

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

Student ID: ..... Signature .....

## COMP 102 : Midterm Test

2023, October 25 \*\* WITH SOLUTIONS \*\*

### Instructions

- Time allowed: **60 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 a question is unclear, ask for clarification.
- Brief Java documentation is provided with the test.
- This test contributes towards 20% of your final grade.  
(But your mark will be increased to your exam mark if that is higher.)
- You may use dictionaries and calculators.
- You may write notes and working on this paper, but make sure your answers are clear.
- You may assume all the programs import the ecs100 library and other standard libraries.

### Questions

### Marks

1. User input	[10]	<input type="text"/>
2. Writing programs with <u>if</u>	[10]	<input type="text"/>
3. Defining methods with parameters	[10]	<input type="text"/>
4. Writing methods that use objects	[10]	<input type="text"/>
5. Writing programs with <u>for</u>	[10]	<input type="text"/>
	TOTAL:	<input type="text"/>

**Question 1. User Input****[10 marks]**

Complete the calculateCircleArea() method below so that it:

- Asks the user to input a double number: a *radius*.
- Prints out the area of circle.  
You should use the constant PI for the calculation

For example:

```
Enter radius: 5
The area of the circle is 78.54
```

**Hint:** The area of a circle is  $PI \times radius \times radius$ .

```
public static final double PI = 3.14;

public void calculateCircleArea () {

    double radius = UI.askDouble("Enter radius: ");

    double area = PI * radius * radius;

    UI.println("The area of the circle is " + area);
//OR
    UI.printf("The area of the circle is %.1f\n", area);

}
```

**Question 2. Writing programs with if****[10 marks]**

Your task is to complete the `getMessage(...)` method to **return** a String message for the number specified in the parameter, according to the following table:

<u>Number</u>	<u>Message</u>
from 10 up to 20	good
smaller than 5	too small
larger than 50	too big
other numbers	not sure

For example, if `getMessage(...)` is called with the below arguments, it returns:

<u>argument</u>	<u>returns</u>
40	not sure
60	too big
12	good
2	too small
90	too big
10	good

```
/** Returns the message associated with the given number */  
public String getMessage(double number){
```

```
    if ((number>=10) && (number<=20)) {  
        return "good";  
    }  
    else if (number>50){  
        return "too big";  
    }  
    else if (number<5){  
        return "too small";  
    }  
    else{  
        return "not sure";  
    }  
}
```

```
}
```

**Question 3. Defining methods with parameters**

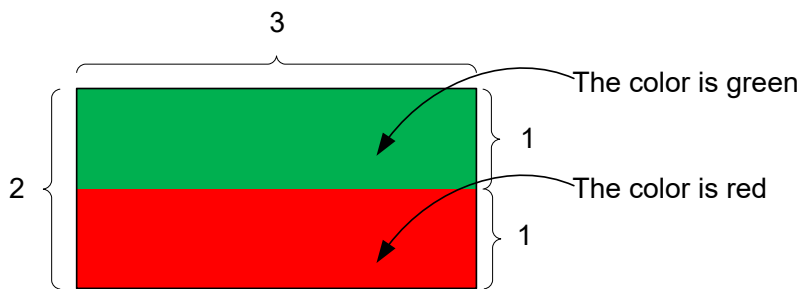
**[10 marks]**

The program on the facing page has two methods. The testDrawFancyRect() method calls the drawFancyRect(...) method three times to draw three different fancy rectangles.

