

Family Name:

Other Names:

ID Number:

Signature

Model Solutions COMP 102: Test 2

19 September, 2013

Instructions

- Time allowed: **45 minutes**
- There are 45 marks in total.
- Answer **all** the questions.
- 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 it is clear where your answers are.

Questions

Marks

1. Understanding Java

[13]

2. Event Driven Input

[10]

3. Defining Classes

[10]

4. File Processing

[12]

TOTAL:

Question 1. Understanding Java

[15 marks]

(a) [7 marks] Consider the following printNumbers method:

```
public void printNumbers(int min, int max){
    int num = max;
    int count = 0;
    while (num > min){
        count++;
        UI.println (count + ": " + num);
        num = num - 1;
    }
    UI.println ("final: " + num);
}
```

What will be printed if printNumbers(10, 16) is called?

```
1: 16
2: 15
3: 14
4: 13
5: 12
6: 11
final: 10
```

(Question 1 continued on next page)

(Question 1 continued)

(b) [6 marks] Consider the following choose method:

```
public void choose(int a, int b, int c){  
    if (a > c)  
        Ul.println ("Monday");  
  
    if (b > c && a > b )  
        Ul.println ("Tuesday");  
    else if (c == b || a == c)  
        Ul.println ("Wednesday");  
    else  
        Ul.println ("Thursday");  
}
```

What will be printed if choose(2, 4, 8) is called?

Thursday

What will be printed if choose(5, 3, 1) is called?

Monday
Tuesday

What will be printed if choose(5, 2, 2) is called?

Monday
Wednesday

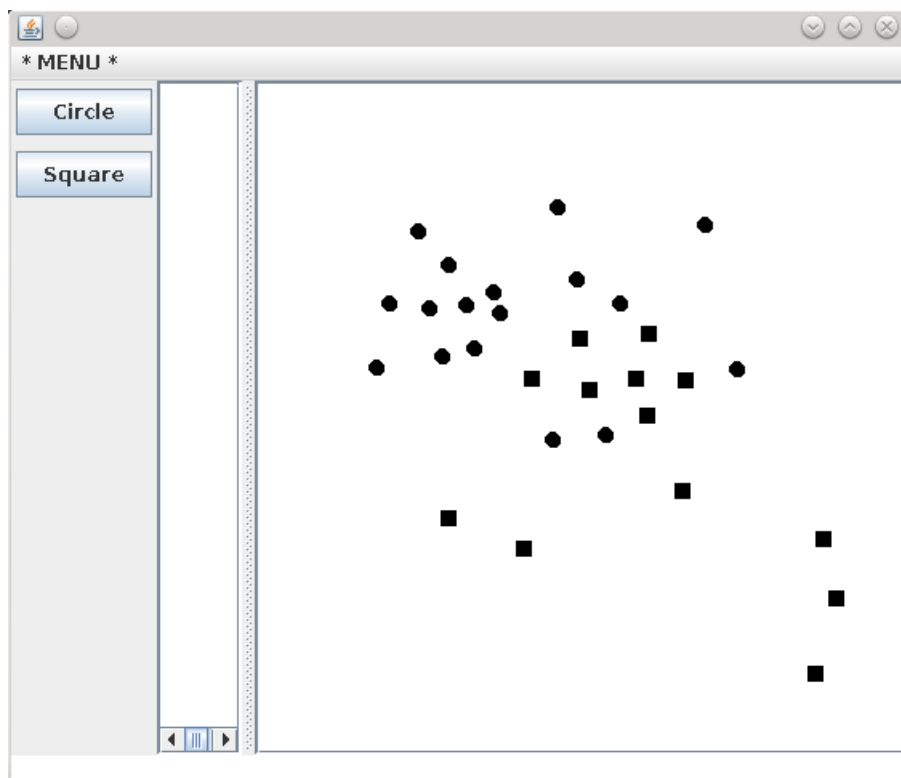
Question 2. Event-Driven Input

[10 marks]

The GuiTester program on the facing page should allow the user to draw small filled circles and squares on the graphics pane, as shown below.

You are to complete the GuiTester program. The program has a field containing the current shape that will be drawn (initially "Circle").

If the user releases the mouse at a point on the graphics pane, then the program should draw the current shape (a circle or a square) whose left top corner is at the point, using a default size of 10 (the diameter or width/height is 10). The two buttons labeled Circle and Square should allow the user to change the current shape.



