Computer Science 51
Spring 2006

Final Examination

Question	Points	Score	Description
1	10		Arrays & Strings
2	10		Two-dimensional Arrays
3	15/10		Recursion
4	9		Debugging & Efficiency
5	6		Inheritance
6	5		Streams
7	5		Debugging
8	10		Linked Lists
9	10		Searching and Sorting
total	80		

This examination is closed book. You have 3 hours to complete the exam.

Your Name (Please print) ______________________________

1. Arrays and Strings (10 pts)

I would like you to write two methods that will allow us to count up the number of occurrences of all of the letters in a string. In writing these methods recall that the Java String class has method `indexOf(target,start)` which returns the first occurrence of `target` in the source string starting with index `start`, and method `toLowerCase()` which converts all characters to their lower case equivalent, Recall also that if `letter` is a `char` variable, then `letter++` increments the value of `letter` to the next character in alphabetical order.

a. First write a method `numOccurs(letter,source)` which returns the number of times that a character named `letter` occurs in `source.`

```
public int numOccurs(char letter, String source)
```

b. Now write a method `countLetters(source)` which returns an array with 26 entries that contains the number of occurrences of each letter 'a' through 'z' in source. That is, if the method returns an array, `count`, then `count[0]` is the number of occurrences of 'a' and 'A' in `source`, `count[1]` is the number of occurrences of 'b' and 'B' in `source`, etc. Use method `numOccurs` above to simplify your code.

```
public int[] countLetters(String source)
```

2. Multi-dimensional arrays (10 pts)

The distance between two cities can be represented as a two dimensional array. The index into each dimension represents a city. The value in a cell (indexed by two cities) represents the distance between those cities. For example, we might have:

	Claremont	Boston	New York	San Francisco
Claremont	0	2956	2749	408
Boston	2956	0	213	3104
New York	2749	213	0	2910
San Francisco	408	3104	2910	0

A trip can be represented as a one dimensional array of cities that contains each of the cities visited, with the starting city in the 0th slot and the last city visited in the last slot. Assume you have the following declarations:

```
public static final int CLAREMONT = 0;
public static final int BOSTON = 1;
public static final int NEW_YORK = 2;
public static final int SAN_FRANCISCO = 3;
private int distances [][];
```

Assume that `distances` is initialized to contain the distances shown in the table above using the city constants above as indices. Write a method `tripDistance` that returns the total distance of a trip. The signature of the method should be:

```
public int tripDistance (int[] trip, int numCitiesInTrip)
```

3. **Recursion** **(15 Points)**

For this question, I would like you to use recursion. The good news is that I will let you choose which of two different programs you write. The bad news is that I won't give you as many points for what I think is the easier problem as I will if you solve the harder problem.

The first problem (a) is what I consider the easier problem. A correct solution is worth a maximum of 10 points. A correct solution to the second problem (b) is worth up to 15 points. If you include work on more than one of these problems, please make it very clear which of the two you want me to grade.

(a) An infinite series such as

1 - 2 + 3 - 4 + 5 - 6 + . . .

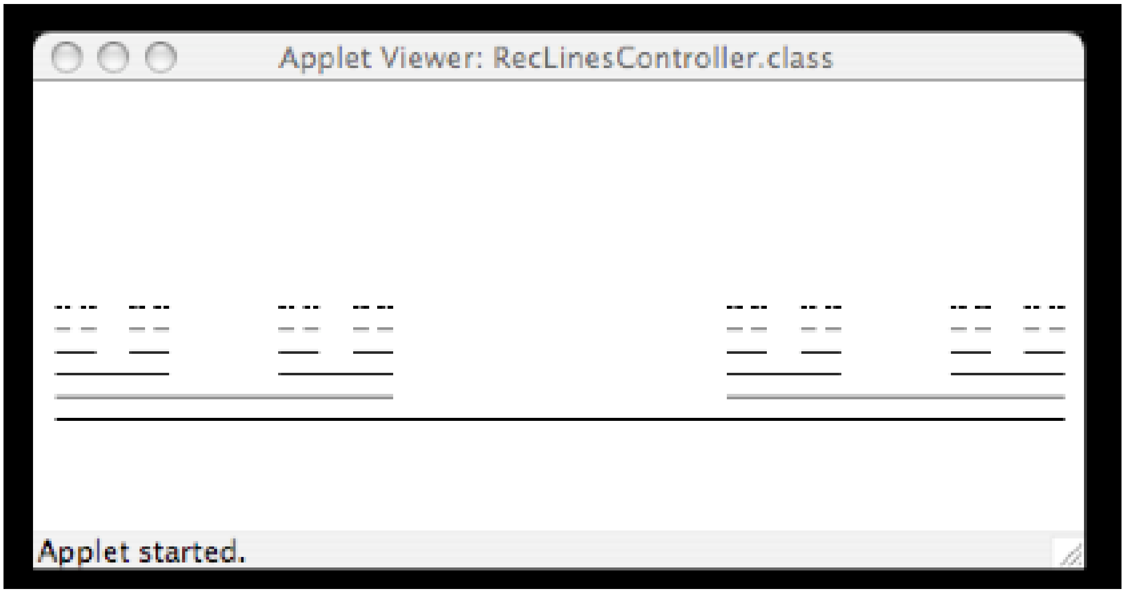
is called an alternating series.

