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

ID Number: . . . . **Signature** . . . .

Model Solutions

COMP 102: Test 2

11 May, 2016

Instructions

- Time allowed: **50 minutes**
- Answer **all** the questions. There are 45 marks in total.
- 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 your answers are clear.

Questions	**Marks**	
1. Understanding while	[5]	
2. Writing with while	[8]	
3. Defining Classes	[10]	
4. Event Driven Input	[9]	
5. Files	[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. Understanding while [5 marks]

Consider the following printOut method.

```
public void printOut(int x, int y) {

    while (x < y) {
        UI.println(x);
        x = x + 1;
        y = y − 1;
    }
    UI.println("Done");
}
```

x: □ y: □

What will be printed if printOut(3,10) is called?

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

```
Write your answers here
3
4
5
6
Done
```

What will be printed if printOut(10, 5) is called?

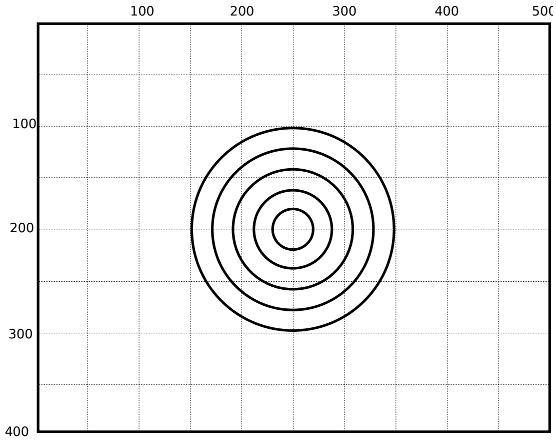
```
Write your answers here
Done
```

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 drawCircles method so that it uses a loop to draw concentric circles with equal spacing, as shown below.



- The parameters specify the location of the center (x, y) and the radius of the biggest circle.
- The radius of each circle is 20 less than the radius of the previous circle.
- The smallest circle must have a radius of at least 20.

For example, drawCircles(250, 200, 100) should draw the pattern above.

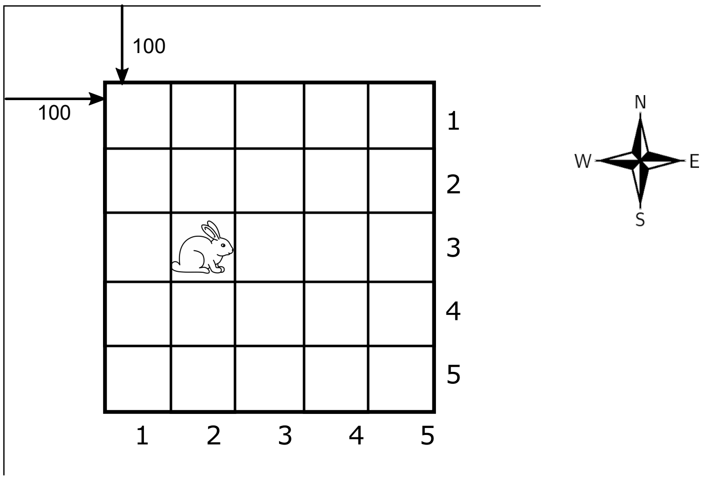
Hint: remember that the diameter of a circle is 2 × the radius.

```
public void drawCircles(double x, double y, double radius) {
    while (radius >= 20) {
        double left = x - radius;
        double top = y - radius;

        UI.drawOval(left, top, radius*2, radius*2);
        radius = radius - 20;
    }
}
```

Question 3. Defining Classes [10 marks]

This question concerns a program for a game in which a rabbit moves around a 5 × 5 board looking for food.



You are to complete the Rabbit class below. Note that the draw method is written for you.

A rabbit knows its current location (the row and column of the board), and its current direction ("North", "South", "East", or "West"). It can turn to a new direction, and it can jump a specified number of squares in its current direction.

Define fields to store the following information:

- location of the Rabbit,
- direction the Rabbit is facing,

Complete

- the constructor which initializes the Rabbit object with the location and the initial direction. Note that the parameters specify the row and the column number.
- the turn(String dir) method, which changes the direction that the Rabbit is facing. The argument can be "North", "South", "East", or "West".
- the jump(int steps) method, which makes the Rabbit change its location by the specified number of steps in the current direction.
  For example, if it is at position (3,2) and facing East (as in the diagram above), jump(2) should move it to position (3, 4).
  jump should never take the rabbit beyond the edge of the board (row or column less than 1 or greater than 5). For example jump(4) from the position (3, 2) should make it end up at (3, 5), not (3,6).

```
class Rabbit {
    // fields
    private int row;
    private int col;
    private String direction;



    public Rabbit (int row, int col, String dir){

        this.row = row;
        this.col = col;
        this.direction = dir;

    }
    public void turn(String newDir){

        this.direction = newDir;

    }
    public void jump(int steps){

        if (this.direction.equals("North")){
            this.row = this.row - steps; if (this.row<1) {this.row = 1;}
        } else if (this.direction.equals("South")){
            this.row = this.row + steps; if (this.row>5) {this.row = 5;}
        } else if (this.direction.equals("West")){
            this.col = this.col - steps; if (this.col<1) {this.col = 1;}
        } else if (this.direction.equals("East")){
            this.col = this.col + steps; if (this.col>5) {this.col = 5;}
        }








    }
    public void draw(){ // Board.SIZE is the size of each square, predefined as 40
        double left = 100+(this.col-1) * Board.SIZE;
        double top = 100+(this.row-1) * Board.SIZE;
        UI.drawImage("rabbit.png", left, top, Board.SIZE, Board.SIZE);
    }
}
```

