

Points off	1	2	3	4	5	6	Total off	Net Score

CS 314 – Exam 1 – Spring 2017

Your Name _____

Your UTEID _____

Instructions:

1. There are **6** questions on this test. 100 points available. Scores will be scaled to 175 points.
2. You have 2 hours to complete the test.
3. Place your final answers on this test. Not on scratch paper. Answer in pencil.
4. You may not use a calculator or any other electronic devices while taking the test.
5. When answering coding questions, ensure you follow the restrictions of the question.
6. Do not write code to check the preconditions.
7. On coding questions, you may implement your own helper methods.
8. On coding questions make your solutions as efficient as possible given the restrictions of the question.
9. Test proctors will not answer any questions regarding the content of the exam. If you think a question is ambiguous or has an error, state your assumptions and answer based on those assumptions.
10. When you complete the test show the proctor your UTID, give them the test and all the scratch paper, used or not, and leave the room quietly.

1. (1 point each, 20 points total) Short answer. Place your answer on the line next to or under the question.

Assume all necessary imports have been made.

- 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 best case order (Big O) of methodA? $N = \text{data.length}$

```
public static int methodA(int[] data) {
    int result = 0;
    for (int i = 0; i < data.length; i++) {
        for (int j = i; j < data.length; j++) {
            for (int k = j + 1; k < data.length; k++) {
                if (data[i] == data[j] && data[j] == data[k]) {
                    result++;
                }
            }
        }
    }
    return result;
}
```

- B. What is the worst case order (Big O) of methodB? $N = \text{list.size()}$
list contains N distinct (no repeats) elements in random order. _____

```
// pre: list.size() >= 1
public static int methodB(ArrayList<Integer> list) {
    int result = list.get(0);
    int count = 0;
    for (int i = 1; i < list.size(); i++) {
        int temp = list.get(i);
        if (temp < result) {
            result = temp;
            count++;
        }
    }
    // System.out.println(count);
    return result;
}
```

- C. Assume the println statement in methodB is commented in. If list contains 1000 distinct elements (no repeats) in random order, what is the expected value printed out? In other words, what is the expected value of count under the given conditions after completion of the loop? _____

- D. What is the worst case order (Big O) of methodD? $N = \text{list.size()}$
Assume the Math.random() method is $O(1)$. _____

```
public static void methodD(ArrayList<Double> list, double a) {
    for (int i = list.size() - 1; i >= 0; i--) {
        double temp = Math.random();
        if (temp < a) {
            list.remove(i);
        }
    }
}
```

- E. What is the best case order (Big O) of methodE? $N = \text{data.length}$ _____

```
public static int methodE(boolean[] data) {
    int r = 0;
    for (int i = 0; i < data.length; i++)
        for (int j = data.length - 1; j > 10; j /= 2)
            if (data[i] || data[j])
                r++;
    return r;
}
```

F. Consider the `resize` method for the `IntList` class we developed in lecture:

```
// in the IntList class:
private void resize() {
    int[] temp = new int[size * 2 + 1];
    for (int i = 0; i < size; i++) {
        temp[i] = con[i];
    }
    con = temp;
}
```

The following method takes 2 seconds to complete when $n = 250,000$. How long do you expect it to complete when $n = 750,000$? Assume the `nextInt` method from the `Random` class is $O(1)$.

```
public static IntList methodF(int n, Random r) {
    IntList res = new IntList();
    for (int i = 0; i < n; i++) {
        res.add(r.nextInt());
    }
    return res;
}
```

G. Assume the following change is made to the `resize` method from 1.F.

`int[] temp = new int[size * 2 + 1];` is replaced with the following line of code:

```
int[] temp = new int[size + 75];
```

`methodF` now takes 5 seconds to complete when $n = 20,000$. What is the expected time for the method to complete when $n = 60,000$?

H. What is output by the following code?

```
ArrayList<String> list = new ArrayList<String>();
list.add("UT");
list.add("UT");
list.add(1, "UNT");
list.add(0, 3);
list.add(list.get(0));
System.out.println(list);
```

I. What is output by the following code?

```
ArrayList<String> list2 = new ArrayList<String>();  
list2.add("CS");  
list2.add(1, "ECE");  
list2.add(0, "M");  
list2.add(list2.get(2));  
list2.add(1, "GOV");  
list2.add(0, list2.remove(2));  
System.out.println(list2);
```

J. A method is $O(N^3)$. It takes 2 seconds for the method to complete when $N = 1,000$.
What is the expected time in seconds for the method to complete when $N = 3,000$?