In the case of 1 - 2 + 3 - 4 + 5 - 6 …, the sequence of partial sums is 1, -1, 2, -2, 3, -3, …

Write a recursive method that calculates the nth partial sum, given a positive integer n.

(b) Write the classes necessary to draw the picture below. The picture is formed by first drawing a horizontal line starting at (x,y) with length size. If the size is greater than MIN then draw two new instances of the picture, each of which is 1/3 of the original size and offset vertically by DIFF pixels from the original. The first should start at the left end of the horizontal line, while the second ends at the right end of the horizontal line. If the size is less than MIN, then simply draw two horizontal lines that are 1/3 of the size of the original and offset vertically by DIFF pixels from the original. As in the general case, one should start at the left end of the horizontal line, while the other ends at the right end.

As well as writing the constructor, please write a move method that moves the entire picture..



The interface and calling class are provided below for you. The result should be a recursive solution to the problem. In particular, there should be no loops in any of your classes

```
public class RecLinesController extends WindowController{
  private Location lastPoint; // where mouse pressed last
  private RecLinesIfc lines; // picture
  public void begin() {
     lines = new RecLines(10,150,450,canvas);
  }
  ...
}

public interface RecLinesIfc {
  // move picture by dx in x-direction & dy in y-direction
  void move(double dx, double dy);
}
```

```
public class BaseRecLines implements RecLinesIfc {



  // Draw horizontal line with length size starting at (x,y)
  public BaseRecLines(double x, double y, double size,
                           DrawingCanvas canvas) {



  }
  // move picture by dx in x-direction and dy in y-direction
  public void move(double dx, double dy) {



  }
}
```

```
public class RecLines implements RecLinesIfc {
   private static final double MIN = 9.0;
   private static final double DIFF = -10;



   // Draw horizontal line of length size starting at (x,y)
   // If size > MIN draw two RecLines with length
   // size/3, one at each end of original line and each DIFF
   // above the original line.  If size < MIN then construct
   // two base cases at same locations and sizes.
   public RecLines(double x, double y, double size,
                         DrawingCanvas canvas) {



   }

   // move picture by dx in x-direction and dy in y-direction
   public void move(double dx, double dy) {



   }
}
```

4. **Debugging & Efficiency (9 pts)**
Let dateList be an array of Date, where Date is a class that includes methods getMonth() and getDay(), both of which return integers. If today is a variable of type Date which represents the scheduled date of this exam, then today.getMonth() would return 5 and today.getDay() would return 10.

Dates date1 and date2 are in order if the month of date1 precedes the month of date2 or their months are the same, and the day of date1 is less than or equal to the day of date2. We have included below the code for a boolean method inOrder() which determines if the elements of DateList are in order.

The code below has three logical (*not syntax*) errors. Please find and correct them. You may assume that you are given the following declarations of instance variables:

```
private Date[] dateList; // dates in list
                         // - initialized elsewhere
private int numEntries;  // # of entries in dateList

public boolean inOrder()
{
   boolean soFarInOrder = true;
   for (int index = 0; index < numEntries; index++)
   {
      Date now = dateList [index];
      Date next = dateList [index++];
      if (now.getMonth() > next.getMonth())
         soFarInOrder = false;
      else if (now.getMonth() == next.getMonth() &&
               now.getDay() > next.getDay())
         soFarInOrder = false;
      else
         soFarInOrder = true;
   }
   return soFarInOrder;
}
```

b.   The code on the previous page is very inefficient in many cases. Please rewrite it below so that it performs as few comparisons as possible.

5. Inheritance (6 pts.)

Use the following class definitions to answer the question below.

```
public class Bird {
     public void moves () {
          System.out.println (“flies.”);
     }
}

public class RBird extends Bird {
}

public class PBird extends Bird {
     public void moves() {
          System.out.println (“swims.”);
     }
}

public class DBird extends Bird {
     public void moves() {
          System.out.print (“swims and ”);
          super.moves();
     }
}
```

With the class definitions above, what is the output produced by the following method:

```
public void birdsMove() {
     Bird b;
     b = new RBird();
     System.out.print (“An RBird “);
     b.moves();

     b = new PBird();
     System.out.print (“A PBird “);
     b.moves();

     b = new DBird();
     System.out.print (“A DBird “);
     b.moves();
}
```

6. Streams (5 pts.)

The following method is supposed to copy the contents of an input stream into an output stream, one line at a time. The method compiles and runs but it does not work correctly. For example, consider the following input:

Q: How many Bill Gates' does it take to change a light bulb?

