

Family Name: Other Names:

Student ID: Signature

COMP 102: Test

2017, May 15 ** WITH SOLUTIONS **

Instructions

- Time allowed: **50 minutes**
- Attempt **all** the questions. There are 50 marks in total.
- Write your answers in this test paper and hand in all sheets.
- If you think some question is unclear, ask for clarification.
- Brief Java documentation is provided with the test
- This test contributes 15% of your final grade
(But your mark will be increased to your exam mark if that is higher.)
- You may use dictionaries.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions

Marks

| | | |
|------------------------|--------|----------------------|
| 1. Understanding while | [5] | <input type="text"/> |
| 2. Writing with while | [8] | <input type="text"/> |
| 3. Defining Classes | [10] | <input type="text"/> |
| 4. Files | [13] | <input type="text"/> |
| 4. Event Driven Input | [8] | <input type="text"/> |
| 5. More Files | [6] | <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. Understanding while**[5 marks]**

Consider the following printOut method.

```
public void printOut(int limit) {  
    int x = 2;  
    while (x < limit) {  
        Ul.println(x);  
        x = x * 2;  
    }  
    Ul.println("Done");  
}
```

limit: x: **(a)** What will be printed if printOut(20) is called?

Hint: Show your working using the boxes for limit and x above.

```
2  
4  
8  
16  
Done
```

(b) What will be printed if printOut(4) is called?

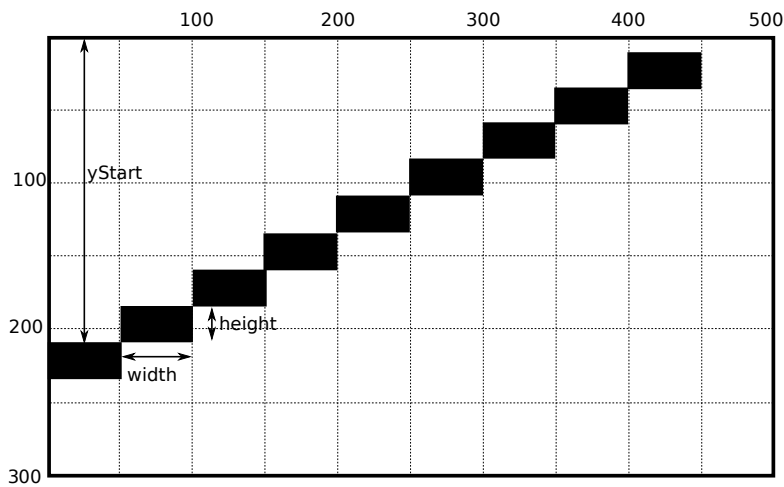
```
2  
Done
```

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 2. Writing with while**[8 marks]**

Complete the following drawStaircase method so that it uses a loop to draw a series of steps with equal spacing, as shown below.



- The parameters specify the horizontal size of each step (width), the height of each step (height), and the y position of the left-most step ($yStart$).
- The steps should continue as far as possible, but should not go over the top of the window.

For example, the pattern above was generated by calling drawStaircase(50, 25, 210).

```

public void drawStaircase(double width, double height, double yStart) {
    double x = 0.0;
    double y = yStart;
    while (y >= 0.0) { // or (y > 0.0) if the edge counts as 'over'
        UI.fillRect(x, y, width, height);
        x = x + width;
        y = y - height;
        UI.println(y);
    }
}

```

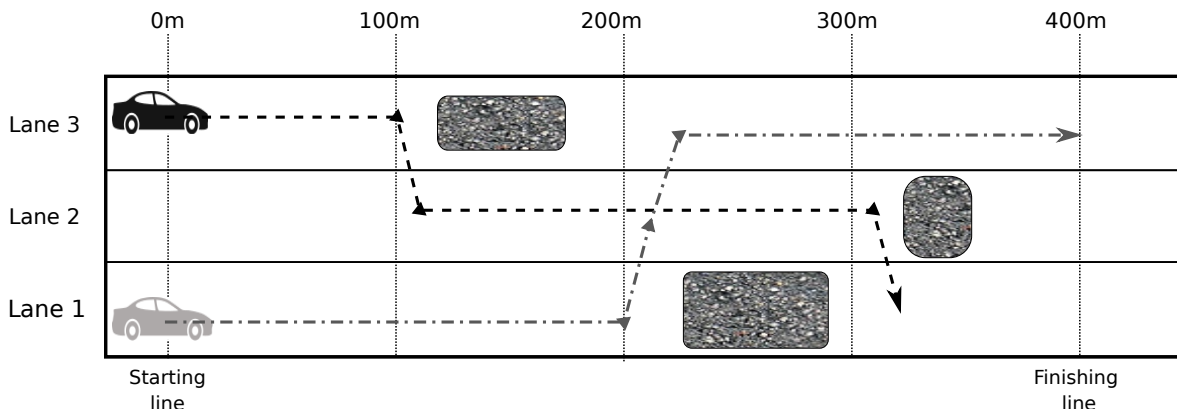
```

}

```

Question 3. Defining Classes**[10 marks]**

In the first Test, you were asked about a program that makes two cars move along a racetrack with three lanes:



For this question, you are to complete the Car class on the facing page.

