

Family Name:

Other Names:

ID Number:

Signature

COMP 102: Test 2

15 September, 2011

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 paper translation dictionaries, and calculators without a full set of alphabet keys.
- You may write notes and working on this paper, but make sure it is clear where your answers are.

Questions

Marks

1. Basic Java

[12]

2. File Processing

[10]

3. Event Driven Input

[10]

4. Defining Classes

[13]

TOTAL:

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. Basic Java

[12 marks]

(a) [4 marks] Consider the following numbers method

```
public void numbers(){  
    int num = 1;  
    int ans = 0;  
    while (num < 4){  
        Ul.println (num);  
        Ul.println (ans);  
        ans = ans + num;  
        num = num + 1;  
    }  
    Ul.println ("finally: " + ans);  
}
```

What will be printed if numbers() is called?

(Question 1 continued on next page)

(Question 1 continued)

(b) [3 marks] Consider the following choose method

```
public void choose(int a, int b, int c){  
    if (a > b){  
        Ul.println ("One");  
    }  
    else if (a < b && b < c) {  
        Ul.println ("Two");  
    }  
    else if (a == c || b == c) {  
        Ul.println ("Three");  
    }  
    else {  
        Ul.println ("Four");  
    }  
}
```

What will be printed if choose(3, 7, 7) is called?

What will be printed if choose(2, 3, 4) is called?

What will be printed if choose(4, 4, 6) is called?

(Question 1 continued on next page)

(Question 1 continued)

(c) [5 marks] Consider the AAA class shown below.

What will the following fragment of code print out?

```
AAA f1 = new AAA("hello");
AAA f2 = new AAA("Tom");
f1.report();
f1.method1("Bob");
f1.report();
f1.method2(10);
f1.report();

f2.method2(8);
f2.report();

f1.report();
```

```
public class AAA{

    private String x;
    private int count;

    public AAA(String nm){
        this.x = nm;
        this.count = 0;
    }

    public void report(){
        Ul.println (this.x + " : " + this.count);
    }

    public void method1(String str){
        this.x = str;
        this.count = this.count + 1;
    }

    public void method2(int n){
        this.count = this.count + n;
    }
}
```

Question 2. File Processing

[10 marks]

Suppose the file called `NumbersAndNames.txt` contains the following text:

```
-5 Alex
2 Bob
-1 Sam
3 Tom
```

(a) [4 marks] What will the following `filePrint` method print out?

```
public void filePrint (){
    try{
        Scanner scan = new Scanner(new File("NumbersAndNames.txt"));
        while (scan.hasNext()){
            int n = scan.nextInt ();
            String w = scan.next();
            UI.print (w);
            if (n > 0){
                UI.print (n);
            }
            UI.println ();
        }
        scan.close();
    }
    catch(IOException e){UI.println("Fail: " + e);}
}
```

(b) [6 marks] Complete the `chooseNames` method on the facing page. `chooseNames` has one parameter — a threshold for the numbers. The method should read the `"NumbersAndNames.txt"` file, print out the names whose number is over or equal to the threshold and save these names to a new file called `"ChosenNames.txt"`.

For example, if the file contained just the data above, then calling

```
chooseNames(-2);
```

should print out the following names and save them to `"ChosenNames.txt"`.

```
Bob
Sam
Tom
```

Note: the method should **not** ask the user for the threshold.

(Question 2 continued on next page)