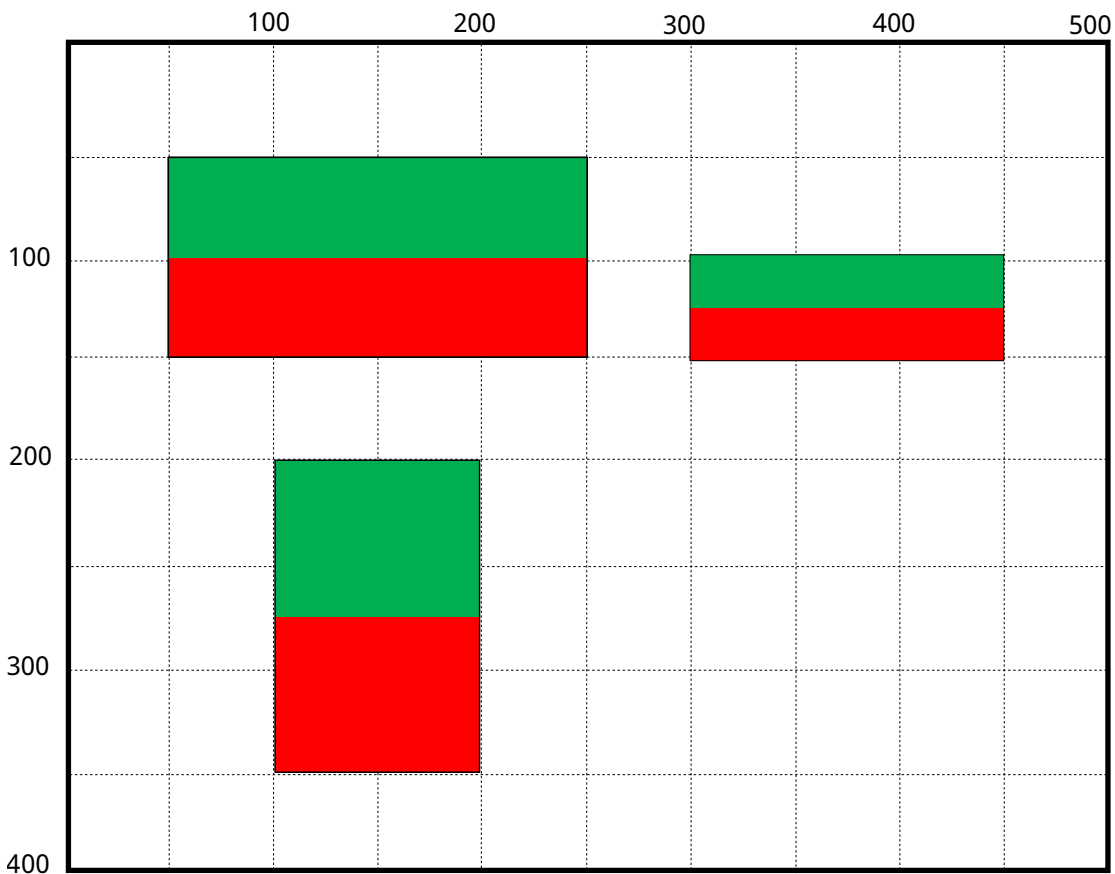
(a) [7 marks] Complete the drawFancyRect(...) method to draw a green-red rectangle with a black rectangular border.

You will need to define four parameters for drawFancyRect(...):

- the x, y position of the fancy rectangle (i.e. top-left of the fancy rectangle).
- the height of the fancy rectangle.
- the ratio of the fancy rectangle (width/height of the fancy rectangle, e.g. 3.0/2.0 or 1.5 in the example below).



(b) [3 marks] Complete the testDrawFancyRect() method to draw three rectangles like the ones in the example picture below. (The ratio of the rectangles does not need to be precisely the same as in the picture, e.g. there is no need to round results from division or multiplication).



```
public class Drawer {  
  
    // Draws one fancy rectangle at given position , height , and ratio .  
    public void drawFancyRect( double x, double y, double height, double ratio ) {  
  
        double width = height * ratio ;  
        double barHeight = height / 2.0;  
        Ul.setColor( Color.green );  
        Ul.fillRect (x, y, width, barHeight);  
        Ul.setColor( Color.red );  
        Ul.fillRect (x, y+barHeight, width, barHeight);  
  
        Ul.setColor( Color.black );  
        Ul.drawRect(x, y, width, height );  
  
    }  
    // Draws three fancy rectangles .  
    public void testDrawFancyRect() {  
        Ul.clearGraphics ();  
        this.drawFancyRect( 50, 50, 100, 4.0/2.0 );  
  
        this.drawFancyRect( 100, 200, 150, 2.0/3.0 );  
  
        this.drawFancyRect( 300, 100, 50, 3.0/1.0 );  
  
    }  
}
```

**Question 4. Writing methods that use objects****[10 marks]**

This question is about a program that creates and animates two cartoon Cats in a  $6 \times 6$  maze. The documentation for the Cat class is given below.

