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

CS 307 – Midterm 1 – Spring 2011

Your Name_____

Your UTEID _____

Circle yours TA's name: Dan Muhibur Oliver

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 answering coding questions ensure you follow the restrictions of the question.
7. 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 samples state the output. If the code would cause a syntax error then answer "syntax error". If it would cause a exception at runtime then answer "exception". If it would result in an infinite loop answer "infinite loop".

A. What is output by the following code when the client code is executed?

```
public int a(int x, int y){
     x++;
     y--;
     x++;
     return x * y;
}

// client code
int x = 3;
int y = a(x, x);
System.out.println(x + " " + y);
```

B. What is output by the following code?

```
int[] list1 = {1, 9, 6, 5};
int[] list2 = list1;
for(int i = 0; i < list1.length; i++)
    list1[i]++;
for(int i = 0; i < list2.length; i++)
    list2[i]++;
for(int i = 0; i < list1.length; i++)
    System.out.print(list1[i] + " "); // single space
```

C. What is output by the following code when the client code is executed?

```
public void c(int[] data){
     data[0]--;
     data[1]++;
     int[] list2 = {2, 3, 4, -5};
     data = list2;
     data[0]++;
}

// client code
int[] data = {10, 6, 4};
c(data);
for(int i = 0; i < data.length; i++)
    System.out.print(data[i] + " "); // single space
```

D. What is output by the following code?

```
String[] names = new String[5];
names[0] = "Olivia";
names[1] = "Isabelle";
int total = 0;
try {
    for(int i = 0; i < names.length; i++)
        total += names[i].length();
}
catch(Exception e) {
    total = -2;
}
System.out.println(total);
```

E. Recall the IntList class we developed in lecture.

```
public IntList() // create an empty IntList
public int size() // return the size of this IntList
public void add(int val) // add val to the end of this IntList
public void insert(int pos, int val) // insert val at the specified position
public int remove(int pos) // remove the value at the specified position
public String toString() // return a String representation of this IntList
```

What is output by the following code?

```
IntList list3 = new IntList();
System.out.print(list3.size() + " "); // single space
list3.add(5);
list3.add(12);
list3.add(-5);
list3.insert(2, 7);
list3.add(list3.remove(2));
System.out.println(list3);
```

F. What is output when the following client code is executed?

```
public class Sample {

    private int sX;
    private String st;

    public Sample() {
      this(5);
      priv();
      System.out.print("A");
    }

    public Sample(int x) {
       sX = x;
       System.out.print("B");
       priv();
    }

    private void priv() { System.out.print("C"); }
}

// client code
Sample s1 = new Sample();
Sample s2 = new Sample(5);
```

For questions G - O consider the following classes and interfaces.

```
public interface GunpowderWeapon {
    public int getCharge();
}

public abstract class GamePiece {
    private String name;

    public GamePiece() { name = "foo"; }

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

    public abstract int getSize();

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

public class Tower extends GamePiece {
    private int height;

    public Tower(String name, int h) {
        super(name);
        height = h;
    }

    public int getSize() { return height / 2; }

    public void improve() { height += 2; }

    public String toString() { return "h: " + height; }
}

public class ArrowTower extends Tower {

    public ArrowTower(String name, int h) { super(name, h); }

    public void improve() {
        super.improve();
        super.improve();
    }
}

public class BallistaTower extends ArrowTower {

    public BallistaTower() { super("BA", 10); }

    public String toString() { return "size: " + getSize(); }
}
```

```
public class CannonTower extends Tower implements GunpowderWeapon {

    private int shotWeight;

    public CannonTower(int sw) {
        super("boom", 6);
        shotWeight = sw;
    }

    public int getSize() { return shotWeight / 10; }

    public int getCharge() { return shotWeight / 5; }

    public String toString() {
      return super.toString() + ", " + shotWeight;
    }
}
```

G. State if each of the following declarations is valid (meaning it will compile with no error) or invalid (meaning it causes a syntax error). (1 point each)

```
GamePiece gp1 = new GamePiece("laser"); // G.1
GamePiece gp2 = new CannonTower(40); // G.2
```

H. State if each of the following declarations is valid (meaning it will compile with no error) or invalid (meaning it causes a syntax error). (1 point each)

```
GamePiece[] gpList = new GamePiece[10]; // H.1
Object obj1 = new ArrowTower("dirge", 10); // H.2
```

I. State if each of the following declarations is valid (meaning it will compile with no error) or invalid (meaning it causes a syntax error). (1 point each)

```
GunpowderWeapon gpw1 = new CannonTower(80); // I.1
Comparable comp1 = new CannonTower(40); // I.2
```