Define:

- Fields to store the relevant information about Cars
- The constructor to initialise the fields for a Car.
For example, the gray car in the picture was produced by calling `new Car(Color.gray, 1)`
- The forward, changeUp, and changeDown methods, as specified in their comments.

Note: The constructor and the methods do not need to redraw the Car.
You do not need to define the draw method.

```

public class Car{
    public static final int LANES = 3;
    public int lane;
    public double distance;
    public Color color;

    /** Creates a new Car object of the specified colour, placing it
        at the starting line in the specified lane. */
    public Car(Color color, int lane){
        this.lane = lane;
        this.distance = 0.0;
        this.color = color;

    }

    /** Moves the Car forward in its current lane by the specified distance. */
    public void forward(double dist){
        this.distance = this.distance + dist;

    }

    /** Moves the Car up a lane and forward by 10, unless already in the top lane */
    public void changeUp(){
        if (this.lane < LANES) {
            this.lane = this.lane + 1; // or this.lane++;
            this.distance = this.distance + 10; // or this.forward(10);
        }

    }

    /** Moves the Car down a lane and forward by 10, unless already in the bottom lane */
    public void changeDown(){
        if (this.lane > 1) {
            this.lane = this.lane - 1; // or this.lane--;
            this.distance = this.distance + 10; // or this.forward(10);
        }

    }

    public void draw(){
        // YOU DO NOT NEED TO COMPLETE THIS METHOD
    }
}

```

Question 4. Files**[13 marks]**

Every month, a company runs a brief anonymous survey of its customers, and the answers are saved in a file, one line with the three answers for each customer.

- The first question is “Do you want to buy this item?”
- The second question is “How many would you order if you did buy it?”
- The last question is “What material would you prefer?”

Here are the first few lines of their June data file.

```
Yes 15 Wool
Yes 25 Cotton
No 12 Wool
Yes 3 Wool
Yes 10 Wool
```

Consider the following printData method.

```
public void printData(){
    try{
        Scanner scan = new Scanner (new File("JuneData.txt"));
        while ( ! scan.hasNextInt() ){
            String a = scan.next();
            String b = scan.next();
            Ul.println (b + "-" + a);
        }
        int num = scan.nextInt();
        Ul.println ("done: " + num);
    } catch(IOException e){Ul.println ("Fail: " + e);}
}
```

(a) **[5 marks]** What will be printed if printData is called, assuming “JuneData.txt” is the example file shown above?

Hint: Keep track of where the Scanner is up to.

```
15-Yes
Yes-Wool
done: 25
```


(b) [8 marks] Complete the following countGoodCustomers method which is passed the name of a survey data file and the name of a material. The countGoodCustomers method should read through the file and count the number of customers who want to buy at least 5 items in the specified material.

For example, using the example file above,

countGoodCustomers("JuneData.txt", "Wool") should print

2 customers.

because only the first and last of the customers want to buy at least 5 of the items in the Wool material.

```
public void countGoodCustomers(String fileName, String material){
    try{
        Scanner scan = new Scanner (new File(fileName));
        int goodCount = 0;
        while (scan.hasNext()){
            String want = scan.next();
            int number = scan.nextInt ();
            String prefMaterial = scan.next ();
            if (want.equals("Yes") && number >= 5 && prefMaterial.equals(material)){
                goodCount++;
            }
        }
        UI.println (goodCount + " customers");
    } catch(IOException e){UI.println ("Fail: " + e);}
}
```