---

```

class Cat {
    // Constructor:
    public Cat(String name, Color col)
    /** Creates a new cat object with the specified name. e.g. "MiMi", and
        the specified colour, e.g. Color.red
        The cat is drawn at the start position, facing the maze entrance */

    //Methods:
    public void turn(String dir)
    /** The parameter can only be either "left" or "right"
        This method makes the cat face a new direction and draws the cat again */

    public void move(int step)
    /** Moves the cat forward for the number of squares (as long as there is space
        in the maze) and draws the cat again */
}

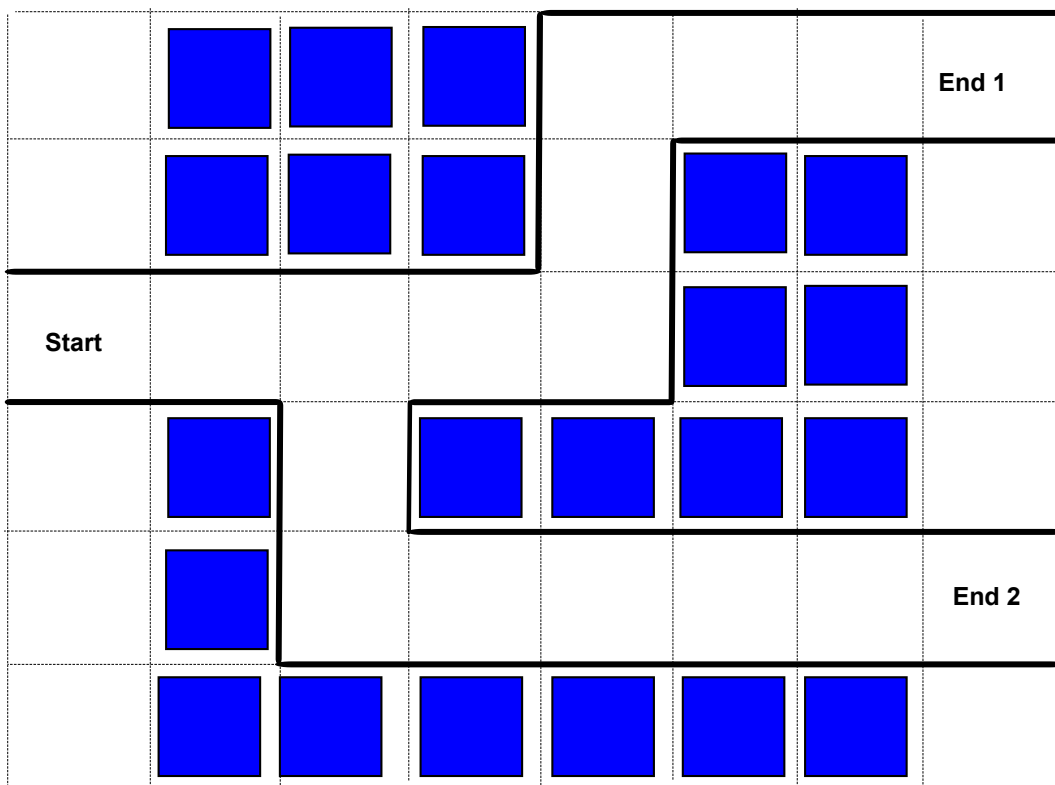
```

---

Complete the animate method in the Cat class on the facing page so that it

1. Creates a brown cat named "MiMi"
2. Creates a blue cat named "Tom"
3. Calls a sequence of five methods on each cat, which will make the first cat (MiMi) to get to the End 1, and will make the second cat (Tom) get to the End 2.

You **should** use methods in the Cat class to do all the steps above, i.e. there is no need to use loops or conditionals to achieve the actions.



```
public class Cat{
    public void animate(){

        Cat c1 = new Cat("MiMi", Color.brown);
        c1.move(4);
        c1.turn("left");
        c1.move("2");
        c1.turn("right");
        c1.move("3");

        Cat t2 = new Cat("Tom", Color.blue);
        c2.move(2);
        c2.turn("right");
        c2.move(2);
        c2.turn("left");
        c2.move(5);

    }
}
```



**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 5. Writing methods with for****[10 marks]**

The program for this question has three methods:

- `analyseNumbers()`, which asks the user for a sequence of numbers and saves the numbers in an `ArrayList`. This method calls the other two methods.
- `findMaximum(...)`
- `plotNumbers(...)`

(a) **[5 marks]** Complete the `findMaximum` method to find and return the maximum of the numbers the given `ArrayList`. Please note that the maximum is printed out by the `analyseNumbers` method, not this method.

```
public void analyseNumbers () {
    ArrayList<Double> nums = UI.askNumbers("Enter numbers [0, 350]");
    UI.println ("The maximum is " + this.findMaximum(nums));
    this.plotNumbers(nums);
}

public double findMaximum(ArrayList<Double> list) {

    double maximum = Double.NEGATIVE_INFINITY;

    for (double num: list) {
        if (maximum<num) maximum = num;
    }

    return maximum;

}
```

