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

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

Model Solutions
COMP 102: Test 2

11 May, 2015

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 if and while	[5]	
2. Writing with if and while	[7]	
3. Files	[6]	
4. Defining Objects	[7]	
5. Event Driven Input	[7]	
6. Defining Classes	[7]	
7. More Loops	[6]	
	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 if and while [5 marks]

Consider the following printOut method.

```
public void printOut( int num) {

    if (num > 2) {
        UI.println ("num= " + num);
    }
    else {
        UI.println ("The number is small");
    }
    int x = num;
    while (x > 2) {
        UI.println (x);
        x = x − 1;
    }
    UI.println ("Done");
}
```

num: [ ]

x: [ ]

What will be printed if printOut(6) is called?

```
num=6
6
5
4
3
Done
```

What will be printed if printOut(1) is called?

```
The number is small
Done
```

Question 2. Writing with if and while [7 marks]

Complete the following printSeries method so that it uses a loop to print all the integers from the minimum of its two arguments to the maximum.

For example, printSeries(5,8) should print out

```
Numbers from 5 to 8:
5
6
7
8
```

If the first argument is bigger than the second argument, it should swap the values of the two arguments and then print the integers so that printSeries(8,5) should do exactly the same printSeries(5,8) (printing from 5 up to 8).

```
public void printSeries( int min, int max){
    // swap the values of max and min if min is greater than max
    if (min > max) {
        int t = min;
        min = max;
        max = t;
    }

    UI.println("Numbers from " + min + " to " + max + ":");

    int i = min;

    while (i <= max) {
        UI.println(i);
        i = i + 1;          // or i++;
    }

}
```

Question 3. Files [6 marks]

Suppose a shopping list file has information about items to buy. Each line has the name of an item (one word only), the number of that item to buy, and the cost per item, for example:

```
bed 2 1000
lamp 5 20
chair 4 200.00
desk 1 249.99
```

Complete the following totalCost method which is passed the name of a file.

totalCost should

- open a Scanner to read from the file,
- intialise the cost,
- loop through the file, reading the item name, count, and price from each line, adding each count times the price to the total cost.
- print out the total cost of all items in the file.

For example, given the file above, it would print

```
Total cost: 3149.99
```

```
public void totalCost(String fileName){
    try{

        double total = 0;
        Scanner scan = new Scanner (new File(fileName));
        while ( scan.hasNext() ) {
            String name = scan.next();
            double count = scan.nextInt ();
            double price = scan.nextDouble();
            total = total + count * price;
        }

        UI. println ("Total cost: " + total);

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

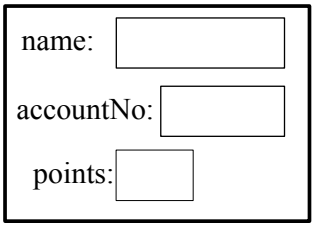
Question 4. Defining Objects [7 marks]

Consider the Account class on the facing page

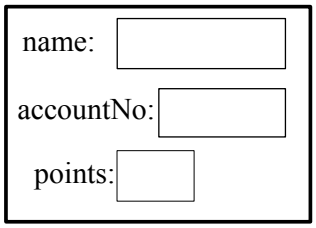
What will the following fragment of code print out?
(You may use the boxes to keep track of the fields in the objects.)

```
Account a = new Account("Adam", 3453, 5);
Account b = new Account("Bob", 5456, 26.5);
UI.println("1------------------");
a.printPoints();
b.printPoints();

UI.println("2------------------");
a.earnPoints();
a.printPoints();

UI.println("3------------------");
b.earnPoints();
b.earnPoints();
b.spendPoints(5);
a.spendPoints(8);

b.printPoints();
a.printPoints();
```

a:



b:




```
1------------------

Adam:  5.0

Bob:  26.5

2------------------

Adam:  15.0

3------------------

Bob:  41.5

Adam:  7.0
```

(Question 4 continued on next page)

(Question 4 continued)

The Account class:

