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 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.

Answer:

```java
public class Line
{
    private Point p1, p2;
    public Line(Point p1, Point p2)
    {
        this.p1 = p1;
        this.p2 = p2;
    }
    public Line(double x1, double y1, double x2, double y2)
    {
        p1 = new Point(x1,y1);
        p2 = new Point(x2,y2);
    }
    public Point getP1() { return p1; }
    public Point getP2() { return p2; }
    public double length()
    {
        double dx = p2.getX() - p1.getX();
        double dy = p2.getY() - p1.getY();
        return Math.sqrt(dx*dx + dy*dy);
    }
    public String toString()
    {
        return "Line[" + p1 + "," + p2 + "]";
    }
}
```
(b) Write a tester class called `LineTester` that contains a `doTest` method to test the `Line` class. Test the class by using a `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 `Line` objects, one for each constructor. Then, using one of these objects, display the results returned by all the methods in the `Line` class. Also show in the tester class how to use the `getX` and `getY` methods in the `Point` class to display the coordinates of the end points of one of the `Line` objects.

**Answer:**

```java
public class LineTester {
    public void doTester() {
        KeyboardReader in = new KeyboardReader();
        System.out.println("Enter x,y for first point");
        double x1 = in.readDouble();
        double y1 = in.readDouble();
        System.out.println("Enter x,y for second point");
        double x2 = in.readDouble();
        double y2 = in.readDouble();
        Point p1 = new Point(x1, y1);
        Point p2 = new Point(x2, y2);
        Line l1 = new Line(p1, p2);
        Line l2 = new Line(x1, y1, x2, y2);
        System.out.println("p1 = " + l1.getP1());
        System.out.println("p2 = " + l2.getP2());
        System.out.println("length = " + l1.length());
        System.out.println("l1 = (" + l1.getP1().getX() + "," + l1.getP1().getY() + ")-(" + l1.getP2().getX() + "," + l1.getP2().getY() + ")");
    }
}
```
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.

```java
GeneralPath leftBow = new GeneralPath();
leftBow.moveTo(100.0f, 0.0f);
leftBow.lineTo(200.0f, 150.0f);
leftBow.lineTo(100.0f, 300.0f);
leftBow.lineTo(0.0f, 150.0f);
leftBow.closePath();

GeneralPath rightBow = new GeneralPath();
rightBow.moveTo(300.0f, 0.0f);
rightBow.lineTo(400.0f, 150.0f);
rightBow.lineTo(300.0f, 300.0f);
rightBow.lineTo(200.0f, 150.0f);
rightBow.closePath();

Ellipse2D.Double leftButton = new Ellipse2D.Double(80,130,40,40);
Ellipse2D.Double rightButton = new Ellipse2D.Double(280,130,40,40);
Rectangle2D.Double centerButton = new Rectangle2D.Double(175,125,50,50);

g2D.setStroke(new BasicStroke(2.0f));
g2D.setPaint(Color.green); g2D.fill(leftBow);
g2D.setPaint(Color.black); g2D.draw(leftBow);
g2D.setPaint(Color.green); g2D.fill(rightBow);
g2D.setPaint(Color.black); g2D.draw(rightBow);
g2D.setPaint(Color.red); g2D.fill(leftButton); g2D.fill(rightButton);
g2D.setPaint(Color.yellow); g2D.fill(centerButton);
g2D.setPaint(Color.red); g2D.draw(centerButton);
```
Question 3 (10 marks)

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

- $0 \leq p < 100,000$ : 3%
- $100,000 \leq p < 250,000$ : 5%
- $250,000 \leq p < 500,000$ : 7%
- $500,000 \leq p$ : 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:

```java
public class CommissionCalculator
{
    public double sellingPrice, commission;

    public CommissionCalculator(double price)
    {
        if (price < 0.0)
            throw new IllegalArgumentException("Invalid price");
        sellingPrice = price;
        doCalculations();
    }

    public void doCalculations()
    {
        if (0 <= sellingPrice && sellingPrice < 100000)
            commission = sellingPrice * 0.03;
        else if (100000 <= sellingPrice && sellingPrice < 250000)
            commission = sellingPrice * 0.05;
        else if (250000 <= sellingPrice && sellingPrice < 500000)
            commission = sellingPrice * 0.07;
        else
            commission = sellingPrice * 0.10;
    }

    public double getSellingPrice()
    {
        return sellingPrice;
    }

    public double getCommission()
    {
        return commission;
    }

    public void setSellingPrice(double p)
    {
        sellingPrice = p;
        doCalculations();
    }
}
```
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:

```java
public void drawTriangle(int n) {
    for (int row = 1; row <= n; row++) {
        for (int spaces = 1; spaces <= n - row; spaces++) {
            System.out.print(" ");
        }
        for (int star = 1; star <= 2*row - 1; star++) {
            System.out.print("*");
        }
        System.out.println();
    }
}
```

(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:

```java
public String reverse(String s) {
    if (s.length() == 0)
        return s;
    else
        return reverse(s.substring(1)) + s.substring(0,1);
}
```
Question 5 (8 marks)

(a) Write a method with prototype

    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:**

```java
public double averageForOneStudent()
{
    KeyboardReader input = new KeyboardReader();
    System.out.println("Enter marks terminated by a negative mark");
    double sum = 0.0;
    int numMarks = 0;
    double mark = input.readDouble();
    while (mark >= 0)
    {
        sum = sum + mark;
        numMarks = numMarks + 1;
        mark = input.readDouble();
    }
    return sum / (double) numMarks;
}
```

(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:**

```java
double sum = 0.0;
for (int k = 1; k <= 10; k++)
{
    double average = averageForOneStudent();
    System.out.println("Average for student "+k+" is "+average);
    sum = sum + average;
}
System.out.println("Class average is "+sum/10.0);
```
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:

```
public int linearSearch(BankAccount[] b, String name)
{
    int index = 0;
    int n = b.length;
    while (index < n && !(b[index].getName()).equals(name))
    {
        index = index + 1;
    }
    if (index >= n)
        return -1;
    else
        return index;
}
```

(Can also do a for loop)

```
public int linearSearch(BankAccount[] b, String name)
{
    int n = b.length;
    for (int index = 0; index < n; index++)
    {
        if (((b[index].getName()).equals(name))
            return index;
        }
    return -1;
}
```
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:**

```
replace double[] a by String[] a
replace a[j] > a[j+1] by a[j].compareTo(a[j+1]) > 0
replace double temp by String temp
```

(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:**

```
replace double[] a by BankAccount[] a
replace a[j] > a[j+1] by (a[j].getName()).compareTo(a[j+1].getName()) > 0
replace double temp by BankAccount temp
```

(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:**

```
replace double[] a by BankAccount[] a
replace a[j] > a[j+1] by a[j].getBalance() < a[j+1].getBalance()
replace double temp by BankAccount temp
```
Brief Documentation for Classes

Some String class method prototypes

```java
// 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

```java
// 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

```java
// 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()