K. A method is $O(N^2)$. It takes 1 second for the method to run when $N = 1,000$. How large can N be if we are willing to allow the method to run for 36 seconds?

L. The following method takes 3 seconds to complete when `data.length = 3,000`.
What is the expected time for the method to complete when `data.length = 6,000`?

```
public static String methodL(char[] data) {  
    String r = "";  
    final int LIMIT = data.length - 3;  
    for (int i = 3; i < LIMIT; i++) {  
        for (int j = i - 3; j < i + 3; j++) {  
            r = data[j] + r;  
        }  
    }  
    return r;  
}
```

M. What is output by the following code? _____

```
String ans = "DBCADEBCABBB";
TreeMap<Character, Integer> tm = new TreeMap<Character, Integer>();
for (int i = ans.length() - 1; i >= 0; i--) {
    tm.put(ans.charAt(i), i);
}
if (tm.size() % 2 == 0) {
    tm.put('I', 3);
}
System.out.print(tm);
// The Map toString takes the form:
// {key1=value1, key2=value2, ..., keyN=valueN}
```

For questions N through T, refer to the classes defined on the sheet at the back of the test.
You may detach that sheet for easier reference.

N. For each line of code write **valid** if the line will compile without error or **invalid** if it causes a compile error. (.5 points each)

```
Object obj1 = new KidViewable(); _____
Musical m1 = new Live(10, "Stomp"); _____
```

O. For each line of code write **valid** if the line will compile without error or **invalid** if it causes a compile error. (.5 points each)

```
PerformArt pa1 = new Dance(30, "42"); _____
KidViewable kv = new Cartoon("fosters"); _____
```

P. What is output by the following code? _____

```
Dance dp = new Dance("swan");
System.out.print(dp.toString());
```

Q. What is output by the following code? _____

```
PerformArt pq = new Animated("ben10");
System.out.println(pq.toString() + " " + pq.time());
```

R. What is output by the following code? _____

```
KidViewable kr = new Musical(10, "bnb", 3);
System.out.println(kr + " " + kr.minAge());
```

S. What is output by the following code? _____

```
Live es = new Live(5, "5th");
System.out.print( (es instanceof PerformArt) + " "
    + (es instanceof KidViewable));
```

T. Consider the following line of code. How many different data types could replace
<Declared Type> in the code below so that the line of code compiles without error?
Answer with an actual number.

```
<Declared Type> tv = new Cartoon("Kim Possible");
```


2. The `GenericList` class (16 points) To demonstrate encapsulation and the syntax for building a class in Java, we developed a `GenericList` class that can store elements of any data type. Recall our `GenericList` class stores the elements of the list in the first N elements of a native array. An element'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 represents.

Create that returns `true` if two given values appear next to each other at some point in the list. The order of the values in the list does not have to match the order of the parameters as seen in the second example.

Examples of calls to the `nextTo` method. (The values shown are `String` objects)

```
[A, B, C, D, A].nextTo(A, B) -> returns true
[A, B, C, D, A].nextTo(B, A) -> returns true
[A, B, C, D, A].nextTo(B, D) -> returns false
[].nextTo(A, B) -> returns false
[A].nextTo(A, B) -> returns false
[A, B, C, D, A].nextTo(BB, AA) -> returns false
[AA, BB, C, D, A].nextTo(BB, A) -> returns false
```

For this question, the `GenericList` will not contain `null` values. The parameters will not store `null`.

The `GenericList` class:

```
public class GenericList<E> {
    private E[] con;
    private int size;
```

You may not use any methods from the `GenericList` class unless you implement them yourself as a part of your solution.

You may call the `equals` method on objects.

Complete the method on the next page.


```
/// pre: val1 != null, val2 != null
// post: return true if val1 and val2 appear next to each other
// in this list.
public boolean nextTo(E val1, E val2) {
```

3. GenericList (16 points) This question uses the `GenericList` class from question 2.

Create an instance method for the `GenericList` class `getMatchingElements`. The method creates and returns a new `GenericList` that contains all the matching elements from the calling list and another list sent as a parameter.

