Points off	1	2	3	4	5		Total off	Net Score

CS 314 – Midterm 1 – Fall 2012

Your Name____________________________________

Your UTEID __________________________________

Circle yours TA's name: John Zihao

Instructions:

1. There are **5** 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 answering coding questions, ensure you follow the restrictions of the question.
7. When you complete the test show the proctor your UTID, give them the test and any scratch paper, and please leave the room quietly.

1. (2 points each, 30 points total) Short answer. Place you answers on the attached answer sheet.
   a. If a question contains a syntax error or other compile error, answer "Compile error".
   b. If a question would result in a runtime error or exception answer "Runtime error".
   c. If a question results in an infinite loop answer "Infinite loop".
   d. Recall when asked for Big O your answer should be the most restrictive correct Big O function. For example Selection Sort has an average case Big O of $O(N^2)$, but per the formal definition of Big O it is correct to say Selection Sort also has a Big O of $O(N^3)$ or $O(N^4)$. I want the most restrictive, correct Big O function. (Closest without going under.)

A. What is the T(N) for method `a`? Recall, T(N) is the function that represents the *actual* number of executable statements for an algorithm. N = `data.length`

```
// pre: data != null, result.length = 5
public void a(int[] data, int[] results) {
     for(int i = 0; i < data.length; i++) {
          int temp = 10;
          for(int j = 0; j < 5; j++) {
               results[j] += data[i] % temp;
               temp *= 10;
          }
     }
}
```

B. What is the order (Big O) of method `a`?

C. What is order (Big O) of method `c`? N = `list.size()`.
Method `check` returns an int and is O(1).

```
public int c(ArrayList<Integer> list){
     int result = 0;
     int limit = list.size();
     for(int i = 0; i < list.size(); i++) {
          result += list.get(i);
          result += check(list, i);
          for(int j = limit - 1; j >= i; j--) {
               result += check(list, j);
               result += check(list, i / j);
          }
          if(result % 2 == 0)
               result--;
          else
               result++;
     }
     return result;
}
```

D. What is the order (Big O) of method `d`? Assume `verify` is O(N) and the `String length` method is O(1). N = `values.size()`

```
public int d(ArrayList<String> values) {
     int c = 0;
     int limit = values.size() / 2;
     for(int i = limit; i < values.size(); i++) {
          c += values.get(i).length();

          String temp = values.remove(i);

          // ArrayList overloads add for inserting
          values.add(0, temp); // position, value

          for(int j = 0; j < values.size(); j++) {
               verify(values);
          }
     }
     return c;
}
```

E. What is the worst case order (Big O) of method `e`? N = `readings.length`

```
public double e(double[] readings, double min) {
     double result = 0.0;
     for(int i = 0; i < readings.length; i++) {
          if(readings[i] > min) {
               for(int j = readings.length - 1; j > 0; j /= 3) {
                    result += readings[j];
                    readings[j] -= 0.5;
               }
          } else {
               result += readings[i];
               readings [i] += 1.5;
          }
     }
     return result;
}
```

F. A method is $O(N^2)$. It takes 3 seconds for the method to run when N = 10,000.
What is the expected time for the method to run when N = 20,000?

G. A method is $O(N\log_2 N)$. It takes 5 seconds for the method to run when N = 1,000,000.
What is the expected time for the method to run when N = 4,000,000?
Assume $\log_2$ 1,000,000 = 20.

H. A method is $O(2^N)$. It takes 10 seconds for the method to run when N = 100.
What is the expected time for the method to run when N = 105?

