Points off	1	2	3	4	5		Total off	Net Score

CS 314 – Exam 1 – Fall 2018

Your Name: ______________________________________

Your UTEID: _____________________________________

Circle your TAs Name:
Amir Andrew Ben Chris Ethan
Hailey Jacob Levi Lucas Madhav
Smruti Suhas

Instructions:

1. There are **5** questions on this test. 80 points available. Scores will be scaled to 160 points.
2. You have 2 hours to complete the test.
3. Place your final answers on this test. Not on the 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, 15 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 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 shall be the most restrictive correct Big O function. For example, Selection Sort is average case Order N^2, but per the formal definition of Big O it is correct to say Selection Sort is Order N^3 and Order N^4.
   Give the most restrictive, correct Big O function. (Closest without going under.)
   e. Assume $\text{log}_2\text{(1,000) = 10}$ and $\text{log}_2\text{(1,000,000) = 20}$.

A. What is the T(N) of the following method? N = n ______________________________

```
public static double methodA(int n) {
      double r = 0.0;
      for (int i = 1; i <= n; i++) {
            int lim = n * n;
            double a = i;
            for (int j = 1; j <= lim; j++) {
                  r += a / j;
                  r++;
            }
      }
      return r;
}
```

B. The following methods takes 3 seconds to complete when `data.length` = 10_000. What is the expected time for the method to complete when `data.length` = 20_000?

Method `test` is O(N) where N is the length of the array sent as an argument to the method.

____

```
public static int methodB(int[] data) {
    int r = 0;
    for (int i = 0; i < data.length; i++) {
        r += test(data, i, i);
        for (int j = 0; j < data.length; j++) {
            r += test(data, i, j);
        }
    }
    return r;
}
```

C. What is the best case order (Big O) of the following method? N = `data.length`. _____________

```
public static int methodC(int[] data, int t) {
    int r = 0;
    for (int i = 0; i < data.length; i++) {
        for (int j = data.length - 1; j > i; j--) {
            if (data[j] < i) {
                r += j * i;
            }
        }
    }
    return r;
}
```

D. What is the order (Big O) of the following method? N = `n` _____________

```
public static int methodD(int n) {
    int r = 0;
    for (int i = 1; i < n; i *= 2) {
        for (int j = 0; j < n; j++) {
            r += i * j * j;
        }
    }
    return r;
}
```

E. A method is O(N!). It takes 2 seconds to complete when N = 39. What is the expected time in seconds for the method to complete when N = 40?

____

F. A method is O(N^2). It takes 1 second for the method to complete when N = 10,000.
What is the expected time in seconds for the method to complete when N = 50,000?

____

G. A method is O($N\log_2 N$). It takes 0.4 seconds for the method to complete when N = 1,000,000.
What is the expected time in seconds for the method to complete when N = 4,000,000?
Give you answer in simplified form.

____

H. We are considering making the default capacity of an array based list a constant. Of the following two options which is better and why?

```
public class GenericList<E> {

     private final int DEFAULT_CAP_A = 10; // Option A

     private static final int DEFAULT_CAP_B = 10; // Option B
```

Answer:

I. The following method takes 2 seconds to complete when the size of both lists is 5,000. What is the expected time in seconds for the method to complete when the size of both lists is 15,000? In the first case the if statement is true 5 times and in the second case the if statement is true 15 times.

____

```
public static ArrayList<Integer> methodI(ArrayList<String> listA,
                                         ArrayList<String> listB) {

     ArrayList<Integer> result = new ArrayList<>();
     for (int i = 0; i < listA.size(); i++) {
          String a = listA.get(i);
          if (listB.contains(a)) {
               int j = listB.indexOf(a);
               result.add(j);
          }
     }
     return result;
}
```

J. What is output by the following code? The `Map.toString` returns elements in the form `{key1=value1, key2=value2, ..., keyN=valueN}`.

