COMP 102 : Test

2022, 25 November

Instructions

- Time allowed: **50 minutes**

- Attempt **all** the questions. There are 45 marks in total.

- **In-person**: Write your answers in this test paper and hand in all sheets.
  Remote: Type your answers in `test1.txt` and submit to "Test 1 Remote" on the COMP 102 submission system.

- If you think a question is unclear, ask for clarification.

- Brief Java documentation is provided with the test.

- This test contributes 10% of your final grade.

- 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. Input, calculation and output	[10]	
2. Debugging code	[15]	
3. Programs with choice	[12]	
4. Designing with parameters	[8]	
	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. Input, calculation and output [10 marks]

Complete the (calculateCourier) method below that calculates the cost of a courier package. The cost consists of a pickup charge and a delivery charge.

- The method should first ask the user to input two numbers: a *distance* and a *speed*. The speed will be an integer: 1, 2 or 3.

- The method should then calculate the cost, which is the *pickup-charge* (same for all packages) plus the delivery cost. The delivery cost is the *distance* multiplied by the *speed* multiplied by the *unit-charge*.
  You should use the two named constants for the *pickup-charge* and the *unit-charge*.

- The method should then print out the cost in the form:
  ```
  Cost of package is $35.43
  ```
  with exactly 2 digits after the decimal point.

```java
public static final double PICKUP_CHARGE = 8.50;
public static final double UNIT_CHARGE = 1.85;

public void calculateCourier () {




}
```

Question 2. Debugging code [15 marks]

The following drawMarker(...) method is intended to draw a circle containing an arrowhead point-ing at the center of the circle.

```
public void drawMarker (double y,  double x) {
      UI. fillOval (x, y,  20,  20);
      UI.drawLine(x, y,  x−20, y−10);
      UI.drawLine(x, y,  x+20, y−10);
   }
```

- The circle should have a radius of 20 units.

- The width of the arrowhead should be the same as the radius (20 units);

- The height should be half the radius (10 units).

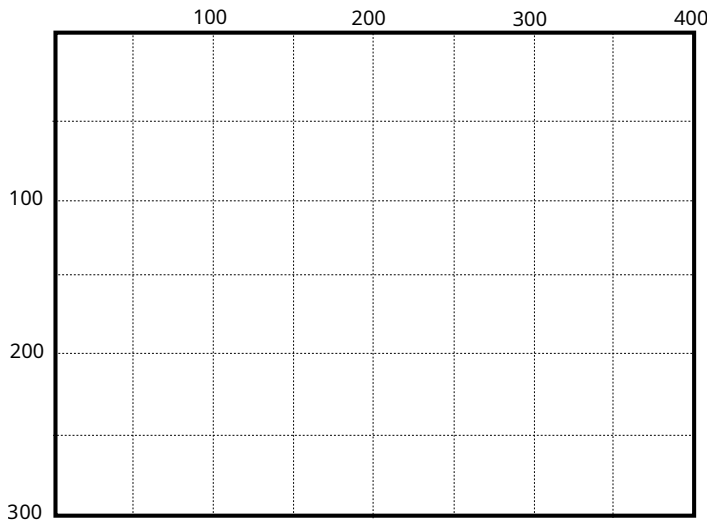- The parameters of drawMarker should be the position of the point of the arrowhead,
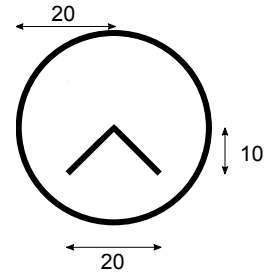
For example:

    drawMarker(200, 100);

should draw a marker like the picture above, centered at the point 200 units over from the left side of the graphics pane and 100 units down from the top of the graphics pane.