I. What is output by the following code segment?

```
ArrayList<String> demo = new ArrayList<String>();
demo.add("A");
demo.add(0, "BB");
demo.add("C");
demo.add("D");
demo.add(1, "EE");
demo.add(demo.remove(3));
demo.add(2, demo.get(demo.size() - 1));
System.out.println(demo.size() + " " + demo.toString());
```

}

For questions J - L consider the following classes and interfaces.:

```
public interface Sport {

     public boolean isTeamSport();

     public String scoreType();
}


public class Soccer implements Sport {

     public boolean isTeamSport() { return true; }

     public String scoreType() { return "Goal"; }
}


public class Cycling implements Sport {
     public int numPeople;

     public Cycling(int n) {numPeople = n; }

     public boolean isTeamSport() { return true; }

     public String scoreType() { return isTeamSport() + " Time"; }
}


public class IndividualPursuit extends Cycling {

     public IndividualPursuit() {super(1); }

     public boolean isTeamSport() { return false; }
}


public class RoadCycling extends Cycling {

     private boolean tour;

     public RoadCycling(int n, boolean t) {
          super(n);
          tour = t;
      }

     public String scoreType() { return "Points"; }
}
```

J. State if each of the following declarations is valid (meaning it will compile without error) or cause a syntax error. (1 point each)

```
RoadCycling rc = new Cycling(12) // J.1
Object obj = new Sport(); // J.2
```

K. What is output by the following code?

```
Sport s1 = new Soccer();
Sport s2 = new RoadCycling(9, true);

System.out.print(s1.isTeamSport() + " " + s2.isTeamSport());
System.out.print(s1.scoreType());
```

L. What is output by the following code?

```
Cycling c1 = new Cycling(5);
IndividualPursuit ip1 = new IndividualPursuit();
Cycling[] cs = {c1, ip1};
for(Cycling c : cs) {
     System.out.println(c.scoreType() + " " + c.isTeamSport());
}
```

M. What is output by the following code?

```
int[] raw = {1, 3, 5, 2, 7, 6};
ArrayList<Integer> list = new ArrayList<Integer>();
for(int x : raw)
     list.add(x);

Iterator<Integer> it = list.iterator();
while(it.hasNext())
     if(it.next() % 2 == 1)
          System.out.print(it.next());
```

N. What is output by the following code?

```
TreeMap<Integer, Integer> tm = new TreeMap<Integer, Integer>();
tm.put(42, 17);
tm.put(17, 31);
tm.put(42, 31);
tm.put(12, 19);
for(Integer x : tm.keySet())
     System.out.print(x + " " + tm.get(x) + " ");
```

O. Which of the following lines causes a syntax error? In one, brief sentance explain the cause of the syntax error. (Compiler warnings are not syntax errors.)

```
ArrayList raw = new ArrayList(); // 1
raw.add(12); // 2
raw.add(15.15); // 3
raw.add(new ArrayList<Integer>()); // 4
raw.add("Isabelle"); // 5
System.out.println(raw.get(2).toString()); // 6
System.out.println(raw.get(3).length()); // 7
System.out.println(raw.get(0).equals(1)); // 8
```

2. The GenericList class. (22 points) To demonstrate encapsulation and the syntax for building a class in Java, we developed a GenericList class to represent a list of any data type. Recall our GenericList class stores the elements of the list in the first N elements of a native array. An elements's position in the list is the same as the element's position in the array. The array may be larger than the list it reprsents. (extra capacity)

Complete an instance method for the GenericList class named interleave. The method inserts elements from the other list into the calling object as shown below. (Each element from the original list is followed by the element at the corresponding position from the other list. If the lists are not the same size, the extra elements are added to the end.)

The method header is:

```
/*   pre: other != null, other != this

     post: interleave the elements of other with this list.
          The list other is not altered by this method.
*/
public void interleave(GenericList<E> other) {
```

Examples of calls to interleave. The list after the -> is the list that invoked the method, after the method is completed.

```
[].interleave([]) -> []
[A, B]. interleave([]) -> [A, B]
[].interleave([A, B]) -> [A, B]
[A, B, C].interleave([D, E, F]) -> [A, D, B, E, C, F]
[A, B].interleave([D, E, F, G]) -> [A, D, B, E, F, G]
[A, B, C, D].interleave([H, I]) -> [A, H, B, I, C, D]
```

You may not use any other methods in the GenericList class unless you define and implement them yourself as part of your answer. You may not use objects or methods from other Java classes, other than native arrays and the Math class.

