void main (String args[]){
      ArrayList <String> aList = new ArrayList<String>();
      String temp;
      aList.add("AB");
      aList.add("CD");
      aList.add("EF");
      aList.add(0,"UV");
      aList.add(0,"WX");
      aList.add(0,"YZ");
      LinkedList nList = new LinkedList ();
      for (int i=aList.size()-1; i>=0;i--)
      {
         if (i%2==0)
         {
            temp = aList.get(i);
            nList.addToFront(temp);
            aList.remove(i);
         }
      }
      System.out.println(aList);
      nList.enumerate();
   }
}
```

```
----jGRASP exec: java -ea Q4C
[WX, AB, EF]
YZ-->UV-->CD-->
----jGRASP: operation complete.
```