CSE 142, Spring 2009, Sample Final Exam #2

Name: _____

Section: _____ **TA:** _____

Student ID #: _____

Rules:

- You have 110 minutes to complete this exam.
  You will receive a deduction if you keep working after the instructor calls for papers.
- This test is open-book/notes.
- You may not use any electronic device of any kind including calculators.
- Your code will be graded on proper behavior/output, not on style.
- You do not need to write any `import` statements in your code.
- Do not abbreviate any code that you write.
- Do not abbreviate any answer asking for output (show the complete output).
- If you enter the room, you must turn in an exam and will not be permitted to leave without doing so.
- You must show your Student ID to a TA or instructor for your submitted exam to be accepted.

Good luck!

Problem	Description	Earned	Max
1	Array Simulation		10
2	Inheritance		7
3	Parameters / References		8
4	Token-Based File Processing		8
5	Line-Based File Processing		10
6	Arrays		10
7	ArrayList		8
8	Critters		15
9	Arrays		14
10	Programming		10
TOTAL	**Total Points**		**100**

1. Array Simulation, 10 points

Consider the following method:

```
public static void arrayMystery(int[] list) {
   for (int i = 0; i < list.length; i++) {
       list[i] = i * list[i];
   }
}
```

Indicate in the right-hand column what values would be stored in the array after the method `arrayMystery` executes if the array in the left-hand column is passed as its parameter.

Original Contents of Array	Final Contents of Array
`int[] a1 = {};` `arrayMystery(a1);`	_____
`int[] a2 = {7};` `arrayMystery(a2);`	_____
`int[] a3 = {3, 2};` `arrayMystery(a3);`	_____
`int[] a4 = {5, 4, 3};` `arrayMystery(a4);`	_____
`int[] a5 = {2, 4, 6, 8};` `arrayMystery(a5);`	_____ _

2. Inheritance, 7 points

Assume the following four classes have been defined:

```
public class A extends B {
   public void method2() {
      System.out.println("a 2");
   }
}

public class B extends C {
   public String toString() {
      return "b";
   }

   public void method2() {
      System.out.println("b 2");
   }
}
```

```
public class C {
   public String toString() {
      return "c";
   }

   public void method1() {
      System.out.println("c 1");
   }

   public void method2() {
      System.out.println("c 2");
   }
}

public class D extends B {
   public void method1() {
      System.out.println("d 1");
   }
}
```

Given the classes above, what output is produced by the following code?

```
C[] elements = {new A(), new B(), new C(), new D()};
for(int i = 0; i < elements.length; i++) {
    System.out.println(elements[i]);
    elements[i].method1();
    elements[i].method2();
    System.out.println();
}
```

3. Parameters and References, 8 points

The program below produces 4 lines of output. What are they?

```
public class Mystery {
   public static void main(String[] args) {
      int x = 0;
      int[] a = new int[4];

      x = x + 1;
      mystery(x,a);
      System.out.println("x is " + x + " and a is " + Arrays.toString(a));

      x = x + 1;
      mystery(x,a);
      System.out.println(x + " " + Arrays.toString(a));
   }

   public static void mystery(int x, int[] a) {
      x = x + 1;
      a[x] = a[x] + 1;
      System.out.println(x + " " + Arrays.toString(a));
   }
}
```

Line 1: _____

Line 2: _____

Line 3: _____

Line 4: _____

4. Token-Based File Processing, 8 points

Write a static method called `evaluate` that takes as a parameter a `Scanner` containing a series of tokens that represents a numeric expression involving addition and subtraction and that returns the value of the expression. For example, if a Scanner called data contains the following tokens:

```
4.2 + 3.4 - 4.1
```

and we make the following call:

```
evaluate(data)
```

the method should evaluate the result as (4.2+3.4-4.1) = (7.6-4.1) = 3.5 and should return this value as its result. Every expression will begin with a real number and then will have a series of operator/number pairs that follow. The operators will be either "+" (addition) or "-" (subtraction). As in the example above, there will be spaces separating numbers and operators. You may assume the expression is legal.

Your program should evaluate operators sequentially from left to right. For example, for this expression:

```
7.3 - 4.1 - 2.0
```

your method should evaluate the operators as follows:

```
7.3 - 4.1 - 2.0 = (7.3 - 4.1) - 2.0 = 3.2 - 2.0 = 1.2
```

The Scanner might contain just a number, in which case your method should return that number as its result.

5. Line-Based File Processing, 10 points

Write a static method called `processScores` that takes as a parameter a `Scanner` containing a series of lines that represent student records. Each student record takes up two lines of input. The first line has the student's name and the second line has a series of plus and minus characters. Below is a sample input:

```
Kane, Erica
--+-+
Chandler, Adam
++-+
Martin, Jake
+++++++
Dillon, Amanda
++-++-+-
```

The number of plus/minus characters will vary, but you may assume that at least one such character appears and that no other characters appear on the second line of each pair.