Hint: Always create a new array to store the elements in the list. The new array may or may not have extra capacity.

Recall this method is in the GenericList class and so you have access to all GenericLists' instance variables.

The GenericList class:

```
public class GenericList<E> {

     private E[] values;
     private int size; // size of list being represented
```

Complete the following instance method for the `GenericList` class.

```
/*   pre: other != null, other != this

     post: interleave the elements of other with this list.
          The list other is not altered by this method.
*/
public void interleave(GenericList<E> other) {
```

3. (16 points total) The `MathMatrix` class. Write an instance method `isIntegerMultiple` for the `MathMatrix` class from assignment 2 that determines if all the values in one matrix (the parameter `other`) are equal to the corresponding cells of the calling `MathMatrix` object times a single, non-zero integer. Neither matrix contains any zeros.

Consider this example with the following 3 x 2 `MathMatrix` as the calling object:

$$\begin{bmatrix} 2 & 3 & 1 \\ -1 & 5 & -2 \end{bmatrix}$$

If `other` is the following `MathMatrix` the method would return `true`. The integer multiple in this case is -3.

$$\begin{bmatrix} -6 & -9 & -3 \\ 3 & -15 & 6 \end{bmatrix}$$

If `other` is the following `MathMatrix` the method would return `false`. The elements in `other` are not equal to the corresponding elements in the calling `MathMatrix` object times a single non-zero integer. (The top left element is equal to 2.5 times the top left element in calling `MathMatrix` object. 2.5 is not an integer.)

$$\begin{bmatrix} 5 & 6 & 2 \\ -2 & 10 & -4 \end{bmatrix}$$

If `other` is the following `MathMatrix` the method would return `false`. Some elements are equal to 2 times the corresponding element in the calling `MathMatrix` object and some are equal to 3 times the corresponding element in the calling `MathMatrix` object.

$$\begin{bmatrix} 4 & 9 & 2 \\ -3 & 15 & -4 \end{bmatrix}$$

Recall the `MathMatrix` class:

```
public class MathMatrix {

    private int[][] elements; // no extra capacity

    public int numRows() // the number of rows in this matrix
    public int numCols() // the number of columns in this matrix

    // pre: 0 <= r < numRows(), 0 <= c < numCols()
    // return the value at the given location
    public int getValue(int r, int c)
```

Complete the instance method for the `MathMatrix` class on the next page.

You may not use any other methods from the `MathMatrix` class other than those shown above unless you implement them yourself as part of your answer. You may not use any other methods or classes from the Java standard library.

```
/*   pre: other != null
          this MathMatrix has at least one row and column.
          other is  the same size as this MathMatrix.
          None of the elements of this or other equal 0.

     post: true if all the elements of other are equal to the
     corresponding cells of the this MathMatrix object times a single,
     non-zero integer.
*/
public boolean isIntegerMultiple(MathMatrix other) {
     // Recall, assume the preconditions are met.
     // Do not write code to check the preconditions.
```

4. Working with ArrayLists, NameSurfer (16 points total) Write an instance method for the `Names` class from assignment 3 that finds all the `NameRecord` objects that have been ranked in every decade and whose ranks never vary by more than a given value. (In statistics this is referred to as the range of a data set.)

For example if the max allowed range is 30, here are some of the names whose range of ranks is less than or equal to 30. (ranks shown in order from first decade to last)

```
William: 2 2 4 4 4 6 7 9 15 19 11       (range of 17)
Joseph: 5 5 7 10 13 13 12 12 10 10 8    (range of 8)
Elizabeth: 6 8 10 17 23 20 17 11 9 8 10 (range of 17)
Victor: 79 67 79 97 91 91 89 92 88 83 94     (range of 30)
David: 29 30 22 11 6 5 2 4 5 11 16      (range of 28)
```

Note, the 30 indicates the max allowed size of the range. It does not mean the ranks must always be in the top 30. Consider Victor. The ranks vary from 67 to 97, a range of 30.