(a) **[6 marks]** The drawMarker method above compiles and runs, but it does not do the right thing. Sketch what drawMarker(200, 100); would actually draw on the graphics pane:

(Question 2 continued)

(b) **[9 marks]** Fix the drawMarker method so that it does what it is supposed to do. You are allowed to define a variable or constant for the radius if you wish.

```
public void drawMarker(                              ){



}
```

Question 3. Programs with choice [12 marks]

Consider the following choose(...) method:
Hint: look carefully at the if's and else's.

```java
public void choose(double num, String colour){
    if (num == 15.0 && colour.equals("blue")){
        UI. println ("First choice");
    }
    else if (num < 10.0 || num > 20.0){
        UI. println ("Second choice");
    }
    else {
        UI. println ("Third choice");
    }
    if (num/2 < 7.0 || colour.startsWith("r")){
        UI. println ("Fourth choice");
    }
    else {
        UI. println ("Fifth choice");
    }
    UI. println ("Sixth choice");
}
```

(a) **[2 marks]** What will be printed by the statement choose(25, "red");

(b) **[2 marks]** What will be printed by the statement choose(15, "blue");

(c) **[2 marks]** What will be printed by the statement choose(12, "green");

(d) **[6 marks]** Complete the following drawShape(..) method that should draw a shape of size 20 units (length and width) centered at (x,y).

The shape should be a circle if the shp argument has the value "circle" and a square if shp has the value "square".

If shp has any other value, then it should not draw anything, but should print the message: "Unknown shape".

```java
public void drawShape(double x, double y, String shp){




}
```

Question 4. Designing with parameters [8 marks]

The following drawWords method draws a "word cloud" of some Java keywords. Each word has a coloured rectangle for the background and the word (capitalised) in a darker color in the rectangle.

The following image shows what the output should be (the printed version may be in black-and-white; the words and backgrounds should be coloured):
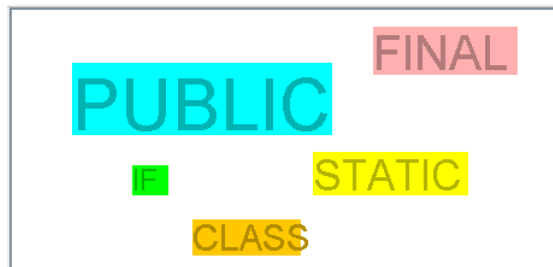
Note:

- the height of a background rectangle depends on the font size.
- the width of a background rectangle depends on the length (number of characters) of the word and the fontsize. On average, the width of a character is about 60% of the font size.

```
public void drawWords(){
    UI.setColor(Color.green);              // green "if"
    UI.fillRect(100, 125, 30, 25);
    UI.setColor(Color.green.darker());
    UI.setFontSize(25);
    UI.drawString("if".toUpperCase(), 100, 150);
    UI.setColor(Color.cyan);               // cyan "public"
    UI.fillRect(50, 40, 216, 60);
    UI.setColor(Color.cyan.darker());
    UI.setFontSize(60);
    UI.drawString("public".toUpperCase(), 50, 100);
    UI.setColor(Color.orange);             // orange "class"
    UI.fillRect(150, 170, 90, 30);
    UI.setColor(Color.orange.darker());
    UI.setFontSize(30);
    UI.drawString("class".toUpperCase(), 150, 200);
    UI.setColor(Color.pink);               // pink "final"
    UI.fillRect(300, 10, 120, 40);
    UI.setColor(Color.pink.darker());
    UI.setFontSize(40);
    UI.drawString("final".toUpperCase(), 300, 50);
    UI.setColor(Color.yellow);             // yellow "static"
    UI.fillRect(250, 114, 129, 36);
    UI.setColor(Color.yellow.darker());
    UI.setFontSize(36);
    UI.drawString("static".toUpperCase(), 250, 150);
}
```

(Question 4 continued)

The given version of drawWords() is very badly designed. Adding an extra word to the "word-cloud", or changing the position of one of the words would be a pain.

Rewrite drawWords() using at least one additional method with parameters.

Your version should have the same output as the original drawWords().

The design should make it much easier to modify the code to add or change a word.

public void drawWords(){

* * * * * * * * * * * * * *

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.