For each student you should produce a line of output with the student's name followed by a colon followed by the percent of plus characters. For example, if the input above is stored in a Scanner called input and we make the following call:

```
processScores(input);
```

The following output should be produced:

```
Kane, Erica: 40.0% plus
Chandler, Adam: 75.0% plus
Martin, Jake: 100.0% plus
Dillon, Amanda: 62.5% plus
```

You must exactly reproduce this format.

6. Arrays, 10 points

Write a static method called `isFibLike` that takes an array of integers as a parameter and that returns whether or not the sequence matches the pattern of the Fibonacci sequence (true if it does, false if it does not). The Fibonacci sequence begins with the number 1 followed by the number 1 and each successive value is the sum of the two previous values:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, and so on.

It is possible to follows this pattern with different starting values. For example, Lucas numbers start with the values 2 and 1 but otherwise follow the Fibonacci pattern.

Your method should determine whether each value after the first two is the sum of the previous two values in the sequence, returning true if the sequence has that pattern and returning false if it does not. If the array has two or fewer values, your method should return true. Below are sample arrays and the value that should be returned for each:

Contents of Array passed to isFibLike	Value returned by isFibLike
`{}`	`true`
`{42}`	`true`
`{18, 42}`	`true`
`{1, 1, 1}`	`false`
`{1, 2, 3}`	`true`
`{0, 0, 0, 0, 0}`	`true`
`{1, 1, 2, 3, 5, 8, 13, 21}`	`true`
`{2, 1, 3, 4, 7, 11, 18, 29}`	`true`
`{1, 1, 2, 3, 5, 12, 17}`	`false`

7. ArrayList, 8 points

Write a static method called `removeShorterStrings` that takes an `ArrayList` of Strings as a parameter and that removes from each successive pair of values the shorter String in the pair. For example, suppose that an ArrayList called `list` contains the following values:

```
["four", "score", "and", "seven", "years", "ago"]
```

In the first pair of Strings ("four" and "score") the shorter String is "four". In the second pair of Strings ("and" and "seven") the shorter String is "and". In the third pair of Strings ("years" and "ago") the shorter string is "ago". Therefore, the call:

```
removeShorterStrings(list);
```

should remove these shorter Strings, leaving the list with the following sequence of values after the method finishes executing:

```
["score", "seven", "years"]
```

If there is a tie (both Strings have the same length), your method should remove the first String in the pair. If there is an odd number of Strings in the list, the final value should be kept in the list. For example, if the list contains the following values:

```
["to", "be", "or", "not", "to", "be", "hamlet"]
```

After calling removeShorterStrings, it should contain the following:

```
["be", "not", "be", "hamlet"]
```

8. Critters, 15 points

Write a class called `Pigeon` that extends the `Critter` class. The instances of the Pigeon class always hop when possible and otherwise randomly choose between turning left and turning right, with each choice being equally likely. Their appearance changes over time. Each Pigeon initially displays as an asterisk ("*"). Then as each Pigeon chooses a move, it changes its appearance to match that move. If it's most recent move was a hop, it displays as "H". If it's most recent move was to turn left, it displays as "L". And if it's most recent move was to turn right, it displays as "R". Its color should be the default color for critters.

9. Arrays, 14 points

Write a static method called append that takes two integer arrays as parameters and that returns a new array that contains the result of appending the values of the second array at the end of the first array. For example, suppose that arrays list1 and list2 have been declared as follows:

```
int[] list1 = {2, 4, 6};
int[] list2 = {1, 2, 3, 4, 5};
```

If we make the following call:

```
append(list1, list2)
```

The method will return a new array that contains the following values:

```
[2, 4, 6, 1, 2, 3, 4, 5]
```

If the call instead had been:

```
append(list2, list1)
```

Then the method would return a new array that contains:

```
[1, 2, 3, 4, 5, 2, 4, 6]
```

Your method should not change either of the arrays passed as parameters.

10. Programming, 10 points

Write a static method called `isUnique` that takes an array of integers as a parameter and that returns a `boolean` value indicating whether or not the values in the array are unique (true for yes, false for no). The values in the list are considered unique if there is no pair of values that are equal. For example, if a variable called `list` stores the following values:

```
[3, 8, 12, 2, 9, 17, 43, -8, 46, 203, 14, 97, 10, 4]
```

then the call:

```
isUnique(list)
```

should return true because there are no duplicated values in this list. If instead the list stored these values:

```
[4, 7, 2, 3, 9, 12, -47, -19, 308, 3, 74]
```

then the call should return false because the value 3 appears twice in this list. Notice that given this definition, a list of 0 or 1 elements would be considered unique.