

**Computer Science 51**  
**Spring 2006**

Final Examination

Question	Points	Score	Description
1	10		Arrays & Strings
2	10		Matrices
3	15/10		Recursion
4	9		Debugging & Efficiency
5	6		Inheritance
6	5		Streams
7	5		Debugging
8	10		<u>Linked Lists</u>
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 `String` class has method `indexOf(target) startingAt (start)` which returns the first occurrence of `target` in the source string starting with index `start`, and method `asLower`, 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) in (source)` which returns the number of times that a `String` named `letter` occurs in `source`. You may assume that `letter` has length 1.

```
method numOccurs(letter: String) in (source: String)
    → Number
```

- b. Now write a method `countLetters(source)` which returns a list 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.at(1)` is the number of occurrences of 'a' and 'A' in `source`, `count[2]` is the number of occurrences of 'b' and 'B' in `source`, etc. Use method `numOccurs` above to simplify your code. Hint: You may assume you are given a string `alphabet = "abcdefghijklmnopqrstuvwxyz"`.

```
method countLetters(source: String) → List[[Number]]
```

## 2. Matrices (10 pts)

The distance between two cities can be represented as a Matrix. 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 first<sup>th</sup> slot and the last city visited in the last slot. Assume you have the following declarations:

```
def claremont: Number = 1
def boston: Number = 2
def newYork: Number = 3
def sanFrancisco: Number = 4
def distances: Matrix[[Number]];
```

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 header of the method should be:

```
method tripDistance (trip: List[[Number]]) → Number
```

### 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

$$1 - 2 + 3 - 4 + 5 - 6 + \dots$$

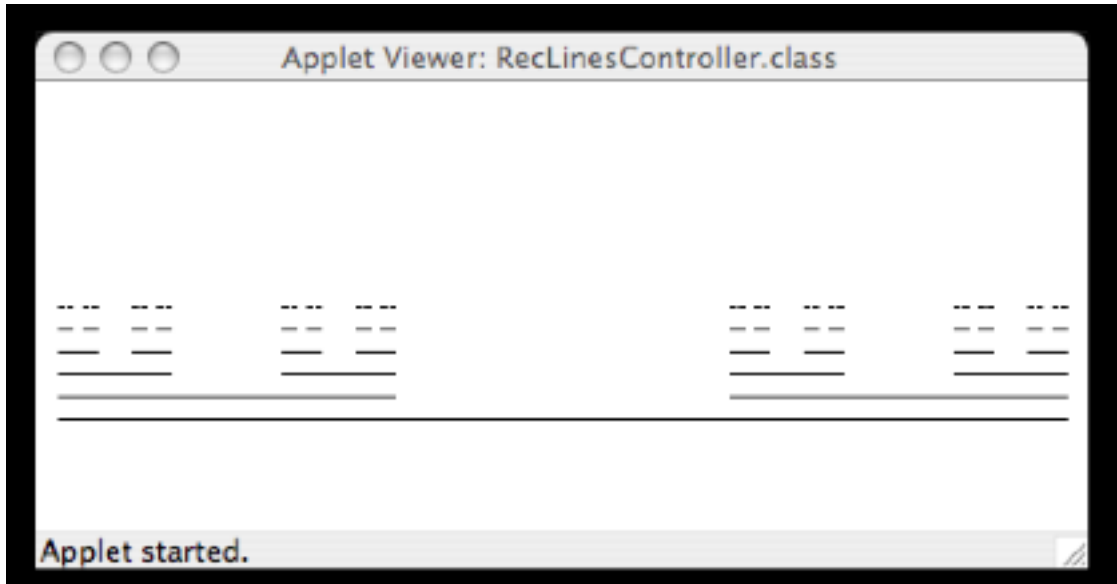
is called an alternating series.

