COSC 1046 EL01/EL02 FINAL EXAM
INTRODUCTION TO COMPUTER SCIENCE I

Thursday, December 18, 2:00 pm, 2003

Time Allowed: 3 hours  Instructors: Osman Abou-Rabia (EL01), Barry G. Adams (EL02)

Name (PLEASE PRINT) ________________________________________________________________

Student # ______________________________________________________________________

1. Answer ALL questions. Write your answers on this questionnaire.
2. Use back of exam pages if necessary.
3. A class summary is provided on the last two pages.
4. Number of Questions: 7
5. Total Marks: 60
Question 1 (12 marks)

(a) Given the Point class specification on page 10, write a Line class that represents a line between two points \((x_1, y_1)\) and \((x_2, y_2)\) as two Point objects. Given the following class specification write the complete class.

```java
public class Line {
    private Point p1, p2;
    public Line(Point p1, Point p2) {...}
    public Line(double x1, double y1, double x2, double y2) {...}
    public Point getP1() {...}
    public Point getP2() {...}
    public double length() {...}
    public String toString() {...}
}
```

Here there are two constructors: one specifies the line using two points and the other specifies the line using the \(x\) and \(y\) coordinates of two points. The \texttt{length} method uses the formula \(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\) to calculate and return the length of the line.

\textbf{Answer:}
(b) Write a tester class called \texttt{LineTester} that contains a \texttt{doTest} method to test the \texttt{Line} class. Test the class by using a \texttt{KeyboardReader} object (see page 11) to read the four double numbers for the $x$ and $y$ coordinates of the line end points. Then construct two \texttt{Line} objects, one for each constructor. Then, using one of these objects, display the results returned by all the methods in the \texttt{Line} class. Also show in the tester class how to use the \texttt{getX} and \texttt{getY} methods in the \texttt{Point} class to display the coordinates of the end points of one of the \texttt{Line} objects.

\textbf{Answer:}
Question 2 (10 marks)

Write the statements for the `paintComponent` method that draw the bow tie shown in the picture. First fill each diamond with green and draw them with a black outline. Then fill the two circles with red with no outline. Each circle should have a diameter of 40 units. Finally, fill the square in the center with yellow and draw it with a red outline. The square should have a side length of 50 units.

Assume that the drawing area is 401 pixels wide (0 to 400.0) and 301 pixels high (0 to 300.0). Use a stroke of `2.0f` to draw lines. It is not necessary to make the picture expand or contract if the window is resized.
Question 3 (10 marks)

A real estate agent gets a commission based on the selling price $p$ of a house according to the table:

- $0 \leq p < 100,000$ \hspace{1cm} 3%
- $100,000 \leq p < 250,000$ \hspace{1cm} 5%
- $250,000 \leq p < 500,000$ \hspace{1cm} 7%
- $500,000 \leq p$ \hspace{1cm} 10%

Write a complete class called `CommissionCalculator` that calculates the commission given the selling price. Do all calculations in a `doCalculations()` method that is called in the constructor. Use get methods to return the selling price and the commission. Also include a set method to change the selling price and recalculate the commission. If an attempt is made to use a negative selling price an `IllegalArgumentException` should be thrown.

**Answer:**
Question 4 (8 marks)

(a) Write a method with prototype `public void drawTriangle(int n)` that draws triangles using asterisks. Here `n` is the number of rows. For example, the result for 4 rows is given by

```
*  
*** 
*****
*******
```

Answer:

(b) Write a recursive method with prototype `public String reverse(String s)` that returns a string that is the reverse of `s`. For example the reverse of `Help` is `pleH`. [HINT: Think of a string as a head (substring consisting of the first character) and a tail (substring consisting of all but the first character). See `String` class on page 10.]

Answer:
Question 5 (8 marks)

(a) Write a method with prototype

```java
public double averageForOneStudent()
```

that uses a `KeyboardReader` object (see page 11) to read marks in the range 0 to 100 as `double` numbers in a while loop and compute their average. Use a negative mark as a sentinel value to indicate that there are no more marks to be entered for this student. Return the average as the value of the method.

**Answer:**

(b) Call this method in a for loop that processes a class of 10 students. Display the average for each student and when the for loop exits also display the class average.

**Answer:**
Question 6 (6 marks)

Write a linear search method with prototype

    public int linearSearch(BankAccount[] b, String name)

that searches the given array of BankAccount objects for one whose owner name is given by name. The method returns the array index if the account is found, otherwise it returns -1.

Answer:
Question 7 (6 marks)

Given the following bubble sort method for an array of double numbers

```java
public void bubbleSort(double[] a)
{
    int n = a.length;
    for (int p = 1; p <= n - 1; p++)
    {
        for (int j = 0; j <= n - 1 - p; j++)
        {
            if (a[j] > a[j + 1])
            {
                double temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }
}
```

(a) What changes are necessary to sort an array of String objects in increasing lexicographical order?

Answer:

(b) What changes are necessary to sort an array of BankAccount objects in increasing order by owner name. (See page 10 for the specification of the BankAccount class.).

Answer:

(c) What changes are necessary to sort an array of BankAccount objects in decreasing order by balance. (See page 10 for the specification of the BankAccount class.).

Answer:
Brief Documentation for Classes

Some String class method prototypes

// return substring with characters from index ind1 to index ind2 of this string
// For example s.substring(3,8) returns characters from index 3 to 7
public String substring(int ind1, int ind2)

// return substring with characters from index ind to the end of this string.
// For example s.substring(5) returns characters from index 5 to the end of s.
public String substring(int ind)

// return the length of a string
public int length()

// return the lowercase version of a string
public String toLowerCase()

// return the uppercase version of a string
public String toUpperCase()

// return true if s has the same characters as this string.
// For example, s1.equals(s2) returns true if s1 and s2 have the same characters
public boolean equals(String s)

// return result of comparing this string and s
// For example, s1.compareTo(s2) will be negative if s1 precedes s2, 0 if
// s1 is equal to s2, and positive if s1 follows s2
public int compareTo(String s)

The Point class constructor and method prototypes

// construct a point object with given x, y coordinates
public Point(double x, double y)

// return the x-coordinate of the point
public double getX()

// return the y-coordinate of the point
public String getY()

// return a string representation of a point in form (x,y)
public String toString()

The BankAccount class constructor and method prototypes

// construct account with account number a, owner name n and initial balance b
public BankAccount(int a, String n, double b)

// withdraw given amount (return true if valid)
public boolean withdraw(double amount)

// deposit given amount
public void deposit(double amount)

// get the values of the three data fields
public int getNumber()
public String getName()
public double getBalance()
The KeyboardReader class constructor and method prototypes

    // construct a KeyboardReader object
    public KeyboardReader()
    // read a line as an integer and return it
    public int readInt()
    // read a line as a double and return it
    public double readDouble()
    // read a line as a string and return it
    public String readLine()