CS 102  
Fall 2008  
Instructor: Audrey St. John

Practice Midterm Exam

General description

There are 4 problems, some subdivided into smaller questions. Do as many as you can, or as much of a bigger problem as you can. There should be enough time to finish, or at least to start working on all the questions. Feel free to use extra paper, but be sure to put your name and which question is being answered at the top.

Start with the ones that you consider to be easiest, then move to those that may take more time.

Be generous with comments. Part of the grade will be based on how readable and neat your code is.

1. **Basic programming concepts (30 points)**
   a. What is the following (partial) program doing? Explain in a few short sentences.

   ```javascript
   var data : Array;
   var n : int;
   var flag : int;

   ... // the flag variable gets a value somewhere in here ...

   n = 10;
   data = new Array(n);
   data[0] = 1;

   if ( (flag > 5) && (flag < 5) )
   {
     for ( var i : int = 1; i < n; i++ )
     {
       data[i] = i + data[i-1];
     }
   }
   else
   {
     for ( var i :int = 1; i < n; i++ )
     {
       data[i] = 2*data[i-1];
     }
   }
   ```
b. Implement a method (using the following skeleton) that takes in a two-dimensional Array full of Boolean values and returns true if there is a full row, column or diagonal of true values. You may define auxiliary methods that you find helpful.

```java
public function checkTicTacToe( board : Array ) : int {
```

c. Debug the following (or indicate that it has no bugs). If you find a bug, suggest a solution that would fix it.

```javascript
package {
    public class Mystery {
        private var array : Array;
        private var s : String;

        public function Mystery() {
            array = new Array(100);

            for ( var i : int = 0; i < array.length; i++ ) {
                array[i+1] = i.toString();
            }

            s = "Boo! " + array[0];
        }
    }
}
```
2. **Arrays and loops (20 points)**

The Catalan numbers arise in many applications and appear in the Catalan triangle. The first 7 rows of the Catalan pyramid is given below:

```
1
1 1
1 2 2
1 3 5 5
1 4 9 14 14
1 5 14 28 42 42
1 6 20 48 90 132 132
```

The first column is all 1s. Subsequent entries are computed by adding the values of entries above and to the left. If there is no entry above, the entry is simply the value of the entry to the left. Write a method that computes the first \(n\) rows of the Catalan triangle and returns a two-dimensional array holding the result.
3. Objects (30 points)
In this problem, you will create a \texttt{Pie} class. Every pie (instance) should contain a type (such as "apple," "cherry," "pecan") and a flag as to whether or not the pie contains fruit.
   a. Define the \texttt{Pie} class; your code should have fewer than 5 lines (not including comments).
   
   b. The \texttt{Pie} class definition should be in a file named ______________________.
   
   c. Write a \texttt{Pie} constructor that takes in two arguments, for the type and whether or not the pie contains fruit. Your code should have fewer than 5 lines (not including comments).
d. Write the code for a `printHealthInformation` method, which prints to the command line (trace) health information, such as “This apple pie is healthy as it contains fruit” or “This pecan pie is not healthy as it contains no fruit.” Your code should have fewer than 10 lines (not including comments).

e. Write the code for a `toString` method, which returns a String representation of the pie, such as “Yummy apple pie.” Your code should have fewer than 5 lines (not including comments).

f. Write a statement that declares a variable of type `Pie`.

g. Write a statement that creates (constructs) a `Pie` instance that represents an apple pie, (which contains fruit…), and assigns it to the variable declared in Part e.

h. Write a statement that simultaneously declares a variable of type `Pie` and assigns to it a new `Pie` instance that represents a pecan pie, which does not contain fruit.
i. Write a statement that simultaneously declares and allocates an array of length 3.

j. Write code to fill the array from step h with the apple pie, pecan pie, and another pie of your choosing. Your code should have fewer than 5 lines (not including comments).

k. Write code to print the health information for each item in the array filled by step i. Your code should have fewer than 5 lines (not including comments).
4. Inheritance (20 points)
Using the Pie class from Problem 3, create a subclass PieALaMode that represents a pie with ice cream. In this problem, you will create a PieALaMode class that subclasses the Pie class you defined in Problem 3. Every pie a la mode (instance) should contain (in addition to the basic pie properties) a type of ice cream (such as “vanilla,” “strawberry,” “chocolate”) and a flag as to whether or not the ice cream contains fruit.

a. Define the PieALaMode class; your code should have fewer than 5 lines (not including comments).

b. Override the inherited method printHealthInformation to print health information after taking the ice cream type into account. For instance, “This apple pie with vanilla ice cream is healthy as it contains fruit” or “This pecan pie with strawberry ice cream is healthy as it contains fruit” or “This pecan pie with vanilla ice cream is not healthy as it contains no fruit.” Your code should have fewer than 5 lines (not including comments).
c. Override the inherited method `toString` to display the type of ice cream as well as the original pie information, i.e., “Yummy apple pie with vanilla ice cream.” Your method definition should not explicitly use the word “Yummy.”