

Family Name:..... Other Names: .....

ID Number: ..... Signature.....

## Model Solutions

### COMP 102: Test

11 September, 2015

#### Instructions

- Time allowed: **45 minutes**
- Answer **all** the questions. There are 45 marks in total.
- Write your answers in the boxes in this test paper and hand in all sheets. You may ask for additional paper if you need it.
- If you think some question is unclear, ask for clarification.
- Brief Java documentation will be supplied with the test.
- This test will contribute 15% of your final grade,  
(But your mark will be boosted up to your exam mark if that is higher.)
- You may use calculators and paper translation dictionaries.
- You may write notes and working on this paper, but make sure your answers are clear.

#### Questions

#### Marks

1. Variables, input and output	[7]	<input type="text"/>
2. Call methods, arguments and parameters	[8]	<input type="text"/>
3. Create objects, call methods	[10]	<input type="text"/>
4. <b>if</b> and <b>while</b>	[10]	<input type="text"/>
5. Files	[10]	<input type="text"/>
	TOTAL:	<input type="text"/>

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.  
Specify the question number for work that you do want marked.

**Question 1. Variables, input and output**

[7 marks]

(a) [3 marks] Consider the following testVariables method.

```
public void testVariables() {  
    int x = 10;  
    double y = 2.5;  
    x = x + 1;  
    double z = x + y + 1;  
    UI.println (z);  
    UI.println (x+y);  
    UI.println (x + " " + y);  
}
```

What will be printed if testVariables is called?

```
14.5  
13.5  
11 2.5
```

(b) [4 marks] Complete the following rectArea method, so it prompts the user to enter the height and the width of a rectangle, calculates and prints out the area of the rectangle to the screen and also returns the area as a double value.

```
public double rectArea() {  
    double h = UI.askDouble("enter height");  
    double w = UI.askDouble("enter width");  
    UI.println ("the area is " + h *w);  
    return h * w;  
}
```

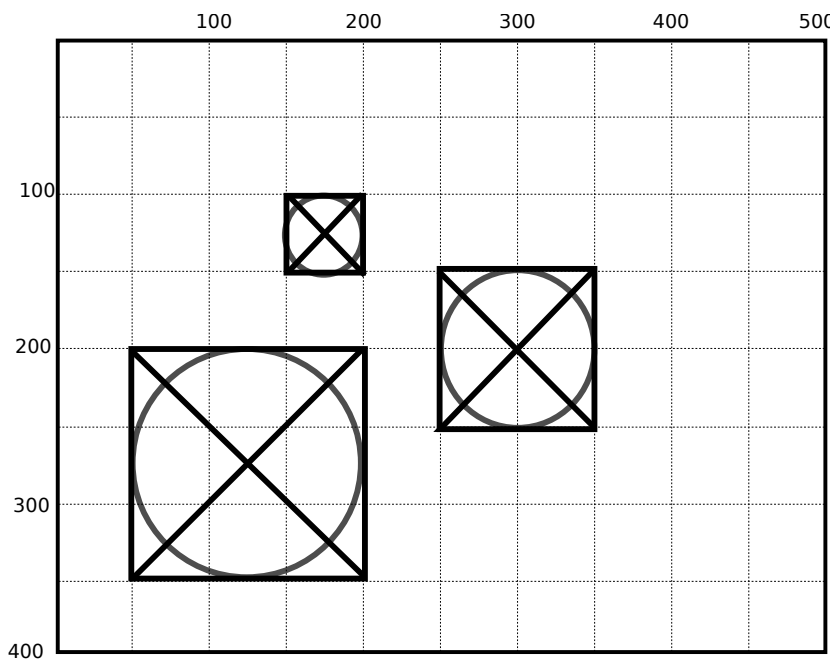
**Question 2. Call methods, arguments and parameters**

[8 marks]

For this question, you are to complete two methods. One method will call the other method three times.

You need to define:

- The drawLable() method: Parameters specify the location x, y and the size. This method draws a picture with a square, a circle and two lines.
- The drawThreeLables() method: calls the drawLable method three times using different arguments, so it draws three different lables at different positions with different sizes as shown in the figure.



```
public class Lables{

    public void drawThreeLables(){

        this.drawLable(150, 100, 50);
        this.drawLable(250, 150, 100);
        this.drawLable(50, 200, 150);

    }

    public void drawLable(                ){

        double x, double y, double size

        UI.drawRect(x, y, size, size);
        UI.drawOval(x, y, size, size);
        UI.drawLine(x, y, x+size, y+size);
        UI.drawLine(x+size, y, x, y+size);

    }

}
```

### Question 3. Create objects, call methods

[10 marks]

Consider the Book class on the facing page

What will the following fragment of code print out?