Consider elements from the lists at the same index. For example, given these two lists

```
      0  1  2  3
list1 [A, B, C, A]
list2 [A, A, C, D]
```

The method would return a new list with elements `[A, C]`.

The elements in the resulting list are in the same relative order as the elements from the original lists.

For this question you may assume **none** of the elements in either list equal `null`. If the lists are different sizes only valid elements are considered.

For example, given these two lists

```
list1 [D, B, C, A, Z, M]
list2 [A, B, Z, M]
```

the method shall return the following list: `[B]`

Final example. Given these two lists:

```
list1 [D, B, C, A, Z, M]
list2 [A, C, Z, M]
```

the method shall return the following list: `[]`

The `GenericList` class:

```
public class GenericList<E> {
    private E[] con;
    private int size;

    public GenericList(int initialCapacity) {
        con = (E[]) (new Object[initialCapacity]);
    }
}
```

You may only use the constructor from the `GenericList` class shown above.

You may not use any other methods from the `GenericList` class unless you implement them yourself as a part of your solution. You may not use any other classes besides `GenericList` and native arrays.

You may call the `equals` method on objects.

```
// pre: other != null, no elements of this or other equal null
// post: per the problem description
```

```
public GenericList<E> getMatchingElements(GenericList<E> other) {
```

4. Math Matrix (16 Points) Create a method for the `MathMatrix` class that returns `true` if the calling object is a *symmetric matrix*, false otherwise. A *symmetric matrix* is a matrix that is equal to the transpose of itself

For example:

m1: 4 0 0 not symmetric
 0 0 4

m2: 1 3 symmetric
 3 5

m3: 5 3 5 6 symmetric
 3 0 2 1
 5 2 8 0
 6 1 0 7

m4: 8 2 1 2 not symmetric

m5: 1 1 1 symmetric
 1 1 1
 1 1 1

Recall the `MathMatrix` class:

```
public class MathMatrix {  
    private int[][] myCells; // no extra capacity
```

Do not use any other Java classes besides `MathMatrix`.

You may not use any other methods from the `MathMatrix` class unless you implement them yourself as a part of your answer to this question.

Recall your method shall be as efficient as possible given the restrictions of the question.

Complete the method on the next page.

```
// pre: none
// post: return true if this MathMatrix is symmetric, false otherwise
public boolean isSymmetric() {
```

5. Baby Names (16 points) Write an instance method named `getSubstringNames` for the `Names` class from assignment 3. The method returns an `ArrayList<String>` of names that appear as substrings in other names and meet two other criteria explained below. For example, the name **Bell** is a substring in the name **Isabelle**. Likewise, the name **Liv** is a substring of the name **Olivia**. A name is not considered to be a substring of itself.

The method accepts two `int` parameters. The first parameter, `requiredNames`, specifies the minimum number of other names a name must appear in as a substring to be returned.

For example, if `requiredNames == 20` and `Bell` is a substring in 25 names and `Liv` is a substring in 5 names, `Bell` meets the criteria, but `Liv` does not.

The second parameter, `requiredRank`, specifies the minimum rank a name must have in order to be returned. A name must have at least one rank equal to or better than the required rank.

For example, if `requiredRank == 200` and the name `Bell` has the following ranks:

```
Bell 278 324 423 584 0 0 0 0 0 0 0 0 0 0
```

it would not be returned as part of the resulting `ArrayList`. `Bell` would have to have at least one rank better than 200 (≤ 200) to be valid. Recall a value of 0, means the name was not in the top 1000 names for the decade.

A name must meet both criteria: it must be a substring in the required number of names **AND** have a rank equal to or better than the required rank.

Here are the `Names` and `NameRecord` classes for this question:

```
public class Names {
    private ArrayList<NameRecord> names;
}

public class NameRecord {
    public String getName() // return this NameRecords name
    public int getNumRanks() // return the number of ranks

    // pre: 0 <= decade < getNumRanks(), returns rank for give decade
    // or 0 (zero) if not ranked in given decade
    public int getRank(int decade)
}

```

Do not use any other methods from the `NameRecord` class.

