Points off	1	2	3A	3B	4		Total off	Net Score

CS 307 – Midterm 1 – Spring 2010

Your Name_____

Your UTEID _____

Circle yours TA's name: Guillermo Reza Shyamala

Instructions:

1. There are **4** questions on this test.
2. You have 2 hours to complete the test.
3. You may not use a calculator or any other electronic devices while taking the test.
4. When writing a method assume the preconditions of the method are met.
5. When writing a method you may add helper methods if you wish.
6. When you complete the test show the proctor your UTID and give them the test and any scratch paper. Please leave the room quietly.

1. (2 points each, 30 points total) Short answer questions. Place your answers on the attached answer sheet. For code sample state the output. If the code would cause a syntax error answer "syntax error". If it would cause a runtime exception answer "exception". If it would result in an infinite loop answer "infinite loop".

Questions A and B use the following method.

```
public void helper(int[] vals){
     vals[vals.length - 1] += 3;
     vals = new int[2];
     vals[0]++;
}
```

A. What is output by the following code when method `a` is called?

```
public void a(){
     int[] data = {2, 3};
     helper(data);
     System.out.println(Arrays.toString(data));
}
```

B. What is output by the following code when method `b` is called?

```
public void b(){
     int[] data = new int[0];
     helper(data);
     System.out.println(Arrays.toString(data));
}
```

For questions C – M consider the following classes and interfaces. Recall, the `Comparable` interface has one method, `compareTo` which returns an `int`.

```
public abstract class Property implements Comparable{
      private String name;

      public abstract int getValue();

      public Property(String str){  name = str;  }

      public String toString(){     return name;  }

      public int compareTo(Object other){
            return getValue() - ((Property)other).getValue();
      }
}

public class Business extends Property{
      private int area;

      public Business(String str, int a){
            super(str);
            area = a;
      }

      public int getValue(){  return area * 100; }
}

public class Residential extends Property {
      private int rooms;

      public Residential(int r){
            super("Res");
            rooms = r;
      }

      public int getValue(){  return rooms * 1000;  }

      public void addOn(){    rooms++;    }
}

public class Rented extends Residential {
      public Rented(int r){   super(r);  }

      public int getValue(){ return super.getValue() / 2; }
}

public class Owned extends Residential {
      private int yardArea;

      public Owned(int y){    this(y, 4);  }

      public Owned(int y, int r){
            super(r);
            yardArea = y;
      }

      public String toString(){    return "" + yardArea;  }
}
```

C. State if the following declarations are valid or invalid (meaning they cause a syntax error). 1 point each

```
Object obj1 = new Rented(3);
Object obj2 = new Property("Gym");
```

D. State if the following declarations are valid or invalid (meaning they cause a syntax error). 1 point each

```
Comparable c1 = new Comparable();
Comparable c2 = new Owned(100);
```

E. State if the following declarations are valid or invalid (meaning they cause a syntax error). 1 point each

```
Owned w = new Residential(5);
Business b = new Owned(100);
```

F. State if the following declarations are valid or invalid (meaning they cause a syntax error). 1 point each

```
Property p1 = new Rented(3);
Property p2 = new Business("7-11", 100);
```

G. What is output by the following code?

```
Owned o1 = new Owned(100);
Owned o2 = new Owned(100);
System.out.println(o1 == o2);
```

H. What is output by the following code when method `h2` is called?

```
public void h1(Residential r){
     r.addOn();
     r.addOn();
}

public void h2(){
     Residential r1 = new Residential(4);
     h1(r1);
     System.out.println(r1.getValue());
}
```

I. What is output by the following code when method `i2` is called?

```
public void i1(Business b){
     b = new Business("Mac's", 100);
}

public void i2(){
     Business b1 = new Business("J's", 200);
     i1(b1);
     System.out.println(b1);
}
```

J. What is the output by the following code?

```
Owned o3 = new Owned(100);
System.out.print(o3 + " ");
System.out.println(o3.getValue());
```

K. What is the output by the following code when method `k2` is called?

```
public void k1(Residential r){
     System.out.print(r.getValue() + " ");
}

public void k2(){
     Rented d = new Rented(4);
     k1(d);
     Residential s = new Residential(4);
     k1(s);
}
```