**(Question 5 continued)**

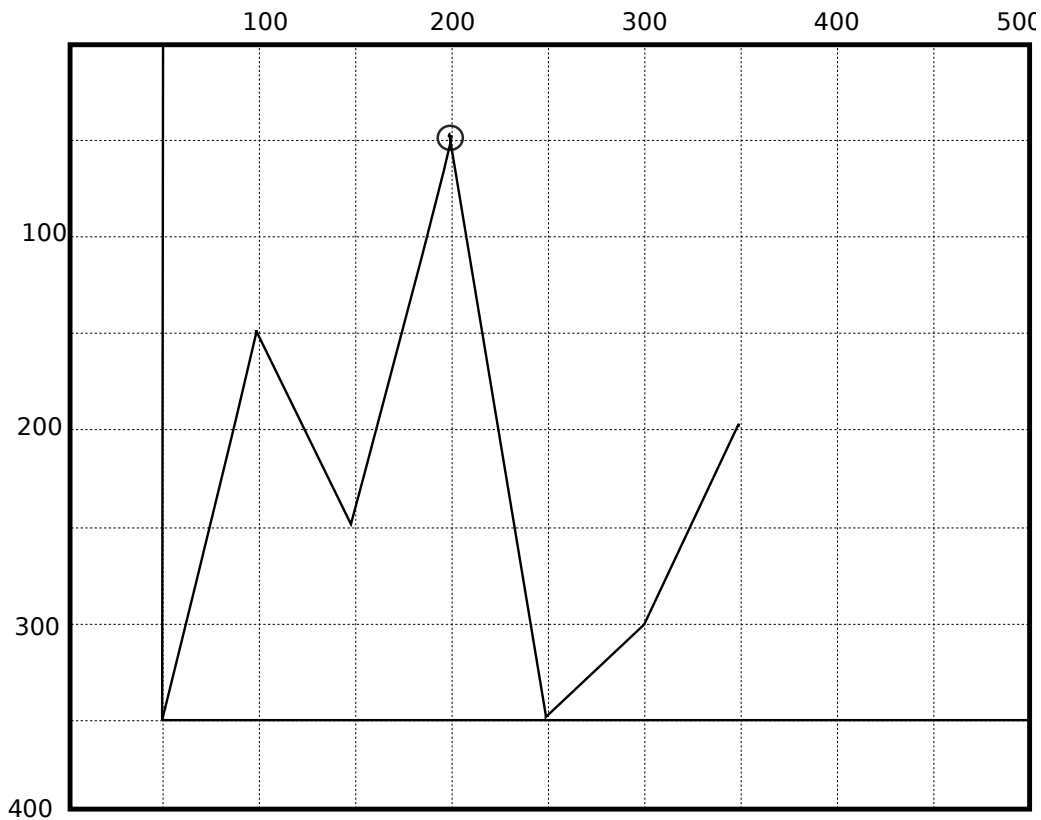
(b) [5 marks] Complete the plotNumbers method to draw a line plot of the numbers on the graphics pane with a small circle on the maximum value(s)

- Draw the x-axis and the y-axis. The origin should be at (50, 350)
- Draw the points every 50 units. You should start the line plot from the origin. You may assume the user enters at least one and at most nine numbers and all numbers are in the range of [0, 350].
- Draw a small circle of diameter 6 for the maximum number(s).

For example, if the user enters the following numbers:

```
200
100
300
0
50
150
done
```

The line plot should look like this:



**(Question 5 continued)**

```
public void plotNumbers(ArrayList<Double> numbers){

    UI.drawLine(50, 350, 500, 350);
    UI.drawLine(50, 350, 50, 0);
    double prevX=50;
    double prevY=350;

    for (double n : numbers) {
        UI.drawLine(prevX, prevY, prevX+50, 350-n);
        prevX=prevX+50;
        prevY=350-n;
        if (n == this.findMaximum(numbers)){
            UI.drawOval(prevX-3, prevY-3, 6, 6);
        }
    }

    // alternatively , the oval can be done separately
    double x=50;
    for (double n: numbers){
        x= x+50;
        if (n == this.findMaximum){
            UI.drawOval(x-3, 350-n-3, 6, 6);
        }
    }

}
```

\*\*\*\*\*

**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.