You may create and use an `ArrayList<String>`. You may use any methods from the `ArrayList` class.

You may use any methods from the `String` class you want.

Complete the following instance method for the `Names` class. Do not use any other methods in the `Names` class unless you implement them yourself. Do not alter the `names` field. Recall, your method shall be as efficient as possible given the restrictions of the question. We expect the `names.size()` to be much larger than the number of ranks for the `NameRecord` objects,

```
// pre: requiredNames > 0, 0 < requiredRank < 1000
public ArrayList<String> getSubstringNames(int requiredNames,
                                           int requiredRank) {
```

6. Other data structures (16 points) If most of the values in a matrix are the same value, usually 0, the matrix is said to be *sparse*. Many problems in computing, science, and engineering involve large, sparse matrices. 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 matrix.

2	0	4	0	0
0	12	0	0	0
0	0	0	-131	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 a non-zero element in the matrix.)

Array of `SMEEntry` objects. Extra capacity shown as nulls.

```
[(0,0,2), (0,2,4), (1,1,12), (2,3,-131), (3,2,5), (3,4,2), null, null]
```

```
public class SparseMatrix {
    private int numRows;
    private int numCols;
    private SMEEntry[] nonZeros; // extra capacity possible
    private int numNonZeros;

    public void setNonZeroValue (int row, int col, int val) // todo
}
```

A `SparseMatrix` object that models the matrix shown above would have `numRows = 4`, `numCols = 5`, `numNonZeros = 6`, and an array of `SMEEntry` objects as shown above.

The `SMEEntry` objects are stored in row major order in the array. In other words, the elements in the array appear in the order you would get scanning the matrix from left to right, top to bottom.

The `SMEEntry` class is static, nested class inside the `SparseMatrix` class, so you have access to any `SMEEntry` object's private instance variables.

```
private static class SMEEntry {
    private int row;
    private int col;
    private int val;
}
```

Implement the `setNonZeroValue` method for the `SparseMatrix` class. The method sets the value of an element in the `SparseMatrix` that is initially a non-zero value to a new, given value.

Example of effects on the given array with calls to `setNonZeroValue(int row, int col, int val)`

```
m.setNonZeroValue(1, 1, -3) -> array becomes
[(0,0,2), (0,2,4), (1,1,-3), (2,3,-131), (3,2,5), (3,4,2), null, null]
```

```
m.setNonZeroValue(2, 3, 0) -> array becomes
[(0,0,2), (0,2,4), (1,1,-3), (3,2,5), (3,4,2), null, null, null]
```


You may not use any other methods from the `SparseMatrix` class unless you implement them yourself as part of your answer.

```
// pre: row and col are in bounds for this SparseMatrix,  
// getVal(row, col) != 0  
public void setNonZeroValue(int row, int col, int val) {
```


For questions N - T, consider the following classes. **You may detach this sheet from the test.**

```
public abstract class PerformArt {
    private String name;

    public PerformArt(String n) { name = n; }

    public abstract int time();

    public String toString() {return name + time(); }

    public String getName() { return name; }
}

public class Live extends PerformArt {

    private int tx;

    public Live(int t, String n) {
        super (n);
        tx = t;
    }

    public int time() { return tx * 2; }
}

public class Recorded extends PerformArt {

    public Recorded(String s) { super(s); }

    public int time() { return 60; }
}

public interface KidViewable {
    public int minAge();
}

public class Dance extends Live {
    public Dance(int t, String n) { super(t, n); }

    public Dance(String n) { super(30, n); }

    public int time() { return 15; }
}

public class Animated extends Recorded implements KidViewable {

    public Animated(String s) { super(s); }

    public String toString() { return getName(); }

    public int minAge() { return 10; }
}
```

```
public class Musical extends Live implements KidViewable {  
  
    private int acts;  
  
    public Musical(int t, String n, int a) {  
        super(t, n);  
        acts = a;  
    }  
  
    public int time() { return minAge() * 2; }  
  
    public int minAge() { return acts * 2; }  
}  
  
public class Cartoon extends Animated {  
  
    public Cartoon(String s) {  
        super(s + "EX!");  
    }  
}
```