(Question 2 continued)

```
public void chooseNames(int t){
    try{
        Scanner scan = new Scanner(new File("NumbersAndNames.txt"));
        PrintStream out = new PrintStream(new File("ChosenNames.txt"));

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

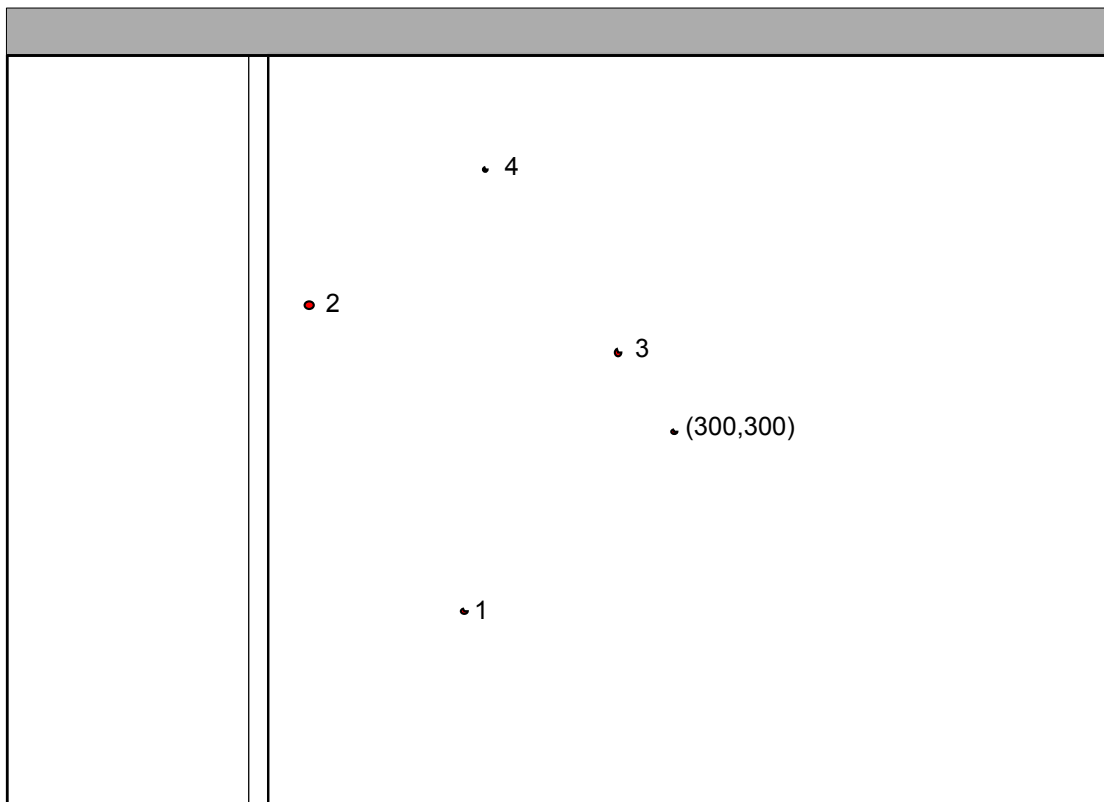
Question 3. Event-Driven Input

[10 marks]

Consider the `PatternWriter` class on the facing page.

(a) [4 marks] Sketch below what the program would draw on the graphics pane if the user took the following mouse actions in sequence:

1. press and release mouse at point 1
2. press and release mouse at point 2
3. press at point 3 and release mouse at point 4



(b) [6 marks] Extend the program to add a `Clear` button. When the `Clear` button is pressed, it should clear the graphics window and reset all the data to get back to the initial state.

(Answer by modifying/extending the code on the facing page)

(Question 3 continued on next page)

(Question 3 continued)

```
import comp102.*;

public class PatternWriter implements UIMouseListener{

    // Fields

    private double xPos = 300;
    private double yPos = 300;

    public PatternWriter(){
        UI.setMouseListener(this);
    }

    public void mousePerformed(String action, double x, double y) {
        if (action.equals("released")) {
            UI.drawLine(this.xPos, this.yPos, x, y);
            this.xPos = x;
            this.yPos = y;
        }
    }

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

Question 4. Defining Classes

[13 marks]

This question is the same as the Bouncer question in Assignment 6. Your task is to complete the BouncingBall class on the facing page.

- Complete the fields: The state of the ball consists of its position, velocity, size and color.
 - The code on the facing page represents the position using two fields: xPos to store the ball's horizontal position and height to store the ball's vertical position above the ground (You may change them if you really want to). The vertical position of the ground is specified by a field ground.
 - You can represent the velocity by two more fields that store the horizontal speed - the distance it will move to the right on each step - and the vertical speed - the distance it will move up (or down if the speed is negative) on each step.
 - A BouncingBall will also need additional information stored in fields - its size, and its colour.
- Complete the constructor: Each new BouncingBall will have a different starting position and a different horizontal speed. Therefore, the constructor for BouncingBall must have several parameters for initial horizontal position, the initial height and horizontal speed. You need to define the constructor to have these parameters (done for you), and to store the values in the appropriate fields. The initial vertical speed is always 0, and the size should be 30. The position and speed fields should be of type double. The colour of the ball could be the same for every ball, or it could be a new random colour for each ball; if so, the constructor must create the colour and store it in a field.
- Complete the three methods:
 - The draw() method should draw the ball at its current position (as stored in the fields). Note that you will need to convert from the height of the ball above the ground into a distance down from the top of the screen.
 - The getX() method should return the horizontal position of the ball (which is stored in a field of the object).
 - The move() method should make the ball move one step; it should
 - * change the position, by adding the value in the speed fields to the position fields. If this would take it past the ground (ie, the height has gone below 0), then the ball has to bounce: the height should be set to 0 and the vertical speed should be inverted so that the ball will move upwards in the next step.
 - * update its speed. The horizontal speed doesn't change. To implement gravity, the vertical speed must be changed every step: if the ball is falling, the magnitude of the vertical speed will increase by a fixed amount (eg 0.3), and if the ball is rising, the magnitude of the vertical speed should decrease by the same fixed amount.

As an example of using the BouncingBall class, the following method below creates a ball, draws it, moves it, draws it again and reports its x position.

```
public void testBouncingBall(){
    BouncingBall ball = new BouncingBall(100, 150, 2);
    ball.draw();
    ball.move();
    ball.draw();
    UI.drawString("X position = "+ ball.getX(), 10, 20);
}
```

(Question 4 continued on next page)

(Question 4 continued)

```
public class BouncingBall{
    private double xPos;    // x position of the ball
    private double height; // height of the ball above the ground
    private double ground = 450; // y position of the ground

    // Constructor
    public BouncingBall(double x, double h, double s){
        this.xPos = x;
    }

    public void draw(){

    }

    public double getX(){

    }

    public void move(){

    }
}
```

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.