---

```
Book b1 = new Book("Java Foundations", 130);
Book b2 = new Book("Data Mining", 100);
b1.like ();
b2.like ();
b2.dislike ();
if (b1.recommend()){
    UI.println (b1.toString ());
}
UI.println (b2.toString ());

b2.putOnSale(0.7);
```

---

```
Java Foundations Thumb up

Data mining Thumb up

Data mining Thumb down

Java Foundations ups: 1 downs: 0

Data Mining ups: 1 downs: 1

Data mining special price: 70.0
```

**(Question 3 continued)**The Book class:

---

```
class Book{
    private String description;
    private double price;
    private int likes = 0;
    private int dislikes = 0;

    public Book(String d, double p){
        this.description = d;
        this.price = p;
    }

    public void like(){
        this.likes++;
        UI.println(this.description + " Thumb up");
    }
    public void dislike(){
        this.dislikes++;
        UI.println(this.description + " Thumb down");
    }
    public boolean recommend(){
        return (this.likes > this.dislikes);
    }
    public void putOnSale(double d){
        this.price = this.price * d;
        UI.println(this.description + " special price: " + this.price);
    }
    public String toString(){
        return (this.description + " ups: " + this.likes + " downs: " + this.dislikes);
    }
}
```

---

**Question 4. if and while**

[10 marks]

Complete the `numbers` method on the facing page so that it gets a sequence of integers from the user, counts the number of positive integers and negative integers, and finds the minimum of all positive integers. You may assume the list does not include zeros.

For example, if the user enters the following:

```
10 20 5 -2 10 30 -20 50 -1 done
```

`numbers()` method should print out

```
There are 6 positive integers and 3 negative integers.  
The minimum of the positive numbers is: 5
```



```
public void numbers(){
    UI.println ("please enter some integers, end with done");
    int count1 = 0;
    int count2 = 0;
    int min = Integer.MAX_VALUE;
    while(UI.hasNextInt()){
        int x = UI.nextInt ();
        if ( x > 0){
            count1++;
            if ( x < min) {
                min = x;
            }
        }else {
            count2++;
        }
    }
    UI.println ("There are " + count1 + " positive integers and " + count2 + " nega
    UI.println ("The minimum positive integer is " + min);
}
```

## Question 5. Files

[10 marks]

Suppose a file contains student assignment scores. Each line has the login name of a student (one word only), and a sequence of numbers where each number is a score of one assignment. The students may have completed different number of assignments. An example file would look like this:

```
smithpete 70 82.5 50 60
allendani 35 90 70.5
lewissamu 70 80 9 60 40
chasejohn 40 60.5
```

Complete the `report` method on the facing page which is passed the name of a file. For each student in the file, this method should print the number of assignments the student has completed and the average score of the completed assignments.

For example, given the file above, it would print

```
smithpete 4, 65.6
allendani 3, 65.1
lewissamu 5, 51.8
chasejohn 2, 50.3
```

The `report` method should

- open a `Scanner` to read from the file,
- initialize variables,
- loop through the file, for each line,
  - read the login name,
  - read the scores one by one,
  - print out the number of assignments, the average score for each student

You may assume that the format of the file is correct.

```
public void report(String fileName){
    try{ Scanner scan = new Scanner(new File(fileName));
        //PrintStream out = new PrintStream(new File("updated-" + fileName));
        while ( scan.hasNext() ) {
            String name = scan.next();
            double sum = 0;
            int count = 0;
            while(scan.hasNextDouble()){

                double x = scan.nextDouble();
                sum = sum + x;
                count++;
            }
            UI.println (name + ": " + count + ", " + sum/count);
            //out.println (name + ": " + count + ", " + sum/count);
        }

    } catch(IOException e){UI.println("Fail: " + e);}
}
```

\*\*\*\*\*

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.  
Specify the question number for work that you do want marked.

Student ID: .....

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.  
Specify the question number for work that you do want marked.

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.  
Specify the question number for work that you do want marked.