____

```
TreeMap<Integer, String> m = new TreeMap<>();
m.put(5, "MMM");
m.put(3, "A");
m.put(2, "BB");
m.put(m.get(5).length(), m.get(2));
m.put(2, "IT");
m.put(5, "Q");
System.out.print(m);
```

<u>For questions K through O, refer to the following classes.</u>

```
public abstract class WritingImplement {

      public void show() {  System.out.print("imp "); }

      public void print() { show(); }

      public abstract int uses();

      public void test() { System.out.print("NA"); }
}

public class Pencil extends WritingImplement {

      private int len;

      public Pencil(int start) { len = start; }

      public int uses() { return len; }

      public void test() { len -= 2; }
}

public class Pen extends WritingImplement {

      private String color;
      private int cap;

      public Pen(String c) { color = c; cap = 5; }

      public void test() { cap -= 1; }

      public void refill(int c) { cap += c; }

      public int uses() { return cap * 10; }

      public void show() { System.out.print(color + " "); }
}
```

K. What is output by the following code? ____________________________________

```
WritingImplement[] wk = {new Pen("red"), new Pencil(5), new Pen("b")};
for (WritingImplement w : wk) {
     w.test();
     System.out.print(w.uses() + " ");
}
```

L. What is output by the following code? ____________________________________

```
Pencil p1 = new Pencil(5);
p1.print();
System.out.print(p1.uses());
```

M. What is output by the following code? ____________________________________

```
WritingImplement w = new Pen("red");
w.show();
w.refill(10);
w.show();
```

N. What is output by the following code? ____________________________________

```
Pen p2 = new Pen("red");
p2.refill(5);
p2.print();
System.out.print(p2.uses());
```

O. What is output by the following code? ____________________________________

```
Pen p3 = new Pen("red");
p3.test();
WritingImplement w3 = p3;
w3.test();
w3.test();
p3.refill(2);
w3 = new Pen("blue");
p3.test();
w3.test();
System.out.print(p3.uses());
```

2. The GenericList class (14 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 have extra capacity and thus be larger than the list it represents.

Complete a method that removes all elements at the end of the list equal to a given value.

```
/*   pre: tgt != null
     post: Removes all elements from the end of this list equal to
     tgt. Returns the number of elements removed. */
public int removeFromEnd(E tgt) {
```

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

```
[A, B, C, D, A, B].removeFromEnd(A) -> returns 0, list unchanged

[].removeFromEnd(A) -> returns 0, list unchanged

[A, A].removeFromEnd(A) -> returns 2, list becomes []

[A, A, B, A, B, A, A, A].removeFromEnd(A)
                         -> returns 3, list becomes[A, A, B, A, B]
```

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. Do not use any other Java classes or methods except the equals method.

The list does not store null elements.

Complete the method on the next page.

```
/*   pre: tgt != null
     post: Removes all elements from the end of this list equal to
     tgt. Returns the number of elements removed. */
public int removeFromEnd(E tgt) {
```

3. Math Matrix (17 Points) Create an instance method for the `MathMatrix` class that returns `true` if the calling object contains a row that is an integer multiple of another row, `false` otherwise.

A row r2 is an integer multiple of another row r1 if r2[i] = r1[i] * c for all elements in each row, where c is a non-zero integer.

For example:

```
r1 =  -5   3  -2    6   10
r2 =  15  -9   6  -18  -30
```

Each element in r2 is equal to the corresponding element in r1 * -3.

In the following example r2 is not an integer multiple of r1

```
r1 =  -3   1   -5    3    3
r2 =  -6   2  -10    5    6
```

It appears c may equal 2, but 3 * 2 != 5, therefore r2 is not an integer multiple of r1

Recall the `MathMatrix` class:

```
public class MathMatrix {

     private int[][] cells; // no extra capacity
```

Do not use any other Java classes or methods besides native arrays. 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 in terms of time and space give the restrictions.

Your method is returning `true` if the calling `MathMatrix` object has a row that is an integer multiple of another row besides itself. You do not have to return which rows they are or if more than one pair of rows are integer multiples of each other.

The precondition for the method is that none of the elements of the calling `MathMatrix` object equal 0.

Complete the method on the next page.

```
/*   pre: no element in this MathMatrix equals 0
     post: return true if this MathMatrix has at least one row that is
     an integer multiple of another row.
     This MathMatrix is not altered.*/
public boolean hasIntegerMultipleRow() {
```

4. Baby Names and Using Maps (17 points) Complete an instance method for the `Names` class that returns an `ArrayList<String>` of names from a given `ArrayList<String>` that are both present in a the `Names` object and are *steady names*. A steady name is one where the difference in ranks from **consecutive decades** never exceeds a given `limit`.