L. What is the output by the following code?

```
Property[] ps = new Property[2];
ps[0] = new Business("JP", 100);
ps[1] = new Residential(5);
System.out.println( ps[0].compareTo(ps[1]) );
```

M. Explain in one sentence why the following client code contains a syntax error.

```
Owned o5 = new Owned(100, 5);
System.out.println(o5);
o5.rooms = 7;
System.out.println(o5.getValue());
```

N. Recall the following methods from the `IntList` class we developed in lecture.

```
public class IntList{
     public IntList() // create an empty IntList

     public int add(int val) // add to end
     public int insert(int pos, int val) // add at specified location
     public int size() // return size of list

     public String toString();
     // returns String of the form [val1, val2, ..., lastVal]
}
```

What is output by the following code?

```
IntList list = new IntList();
System.out.print( list.size() + " " );
list.add(12);
list.insert(0, 5); // parameters are position, value
System.out.print( list );
```

O. What is returned by the following method if the file `/bin/scores.tx` does not exist?

```
public int countTokens() {
     int result = 0;
     try {
          Scanner sc = new Scanner(new File("/bin/scores.tx"));
          while(sc.hasNext()){
               sc.next();
          }
          result = 5;
     }
     catch(FileNotFoundException e){
          result = -1;
     }
     return result;
}
```

2. The `GenericList` class. (25 points) In lecture we developed a `GenericList` class to represent a list of objects to demonstrate encapsulation and the syntax for building a class in Java. As discussed in class the internal storage container may have extra capacity.

Complete a method for the `GenericList` class named `addNewValues`. This is an instance method in the `GenericList` class that adds all the elements from another `GenericList` that were not present in the original list to the end of the original list. If a value appears multiple times in the other `GenericList` but was not present in the original, the multiple values will all be added. (See examples below.)

Here is the method header:

```
/*
pre: other != null

post: all elements in other that were not present in this GenericList
before the method call are added to the end of this list. The relative
order of the added items is the same as they appear in other. The
GenericList other is not altered as a result of this method.
No elements in either GenericList equal null.
*/
public void addNewValues(GenericList other) {
```

Examples of calls to `addNewValues`: (Assume values shown are Strings.)

```
[A, B, A, C].addNewValues( [E, A, D] ) -> [A, B, A, C, E, D]

[].addNewValues( [A, A, B, A] ) -> [A, A, B, A]

[A, Z, S].addNewValues( [B, A, T, A, B, B] ) ->
     [A, Z, S, B, T, B, B]

[A, D, B].addNewValues( [] ) -> [A, D, B]

[A, X, S, X, T].addNewValues( [X, X, A, S, S, T, A] ) -> [A, X, S, X, T]

[A, B, C, B].addNewValues( [B, C, X, A, X] ) -> [A, B, C, B, X, X]

[A, B, C, B].addNewValues( [X, C, B, A, Z] ) -> [A, B, C, B, X, Z]
```

You may not use any other methods in the `GenericList` class except for the `resize` method unless you define and implement them yourself in this question. Recall that this method is in the `GenericList` class so you have access to all `GenericList` objects' private instance variables.

You may not use objects or methods from other Java classes other than native arrays and the provided `GenericList` method. If you want to use any other `GenericList` methods you must define and implement them yourself.

You are not required to check the preconditions of the method and when you write the method you may assume the preconditions of the method are met.

```
public class GenericList{

     private Object[] container;
     private int listSize;

     // pre: none
     // post: container refers to an array that is twice as large
     // as the previous array. All elements have been copied over.
     // container will refer to an array with a size of at least 1.
     private void resize()

}

/*
pre: other != null

post: all elements in other that were not present in this GenericList
before the method call are added to the end of this list. The relative
order of the added items is the same as they appear in other. The
GenericList other is not altered as a result of this method.
No elements in either GenericList equal null.
*/
public void addNewValues(GenericList other) {
     // do not write code to check the precondition









// More room on next page
```

3. (25 points total) This question has two parts. It involves the `MathMatrix` class from assignment 3 and a new way of representing a mathematical matrix.