Question 5. Event-Driven Input**[8 marks]**

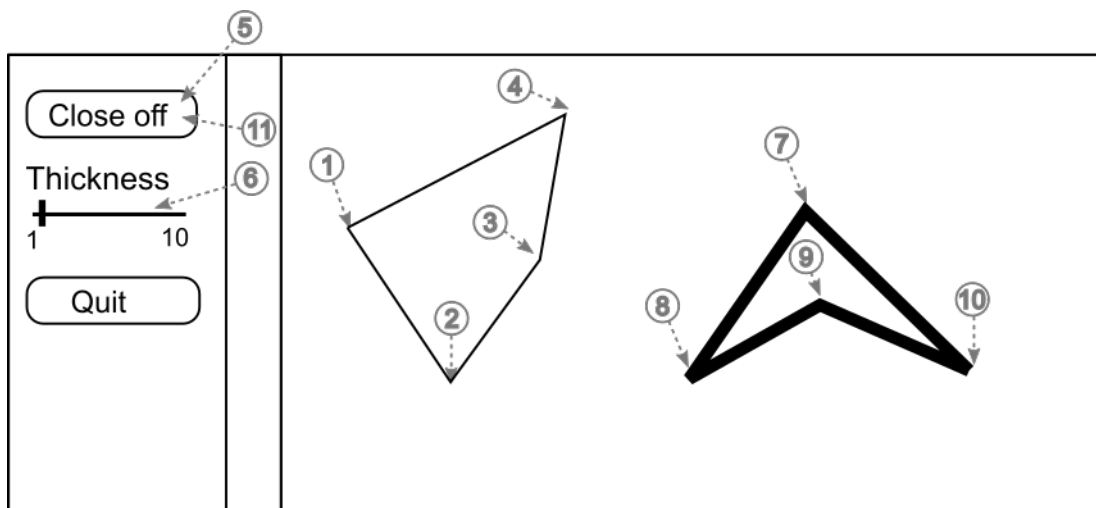
Complete the PolygonDrawer program on the facing page so that it allows the user to draw polygons on the graphics pane.

The program should have two buttons and a slider:

- "Close off" button, which should allow the user to close off the current polygon (joining the last point to the first point) and get ready to start a new polygon
- "Thickness" slider (range 1 to 10), which should set the thickness of the lines.
- "Quit" button, which should quit the program.

For example, the diagram shows what the program might draw if the user carried out the actions below. The circled numbers indicate where the user clicked.

- clicked the mouse at positions ① to ④, then
- clicked the Close off button ⑤, then
- moved the slider to 8, ⑥ then
- clicked the mouse at positions ⑦ to ⑩, then
- clicked the Close off button again ⑪.

**Hints:**

- When the user clicks on the first point of a polygon, the program should not draw anything.
- The program needs to remember the first point of the current polygon (to be able to close it off) and also the latest point.
- See the documentation on page 2.

(Question 5 continued on next page)

(Question 5 continued)

```

public class PolygonDrawer {
    private double startX, startY;
    private double prevX, prevY;
    private boolean drawingNow = false;

    public PolygonDrawer(){
        UI.addMouseListener(this :: doMouse);
        UI.addButton("Close Off", this::doClose);
        UI.addSlider("Thickness", 1, 10, 1, this :: doThickness);
        UI.addButton("Quit", UI::quit);
    }
    public void doClose(){
        if (this.drawingNow){
            UI.drawLine(this.prevX, this.prevY, this.startX, this.startY);
            this.drawingNow = false;
        }
    }
    public void doThickness(double v){
        UI.setLineWidth(v);
    }
    public void doMouse(String action, double x, double y){
        if (action.equals("released")){ \\ "pressed" would work too.
            if (this.drawingNow){
                UI.drawLine(this.prevX, this.prevY, x, y);
            }
            else {
                this.startX = x;
                this.startY = y;
                this.drawingNow=true;
            }
            this.prevX = x;
            this.prevY = y;
        }
    }
}

```

Question 6. More Files**[6 marks]**

A science researcher has performed a series of experiments on some samples that generate a sequence of readings for each sample. Each sample may have a different number of readings. The results have been stored in a file in the following format.

Each sample has two lines of data:

- A line with the sample ID and the number of readings for that sample.
- A line with a sequence of numbers (the readings)

For example, the file might contain:

```
ER3001 5
4.5 2.3 8.3 9.8 2.0
FD2778 3
7.0 4.5 2.3
MM7631 10
8.3 9.8 2.0 12.0 8.3 9.8 2.0 12.0 8.3 9.8
```

Complete the reportData method on the facing page. This should read the data from the file, printing out the sample ID and the average of the readings for each sample.

For example, with the file above, reportData should print out

```
ER3001 5.38
FD2778 4.6
MM7631 8.23
```

If a sample has 0 readings, reportData should print a “-” in place of the average.

(Question 6 continued)

```
public void reportData(String filename){
    try{
        Scanner scan = new Scanner (new File(filename));
        while (scan.hasNext()){
            String id = scan.next();
            int count = scan.nextInt ();
            if (count==0){
                Ul.println (id +" - ");
            }
            else {
                double tot = 0;
                int i = 0;
                while (i < count){
                    tot = tot + scan.nextDouble();
                    i++;
                }
                Ul.println (id +" " + tot/count);
            }
        }
    }
} catch(IOException e){Ul.println ("Fail: " + e);}
}
```