Recall, a value of 0 indicates the name had a rank greater than 1000 for the decade. **For this question assume a 0 indicates a rank of 1001 when calculating the difference between ranks.** Consider the following examples:

`Olivia 315 311 290 307 270` **`355 504`** `324 193 47 16`
With a limit of 100, Olivia is not steady. |355 - 504| > 100. Olivia would require a limit >= 180 to be steady. |324 - 504| = 180

`Orvil 0` **`0 945`** `0 0 0 0 0 0 0 0`
With a limit of 100, Orvil is steady. |1001 - 945| <= 100. Recall each rank of 0 is treated as a rank of 1001.

`Pablo 325` **`367 287`** `285 327 366 396 379 332 324 286`
With a limit of 100 Pablo is steady. Pablo is steady with any limit >= 80, |287 - 387| = 80 and not steady for any limit < 80.

Unlike the assignment this `Names` class uses a `Map` whose keys are `String`s and values are the associated `NameRecord` object. The map keys are the baby names from the associated NameRecord. In other words the key is the same as getName() from the value associated with that key.

The `Names` class you will use on this question is as follows:

```
public class Names {
      // The NameRecords in this Names object.
      // All NameRecords have numDecades ranks.
      private Map<String, NameRecord> myRecs;
      private int numDecades;
}
```

<u>Methods you may use from `NameRecord`: **You may not add methods to the NameRecord class.**</u>

```
String getName() - return the name for this NameRecord
int getRank(int decade) - return the rank for the given decade. Uses 0 based
indexing. Returns 0 if unranked in the given decade.
```

<u>Methods you may use from the `Map` interface:</u>

```
V get(K key) return the value associated with the given key or null if
           given key is not present in this map
int size() the number of key value pairs in this Map
V put(K key, V val) associates the given val with the given key in
      this map. If key is already present replace and return the old value.
Set<K> keySet() returns a set of all keys in this Map.
```

<u>Methods you may use from the `ArrayList` class:</u>

```
ArrayList() - construct an empty ArrayList
add(E obj) - add obj to the end of this ArrayList
int size() - number of elements in this ArrayList
E get(int pos) - access element at given position
int indexOf(Object o) - return first index of o in list or -1 if not present
```

You may also use the `equals` method on `String`s, the `Math.abs` method, and `for-each` loops.

```
/* pre: names != null, no elements of names == null, limit > 0
   post: return an ArrayList of elements from names that are present
      in this Names object and steady based on the given limit.
      Neither names or this Names object are altered by this method. */
public ArrayList<String> getSteadyNames(ArrayList<String> names, int limit) {
```

5. Maps (17 points) Complete the `put` method for the `ArrayMap` class.

Recall, the `put` method accepts a key-value pair. If the key is not present, the key-value pair is added to the map. The method returns `null` in this case. If the key is already present, the new, given value replaces the old value, and the old value is returned.

The `ArrayMap` classes uses a 2d array of `Object`s with 2 rows to store the key-value pairs.

The first row of the 2d array stores the map's keys and the second row stores the associated value for a key at the same index. The map does not allow `null` keys or `null` values.

The `ArrayMap` class is generic based on the Java inheritance requirement and polymorphism, not Java generics.

The `ArrayMap` class does **not** implement the Java `Map` interface,

Like the `GenericList` class we developed in lecture, the array in the `ArrayMap` class may have extra capacity, and thus has a `size` variable to track the number of key-values pairs in the map.

The given `ArrayMap` class:

```
public class ArrayMap {

     private Object[][] kvPairs;
     private int size; // The number of key-value pairs in this Map.

     public ArrayMap() {
          kvPairs = new Object[2][10];
     }

     /* Resizes kvPairs to the given capacity and copies over
          existing elements. */
     private void resizeArray(int newCapacity)

     /* pre: key != null, value != null
        post: per the problem description */
     public Object put(Object key, Object value)
```

Do not use any other Java methods or classes except the `equals` method. Do not use any `ArrayMap` methods except those given above unless you implement them as a part of your answer.

Of course you can use native arrays including the `length` field.

Complete the method on the next page.

```
/* pre: key != null, val != null
   post: per the problem description */
public Object put(Object key, Object val) {
```