3A. (5 points) If most of the values in a matrix are the same value, usually 0, the matrix is said to be *sparse*. If most of the values in the matrix are equal to 0 it may be possible to save time and space by not using a 2d array of ints to represent the matrix. Instead, only the values that are not equal to zero are stored. If this is done the positional data about each non-zero element must also be stored. For example, consider the following sparse matrix.

2	0	4	0	0
0	2	0	0	0
0	0	0	1	0
0	0	5	0	2

The same matrix could be represented as follows: (Each set of numbers represents the row, column, and value for all elements in the matrix that are not equal to zero.

[(0,0,2), (0,2,4), (1,1,2), (2,3,1), (3,2,5), (3,4,2)]

As you can see if most of the elements are zero it may be more efficient in time and space to only store the elements that not equal to 0 with their positional data.

For part A explain what instance variables you would use for a `SparseMatrix` class. If you feel you would need to implement any other classes explain those as well.

3A. (20 points) In part B code the instance variables for your `SparseMatrix` and a constructor that given a rectangular 2d array of ints sets the instance variables of your `SparseMatrix` class correctly for all non zero elements in the 2d array of ints. If you intend to use any non standard Java classes as instance variables write the code for the public intrerface of that class. (The public interface is not the same as a Java `interface`. It is the list of public constructors and methods in the class as shown in question 1.N for the `IntList` class .)

```
public class SparseMatrix{

// instance variables









// pre: values != null, values is rectangular, values has a least
// one row, and at least one column
public SparseMatrix(int[][] values) {









// more room on next page if needed
```

4 Working with objects (20 points total) This question involves the `NameRecord` class from Assignment 4, the NameSurfer assignment.

Complete an instance method in the `NameRecord` class that returns `true` if the `NameRecord` object had a change in popularity between any two consecutive decades greater than or equal to the value of a parameter sent to the method. Here is the method header.

```
/*
pre: none
post: return true if this NameRecord had a change between two consecutive
decades greater than or equal to the value of move. If move is positive
the change has to be an increase in popularity. If move is negative the
change has to be a decrease in popularity.
*/
public boolean hadMove(int move)
```

Here is an example. The ranks for the name Lisa are: `0 0 0 0 464 38 1 6 31 113 298`

Recall the first integer is the rank for the decade 1900 - 1909, the second integer is the rank for the decade from 1910 - 1919, and so forth. Recall a lower number (except for 0's) means a name was more polular. A value of 0 indicates the name had a rank greater than 1000 for the decade. The 0's will be stored as 0's but for this question you may assume a 0 indicates a rank of 1001.

If `nr` refers to a `NameRecord` object that contains the data for the name Lisa here are some result of method calls to `hadMove`.

```
nr.hadMove(500) -> returns true. Change from 0 to 464 was an increase in
popularity of 537 places, 1001 - 464 = 537

nr.hadMove(600) -> returns false. No increase in popularity greater than
or equal to 600 places.

nr.hadMove(-600) -> returns false. The name Lisa never saw a decrease in
popularity of greater than or equal to 600 places.

nr.hadMove(-100) -> returns true. Change from 113 to 298 was a decrease of
185 places.
```

Here is the NameRecord class:

```
public class NameRecord {
     private String name;
     private ArrayList<Integer> ranks;

     /*
     pre: none
     post: return true if this NameRecord had a change between two
     consecutive decades greater than or equal to the value of move. If
     move is positive the change has to be an increase in popularity. If
     move is negative the change has to be a decrease in popularity.
     */
     public boolean hadMove(int move)
```

From the `ArrayList` class:

```
public int size() - returns the number of elements in this ArrayList

public E get(int pos) returns the element at the specified position in
this list. pre: 0 <= pos < size()
```

Complete the `hadMove` method from the `NameRecord` class.

```
public boolean hadMove(int move)
```

Question 1 answer Sheet

Name_____

A. _____

B. _____

1.

2.
C. _____

1.

2.
D. _____

1.

2.
E. _____

1.

2.
F. _____

G. _____

H. _____

I. _____

J. _____

K. _____

L. _____

M. _____

N. _____

O. _____