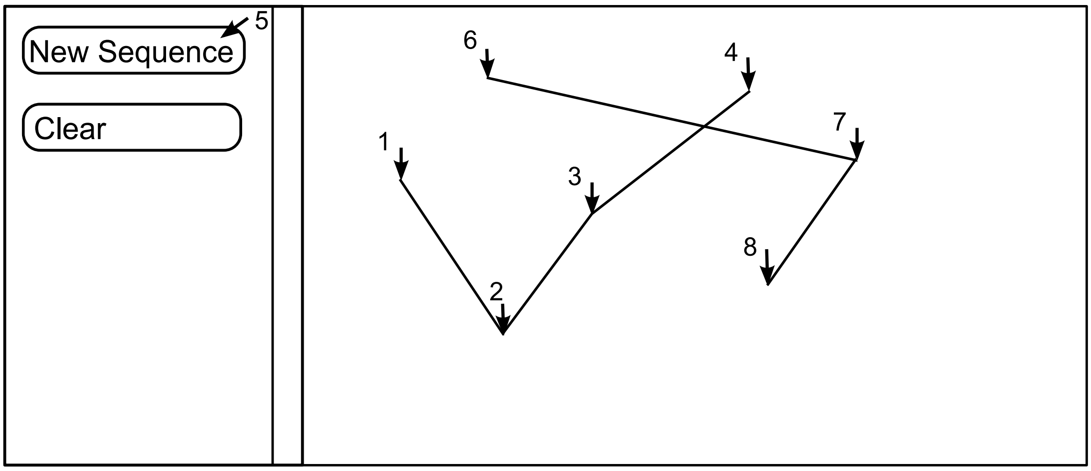
Question 4. Event-Driven Input [9 marks]

Complete the DrawLines program on the facing page so that it allows the user to draw sequences of connected lines on the graphics pane.

The program should have two buttons:

- "New Sequence", which should allow the user to start a new sequence of lines
- "Clear", which should clear the graphics pane.

The diagram shows an example of what the program might draw if the user clicked the mouse at positions 1 to 4, then clicked the New Sequence button, and then clicked the mouse at positions 6 to 8. (The arrows show where the user clicked.)



Hint: When the user clicks in the graphics pane:

- If the program is starting a new sequence, it should simply remember the mouse position.
- If the program is in the middle of a sequence, it should draw a line from the previous mouse position to the new mouse position.

(Question 4 continued on next page)

(Question 4 continued)

```
public class DrawLines {
    // Fields
    private double preX;
    private double preY;
    private boolean start = true;


    // Constructor: set up buttons and mouse
    public DrawLines(){


        UI.setMouseListener(this::doMouse);
        UI.addButton("New Sequence", this::doLine);
        UI.addButton("Clear", UI::clearGraphics);

    }
    // Methods to respond to buttons and mouse

    public void doLine(){
        this.start = true;
    }

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

        if (action.equals("released")){
            if (this.start){
                this.preX = x;
                this.preY = y;
                this.start = false;
            }
            else {
                UI.drawLine(this.preX, this.preY, x, y);
                this.preX = x;
                this.preY = y;
            }
        }
    }

    }




}
```

Question 5. Files [13 marks]

Suppose a file has information about departing flights. Each line has the name of the airline, the flight number, destination, time, the boarding gate number and some notes, for example:

```
Virgin Australia VA113 Palmerston North 16:35 4 boarding
Air New Zealand NZ6204 Auckland 21:55 6 go to gate
Air New Zealand NZ5372 Christchurch 22:45 1 relax
```

Consider the following readFromFile method.

```
public void readFromFile(String fileName){
    try{
        Scanner scan = new Scanner (new File(fileName));
        String line = scan.nextLine();
        String token = scan.next();
        UI. println (scan.next ());
        if (scan.hasNextInt()){
            UI. println (scan.nextInt ());
        }else{
            scan.next();
        }
        UI. println (scan.nextLine ());
        UI. println (token);
    } catch(IOException e){UI.println("Fail: " + e);}
}
```

(a) [5 marks] What will be printed if this method is called using the example file shown above?

```
Write your answers here
New
NZ6204 Auckland 21:55 6 go to gate
Air
```

(b) [8 marks] Complete the following findGate method which is passed the name of a file and the flight number. The findGate method should go through the file, find the flight number, then find its boarding gate number, and print them out. If the flight number is not found, it should print a "not found" message. For example, using the example file above,

findGate("example.txt", "NZ6204") should print

```
NZ6204 at gate: 6
```

Hint: Note that the airline name, the city name and the notes can have multiple words. You may assume that each line of the file is always in the same format. There is no missing data. The flight number is always one word, and the gate numbers are the only integers in the file.

```
public void findGate(String fileName, String flightNo){
  try {

            Scanner scan = new Scanner (new File(fileName));

            while ( scan.hasNext() ) {
                String n = scan.next();
                if (n.equals(flightNo)){
                    while(!scan.hasNextInt()){
                        scan.next();
                    }
                    if (scan.hasNextInt()){
                            UI.println (flightNo + " at Gate: " + scan.nextInt());
                            return;
                    }
                }
            }
            UI.println (flightNo + " was not found");
        } catch(IOException e){UI.println("Fail: " + e);}
    }

  } catch(IOException e){UI.println("Fail: " + 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.