The `NameRecord` class for this question is:

```
public class NameRecord {
     // pre: 0 <= num < Names.NUM_DECADES
     // post: return the rank of this name record for the
     // specified decade. If the name was unranked, returns 0.
     public int getRank(int num)

     // return the name for this name record
     public String getName()

     // return true if this name record is ranked in the top
     // 1000 for every decade
     public boolean alwaysPresent()
```

Note you may not use any other methods from the `NameRecord` class in this question other than the three shown above. You are not allowed to add your own methods to the `NameRecord` class.

The `Names` class for this question

```
public class Names {

     public static final int NUM_DECADES = 11;
     public static final int MAX_RANK = 1000;
     public ArrayList<NameRecord> nameList;
```

You may not use any other methods from the `Names` class other than those shown above unless you implement them yourself as part of your answer. You are not allowed to use any methods or classes except those listed in this question and methods and constructors from the `ArrayList` class and the `Iterator` class.

```
// pre: 0 < rangeSize <= MAX_RANK
// post: Return an ArrayList of names (Strings) that are ranked
// in every decade and whose range does not exceed rangeSize.
public ArrayList<String> inRange(rangeSize) {
```

5. (Lists 16 points). In class implemented lists that store every value in the list explicitly. However, what if most of the elements of the list equal the same value? Consider the following list:

```
 0  1  2  3  4  5  6  7  8  9   10 11  12 13 14 15 16  // position
[A, B, A, A, A, A, A, A, A, A, AAA, A,  B, A, A, C, A] // element
```

Storing all those A's seems like a waste of space. In a ***sparse list***, only the elements not equal to the default value are stored explicitly. The default value is set when the list is created and does not change for a given list. Internally if we use a native array as our storage container we must also store the position of each element, because the position in the array does not necessarily equal the position in the list.

Consider the following internal representation of the list shown above. Each element in the array is a `ListElem` object that stores one element of data and the position of that element in the list. `(position, non-default element)`

The elements not equal to the default element, are stored in the array in ascending order based on their position in the list.

```
     0         1        2        3     index in array
[(1, B), (10, AAA), (12, B), (15, C)]
size of list = 17
elements stored = 4
All elements not stored explicitly equal A for this list.
```

Complete the `get(int pos)` method for a `SparseList` class.

Consider these examples using the list shown above.

```
list.get(0) -> returns A

list.get(1) -> returns B

list.get(11) -> returns A

list.get(15) -> returns C

list.get(16) -> returns A
```

Here is the `ListElem` class:

```
public class ListElem<E> {

     public ListElem(int position, E data) // create element

     public E getData() // return data of this element
     public int getPosition() // return position of this element

     public void setPos(int pos) // change position of this element
     public void setData(E data) // change data of this element
```

The properties of the `SparseList` class are:

- the internal storage container is a native array of `ListElem` objects
- there may be extra capacity in the native array
- only elements not equal to the default element are stored explicitly
- the non-default elements are stored at the beginning of the array in ascending order based on their position in the list
- the size of the list, the number of elements stored explicitly in the array, and the default value are stored in separate instance variables
- any elements in the array that are not referring to active elements of the list are set to `null`
- the default list value never equals `null`

```
public class SparseList<E> {

     private ListElem<E>[] values;
     private int sizeOfList;

     // All values not stored explicitly in values equal defaultValue.
     // defaultValue never equals null.
     private E defaultValue;

     // Number of elements stored explicitly in values.
     // The elements are stored at the beginning of the array.
     // This value could be 0 even if sizeOfList > 0 indicating
     // every element in the list is the default value.
     private int elementsStored;
```

Complete the `get(int pos)` instance method of the `SparseList` class on the next page.

```
/*   pre: 0 <= position < size() of list
     post: Return the element at the given position in this list.
               This list is not altered as a result of this method.
*/
public E get(int pos) {
```

Question 1 answer Sheet.

Name_______________________________________ UTEID __________________________

A.

____

B.

____

C.

____

D.

____

E.

____

F.

____

G.

____

H.

____

I.

____

1.

J. 2.

____

K.

____

L.

____

M.

____

N.

____

O.

____