(Question 2 continued on next page)

(Question 2 continued)

```
public class GuiTester implements UIMouseListener, UIButtonListener
{
    // Fields
    private String shape = "Circle";

    // Set up interface
    public GuiTester(){
        UI.setMouseListener(this);
        UI.addButton("Circle", this);
        UI.addButton("Square", this);
    }

    // Respond to button presses
    public void buttonPerformed(String button){
        this.shape = button
        OR

        if (button.equals("Circle")){
            this.shape = "Circle";
        }
        else if (button.equals("Square")){
            this.shape = "Square";
        }
    }

    public void mousePerformed(String action, double x, double y) {

        if (action.equals("released")) {
            if (this.shape.equals("Circle"))
                UI.fillOval (x, y, 10, 10);
            else
                UI.fillRect (x, y, 10, 10);
        }
    }

    public static void main(String[] arguments){
        new GuiTester();
    }
}
```

Question 3. Defining Classes

[10 marks]

For this question, you are to complete part of a program for recording the tutor hours and their payment.

The payment is \$15 per hour, however, if the tutor is a PhD student, he does not get paid in this tutor system (his hour will be recorded and will be paid as part of his scholarship). So you should have a field to store whether a student is a PhD student and adjust the `printPayment` method accordingly.

Complete the `Tutor` class on the facing page which defines `Tutor` objects. You need to define

- Fields: to store the name, ID, hours and whether the student is a PhD student or not.
- The constructor: initialises the `Tutor` object.
- The `addHours` method: the two parameters are the hours and the type of job the tutor did. If the job is "tutorial", the hours should be doubled and then added. For example, if a tutor did 2 hour tutorials, then 4 hours should be added to his total hour. We pay tutors more for tutorials because they require more preparation time.
- The `printPayment` method: no parameters. If a tutor is a PhD student, only print his name, hours and a thank you message. For other students, print name, hours and the amount to pay.

As an example of using the `Tutor` class, the `testTutor` method below creates two `Tutor` objects, adds some hours and prints payment for them.

```
public static void testTutor(){
    Tutor b1 = new Tutor("Adam", 3335556, true);
    b1.addHours(4, "lab");
    b1.addHours(2, "tutorial");
    Tutor b2 = new Tutor("Bob", 4584874, false);
    b2.addHours(10,"lab");
    b2.addHours(3, "tutorial");
    b1.printPayment();
    b2.printPayment();
}
```

(Question 3 continued)

```
public class Tutor {
    private static final double payrate = 15;

    private String name;
    private int ID = 0;
    private double hours = 0;
    private boolean phd = false;

    public Tutor(String n, int id, boolean b){

        this.name = n;
        this.ID = id;
        this.phd = b;

    }

    public void addHours(double h, String job){

        if (jobs.equals("tutorial"))
            h = h * 2 ;
        this.hours = this.hours + h;

    }

    public void printPayment(){

        UI.println (this.name + " for " + this.hours + " hours");
        if (!this.phd){
            double pay = this.hours * this.payrate;
            UI.println (" \ $" + pay);
        }else
            UI.println ("Thank you, will be paid as part of your scholarship");

    }
}
```

Question 4. File Processing

[12 marks]

Suppose the file `marksheet.txt` is the Assignment 3 marking sheet of a student.

For example, part of the file might be:

```
Marking Sheet for COMP102 2013 T2    Assignment 3
LoginName: Smithpet
Name: Smith, Peter
ID: 300291666
```

```
numQns: 3
Marks: 80 90 100
```

```
Marked by:
-----
Marks guide:
```

```
Comments
```

Complete the `getMarks` method on the facing page so that it reads the “`marksheet.txt`” file and prints out the `LoginName` of the student and the mark of this assignment which is a combination of two marks: the average marks of the program questions (first $n-1$ questions) with a weight of 90%, and the mark of the reflection question (the last question) with a weight of 10%. For example, the mark for the above example is $(80+90)/2 * 0.9 + 100 * 0.1 = 86.5$.

If the file contained just the data above, then calling

```
    getMarks();
```

should print out

```
Smithpet
86.5
```

The file is automatically generated from a template, so the keywords such as `LoginName:`, `ID:`, `numQns:`, `Marks:` are predefined and do not change. The order of these keywords and the format of the file do not change, but there might be blank lines and students with very long names. If a question is not done, a zero mark is recorded for that question.

Your method should work for assignments with any number of questions.

(Question 4 continued on next page)

(Question 4 continued)

```
public void getMarks(){
    int n = 1;
    double mark = 0;
    try{
        Scanner scan = new Scanner(new File("marksheet.txt"));
        while (scan.hasNext()){
            String word = scan.next();
            if (word.equals("LoginName: ")){
                UI.println (scan.next ());
            }
            else if (word.equals("numQns: ")){
                n = scan.nextInt ();
            }
            else if (word.equals("Marks: ")){
                int i = 1;
                while(i < n){
                    mark = mark + scan.nextDouble();
                    i++;
                }
                mark = mark/(n-1) *0.9 + scan.nextDouble()*0.1;
                UI.println (mark);
            }
        }
    }
}

catch(IOException e){UI.println("File Failure: " + 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.