A: One. He puts the bulb in and lets the world revolve around him.

Q: How many Microsoft engineers does it take to change a light bulb?

A: None. Bill Gates will just redefine Darkness(TM) as the new industry standard.

Unfortunately, the output only shows the answers, not the questions. There is one error in this method. Find and fix it.

```
public void copy (BufferedReader inStream,
                  PrintWriter outStream)
{
     try
     {
          while (inStream.readLine () != null)
          {
               outStream.println (inStream.readLine());
          }
     catch (IOException e)
     {
          System.out.println (e);
     }
}
```

7. **Debugging (5 pts.)**

The drawing portion of Pictionary allowed the user to draw outside the rectangular frame that was drawn on the screen. Here is an attempt to prevent that happening. Unfortunately, it doesn't quite work. When I run it, I sometimes get the following output in the Java console. Find and fix the error. Here is the console message:

```
Exception occurred during event dispatching:
java.lang.NullPointerException
     at objectdraw.Line.<init>(Line.java)
     at Scribble.onMouseDrag(Scribble.java)
     at objectdraw.WindowController.mouseDragged
        (WindowController.java)
     at java.awt.Component.processMouseMotionEvent
        (Component.java)
     at java.awt.Component.processEvent(Component.java)
     at java.awt.Component.dispatchEventImpl(Compiled Code)
     at java.awt.Component.dispatchEvent(Compiled Code)
     at java.awt.EventDispatchThread.run
        (EventDispatchThread.java)
```

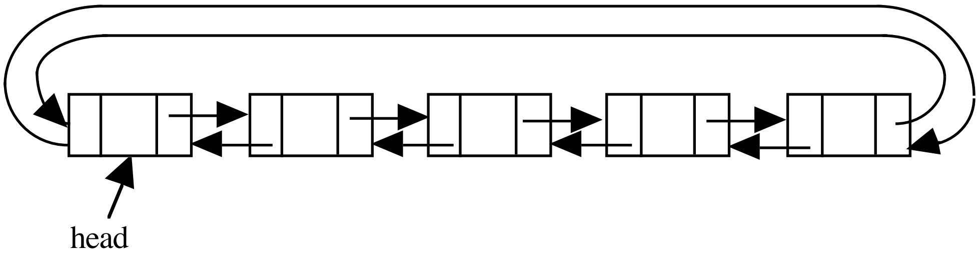
Here is the program:

```
import objectdraw.*;
import java.awt.*;

public class Scribble extends WindowController{
     private static final int SIZE = 200;
     private static final Coords TOP_LEFT = new Coords (10, 10);
     private Coords nextLineStarts;
     private FramedRect frame;

     protected void begin (){
        frame = new FramedRect (TOP_LEFT, SIZE, SIZE, canvas);
     }

     protected void onMousePress(Coords point){
        if (frame.contains (point)) {
            nextLineStarts = point;
     }   }

     protected void onMouseDrag(Coords point){
        if (frame.contains (point)) {
            new Line(nextLineStarts, point, canvas);
            nextLineStarts = point;
        }
     }

   }
```

8. A circular doubly linked list holding elements of type E has instance variables:

```
protected DblyLinkedListElement<E> head;
protected int count;
```

A circular doubly linked list with five elements is represented as in the picture below:



This question revolves around the implementation of the method to remove an element from the list:

// post: remove first element equal to val in list. Return true if found and false otherwise

public bool remove(E val)

a. What special case(s) must you worry about in writing this method?

b. Please write the Java code for this method on the next page. Do not use other methods of the DblyLinkedList class. Do not forget to update `count`. *(Recall that* `DblyLinkedListElement<E>` *has methods* `previous()`, `next()`, `value()`, `setPrevious(DblyLinkedListElement<E> prev)`*, and* `setNext(DblyLinkedListElement<E> nextOne)`*.) Draw a picture to help you write the code.*

(Answer to problem 8b)

9. **Searching and Sorting (10 pts)**

a. Recall that **insertion** sort works by first sorting all but one of the elements of the array. It then finishes by inserting the remaining element where it belongs among the already sorted elements of the array. Suppose that sorting an array of 1,000 integers takes 2 seconds. Approximately how long will it take to sort 2,000 elements using **insertion** sort?

Approximately how long will it take to sort 10,000 elements using **insertion** sort?

b. In a **linear** search you begin looking for the target element at the beginning of the array and compare with each successive element until you find the target element or until you run out of elements. In a **binary** search of a sorted list you first compare the target element with the middle element of the array. Based on that, you throw out either the first or the second half the elements as being too large or too small to hold the target element. Then do a binary search for the target on the remaining elements.

Roughly how many comparisons must be made in the worst case if you are performing a **linear** search on an array with 1,000,000 elements?

… with 2,000,000 elements?

Roughly how many comparisons must be made in the worst case if you are performing a **binary** search on an array with 1,000,000 elements?

… with 2,000,000 elements?

[Hint: 1,000,000 ≈ 2^{20}.]