J. State if each of the following declarations is valid (meaning it will compile with no error) or invalid (meaning it causes a syntax error). (1 point each)

```
BallistaTower bt1 = new ArrowTower("bat", 20); // J.1
ArrowTower at1 = new CannonTower(120); // J.2
```

K. What is output by the following code?

```
ArrowTower at3 = new ArrowTower("AA", 6);
System.out.print(at3);
```

L. What is output by the following code?

```
Tower t4 = new ArrowTower("BB", 4);
t4.improve();
System.out.print(t4.getSize());
```

M. What is output by the following code?

```
Tower tow3 = new CannonTower(40);
System.out.println(tow3.getSize());
```

N. What is output by the following code?

```
ArrowTower at5 = new ArrowTower("CC", 10);
System.out.println( ((BallistaTower) at5).toString() );
```

O. What is output by the following code?

```
ArrowTower at6 = new ArrowTower("DD", 12);
Tower t6 = at6;
at6.improve();
t6.improve();
System.out.println(at6.getSize());
```

2. The `IntList` class. (20 points) In lecture, to demonstrate encapsulation and the syntax for building a class in Java, we developed an `IntList` class to represent a list of `int`s. Recall our `IntList` class stores the elements of the list in the first N elements of a native array. The position in the list corresponds directly to the position in the array. The array may be larger than the list that is being represented.

Complete an instance method for the `IntList` class named `maxSortedLength`. The method returns an `int` equal to the length of the longest contiguous section of the list that is sorted in ascending order.

Your method will be grading for efficiency. (5 out of 20 points) In order to receive full credit your solution shall be no worse than c * N, where c is a positive constant and N is the number of elements in the list.

The method header is:

```
/* pre: none

post: return the length of the longest contiguous section of this list
that is sorted in ascending order. This list is not altered as a result of
this method call
*/
public int maxSortedLength() {
```

Examples of calls to `maxSortedLength`. The longest contiguous section of elements in ascending values is underlined in these examples. (In the case of ties, the first section is underlined.)

```
[].maxSortedLength() returns 0
[12].maxSortedLength() returns 1
[-5, 10, 12, 32, 100].maxSortedLength() returns 5
[-5, -10, -20, 32, 100].maxSortedLength() returns 3
[0, 0, 0, 0].maxSortedLength() returns 4
[10, 5, 0, -5]. maxSortedLength() returns 1
[0, 1, 0, 1, 0, 1, 0, 1]. maxSortedLength() returns 2
[4, 2, 0, 2, 4, 6, 4, 2, 0]. maxSortedLength() returns 4
```

You may not use any other methods in the `IntList` 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 methods from the `Math` class.

Recall the `IntList` class:

```
public class GenericList{

     private int[] container;
     private int size; // size of list being represented

     // other parts of class not shown
```

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

```
/* pre: none

post: return the length of the longest contiguous section of this list
that is sorted in ascending order. This list is not altered as a result of
this method call
*/
public int maxSortedLength() {
```

3. (30 points total) The `MathMatrix` class. Write an instance method for the `MathMatrix` class from assignment 3 that determines if an array of ints is an *eigenvector* for the calling `MathMatrix` object. Given a square matrix M and an array (aka a vector) of values x, then x is an eigenvector for M if multiplying M by x generates a vector that is equal to the original vector multiplied by a constant, other than 0. In other words $Mx = cx$ where M is the matrix, x is the vector, and c is any value besides 0.

Consider the example of a 3 x 3 matrix and an array of 3 values. Mx = v.

$$\underset{M}{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}} * \underset{x}{\begin{bmatrix} j \\ k \\ l \end{bmatrix}} = \underset{v}{\begin{bmatrix} a*j+b*k+c*l \\ d*j+e*k+f*l \\ g*j+h*k+i*l \end{bmatrix}} = \underset{v}{\begin{bmatrix} m \\ n \\ o \end{bmatrix}}$$

Multiplying M and x results in the vector v with values m, n, and o. x is an eigenvector of M if there is a non zero constant c such that j = c * m, k = c * n, and l = c * o. It is possible for c to be a real number.