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

Write a recursive method that calculates the  $n$ th 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 `moveBy` 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

```
def recLinesApp = object{
  inherit graphicApplicationSize(600 @ 600)
  var lastPoint: Point // where mouse pressed last
  // picture
  def lines: RecLinesType = recLinesAt(10@150) size (450)
                                on (canvas)
}
...
}

type RecLinesType = {
  // move picture by dx in x-direction & dy in y-direction
  moveBy(dx: Number, dy: Number) → Done
}
```

```
class BaseRecLinesAt(pt: Point) size (size':Number)
    on (canvas: DrawingCanvas) → RecLinesType {
```

```
    // move picture by dx in x-direction and dy in y-direction
    moveBy(dx: Number, dy: Number) → Done {

    }
}
```

```

// 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.
class RecLinesAt(pt: Point) size (size': Number)
    on (canvas: DrawingCanvas) → RecLinesType {

    def min: Number = 9.0
    def diff: Number = -10;

}

// move picture by dx in x-direction and dy in y-direction
moveBy(dx: Number, dy: Number) → Done {

}
}

```



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 two logical (*not syntax*) errors. Please find and correct them.

```
method inorder(dateList: List[[Date]]) → Boolean {
  var soFarInOrder: Boolean := true
  for (1 .. dateList.size) do {index: Number →
    def now: Date = dateList.at(index)
    def next: Date = dateList.at(index + 1)
    if (now.month > next.month) then {
      soFarInOrder := false;
    } elseif {(now.month == next.month) &&
              (now.day > next.day)} then {
      soFarInOrder := false;
    } else {
      soFarInOrder := true;
    }
  }
  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.

```
type Animal = {
  moves → Done
}

class bird → Animal {
  method moves { print ("flies.")}
}

class rBird → Animal {
  inherit bird
}

class pBird → Animal {
  inherit bird
  method moves {
    print ("swims.")
  }
}

class dBird → Animal {
  inherit bird
  alias oldMoves = moves
  method moves {
    print ("swims and ")
    oldMoves
  }
}
```

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

```
method birdsMove → Done {
  var b: Animal
  b := rBird
  print ("An RBird ")
  b.moves

  b := pBird
  print ("A PBird ")
  b.moves

  b := dBird
  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 always drops the last line during the copy (e.g., the last "A" above). Find the error and fix it. Don't worry about not having the try-catch. That is not the problem.

```
import "io" as inout
method copyFrom (infile: inout.FileStream) to (outfile:
inout.FileStream) → Done{
    var nextLine: String := infile.getline
    while {!infile.eof} do {
        print "writing"
        outfile.write(nextLine++"\n")
        nextLine := infile.getline
    }
}
```

## 7. Debugging (5 pts.)

I want to write a variant of the scribble program where I prevent the user from drawing outside of a framed rectangle drawn on the screen. Here is an attempt to prevent that from happening.

```
def scribble: GraphicApplication = object {
  inherit graphicApplicationSize(400 @ 400)

  def frame: Graphic2D = framedRectAt(10 @ 10) size (380 @
380) on (canvas)

  var nextLineStarts: Point

  method onMousePress(point: Point) → Done {
    if (frame.contains(point)) then {
      nextLineStarts := point
    }
  }

  method onMouseDrag (point: Point) → Done {
    if (frame.contains(point)) then {
      lineFrom(nextLineStarts) to (point) on (canvas)
      nextLineStarts := point
    }
  }

  startGraphics
}
```

Unfortunately, it doesn't quite work. When I run it, I sometimes get the following output in the Grace console when I start dragging. Find and fix the error. Here is the console message:

```
UninitializedVariable on line 18 of OutOfBounds: attempt to read
uninitialised variable nextLineStarts.
  raised at scribble.nextLineStarts at line 18 of OutOfBounds
  requested from scribble.onMouseDrag(_) at line 18 of OutOfBounds
  requested from block.apply(_) at line 1317 of rtobjectdraw

17:         if (frame.contains(point)) then {
18:             lineFrom(nextLineStarts) to (point) on (canvas)
19:             nextLineStarts := point
Stack frames:
onMouseDrag(1)
  point = (390@334)
```

As you can see, line 18 is in the onMouseDrag method. Find and fix the error.

7. *Not relevant*

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

a. Recall that **selection** sort works by first moving the largest element to the end of the list. It then finishes by sorting the remaining elements. Suppose that sorting an array of 1,000 integers takes 2 seconds. Approximately how long will it take to sort 2,000 elements using **selection** sort?

Approximately how long will it take to sort 10,000 elements using **selection** 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:  $1,000,000 \approx 2^{20}$ ]