Here is an example with actual values:

$$\underset{M}{\begin{bmatrix} 1 & 2 & 1 \\ 6 & -1 & 0 \\ -1 & -2 & -1 \end{bmatrix}} * \underset{x}{\begin{bmatrix} 2 \\ 3 \\ -2 \end{bmatrix}} = \underset{v}{\begin{bmatrix} 1*2+2*3+1*-2 \\ 6*2+-1*3+0*-2 \\ -1*2+-2*3+-1*-2 \end{bmatrix}} = \underset{v}{\begin{bmatrix} 6 \\ 9 \\ -6 \end{bmatrix}}$$

x is an eigenvector of M because v = 3 * x.

Consider this example.

$$\underset{M}{\begin{bmatrix} 1 & 2 & 1 \\ 6 & -1 & 0 \\ -1 & -2 & -1 \end{bmatrix}} * \underset{x1}{\begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}} = \underset{v1}{\begin{bmatrix} 1*2+2*1+1*3 \\ 6*2+-1*1+0*3 \\ -1*2+-2*1+-1*3 \end{bmatrix}} = \underset{v1}{\begin{bmatrix} 7 \\ 11 \\ -7 \end{bmatrix}}$$

x1 is not an eigenvector of M. 7 / 2 = 3.5, 11 / 1 = 11, and -7 / 3 = $-2.\overline{3}$. These values do not all equal each other, thus x1 is not an eigenvector of M.

In this problem the x vector will be a 1D array of `int`s, not a `MathMatrix` object. In the examples show x is an array with a length equal to 3. For illustration purposes the array is shown vertically rather than horizontally. In the first example x refers to an array equal to `{2, 3, -2}`.

Recall the `MathMatrix` class:

```
public class MathMatrix {

    private int[][] myCells; // 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 following instance method in the `MathMatrix` class. **You may not use any other methods from the `MathMatrix` class other than those shown on the previous page unless you implement them yourself as part of your answer. You may not use methods or classes from the Java standard library other than native arrays.**

```
/* pre: xVec.length = numRows(), this is a square MathMatrix, numRows > 0
post: return true if xVec is an eigenvector of this MathMatrix, false
otherwise. Neither xVec or this MathMatrix are altered as a result of this
method call.*/
public boolean isEigenVector(int[] xVec) {
     assert (numRows() > 0) && (numRows() == numCols())
          && (numRows() == xVec.length);
```

4. Working with objects (20 points total) Write an instance method for the `Names` class from assignment 4 that returns an `ArrayList` of `NameRecord` objects that have names that are formed from two other names in the `Names` object. These are referred to as *compound names*.

For example, if the names "joe" and "bob" are in the `Names` object is the name "joebob"? You should consider all combinations of two names. For example, given "joe" and "bob",check for the names "joebob", "bobjoe", "joejoe", and "bobbob. You do not need to prevent duplicate `NameRecord`s from being added. For example if the names "jo", "joe", "bob", "ebob", and "joebob" are all present then "joebob" may appear twice in the result.

The database of names you used on the assignment contained 272 compound names including Lu + Lu = Lulu, Stan + Ford = Stanford, Isa + Belle = Isabelle, Jill + Ian = Jillian, and Jo + Ellen = Joellen.

The `NameRecord` class for this question is:

```
public class NameRecord {
     // returns this NameRecord's name
     public String getName()

     // return a copy of this NameRecord object. The returned value
     // may safely be cast to NameRecord.
     public Object clone()
```

Methods from the `ArrayList` class:

```
public ArrayList() - constructor that creates an empty ArrayList

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()

public boolean add(E elem) - Appends the specified element to the end of
this list. Always returns true.
```

The `Names` class for this question is:

```
public class Names {

     // the NameRecords in this Names object.
     private ArrayList<NameRecord> nameList;

     // return the NameRecord associated with name, ignoring case
     // if there is no NameRecord with name, null is returned
     public NameRecord getName(String name)
}
```

You are not allowed to use any other methods from the `Names`, `NameRecord` or `ArrayList` classes other than those shown above plus `String` concatenation, the `String` `toLowerCase` method, and the `equals` method. You may not use any other Java classes.

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

```
/* pre: none
   post: return an ArrayList of the NameRecords in this Names object
that have names that are compound names formed by a pair of names in
this Names object. If there are no such NameRecords return an empty
ArrayList. The return value may contain duplicates. This Names object
is not altered as a result of this method call.
*/
public ArrayList<NameRecord> getCompoundNames() {
```

Scratch Paper

Question 1 answer Sheet.

Name_____

A. _____

B. _____

C. _____

D. _____

E. _____

F. _____

G. 1. 2. _____

H. 1. 2. _____

I. 1. 2. _____

J. 1. 2. _____

K. _____

L. _____

M. _____

N. _____

O. _____