```
public class Account {
    private String name;
    private int accountNo;
    private double points;

    public Account(String n, int aNum, double p){
        this.name = n;
        this.accountNo = aNum;
        this.points = p;
    }

    public void earnPoints(){
        this.points = this.points + 10;
    }

    public void spendPoints(double p){
        this.points = this.points - p;
    }

    public void printPoints (){
        UI. println (this.name + ": " + this.points);
    }
}
```

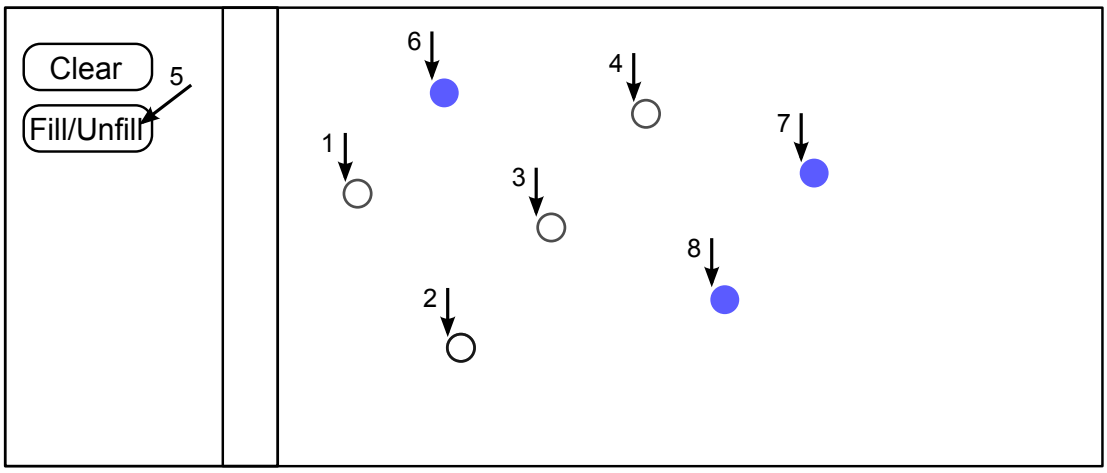
Question 5. Event-Driven Input [7 marks]

Complete the CircleAndDot program on the facing page so that it allows the user to draw a picture made of blue circles and blue dots (filled circles) on the graphics pane. Every time the user releases the mouse at a point on the graphics pane, the program should draw a filled or unfilled blue circle of diameter 10, whose top-left corner is at the point.

The program should have two buttons called "Clear" and "Fill/Unfill":

- When the user clicks the Clear button, the program should clear the graphics pane.
- When the user clicks the Fill/Unfill button, the program should change the state of later circles from fill to unfill or vice versa.

The diagram shows an example picture the program might draw if the user clicked the mouse at four positions, then clicked the Fill/Unfill button, then clicked the mouse at three more positions. (The arrows show where the user clicked.)



(Question 5 continued on next page)

(Question 5 continued)

```
public class CircleAndDot {
    public static final int DIAM = 10; // size of the circles
    private boolean fill = false;



    public CircleAndDot(){
        UI.addButton("Clear", UI::clearGraphics);
        UI.addButton("fill/unfill", this::doFillUnfill);
        UI.setMouseListener(this::doMouse);




    }
    public void doFillUnfill (){
        if (this.fill )
            this.fill = false;

        else
            this.fill = true;




    }
    public void doMouse(String action, double x, double y){
        if (action.equals("released")){
            UI.setColor(Color.blue);
            if (this.fill )
                UI.fillOval (x, y, DIAM, DIAM);
            else
                UI.drawOval(x, y, DIAM, DIAM);
        }




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

Question 6. Defining Classes [7 marks]

For this question, you are to complete part of a program for a food recommender system. The program uses Food objects to store information about food items, including

- a description,
- a price,
- counts of the "likes" and "dislikes", and
- a tag string.

The program allows a user to vote for ("like") or against ("dislike") items, find out if there are more "likes" than "dislikes", and to add information to the end of the tag string.

Initially each food item has zero likes and dislikes and an empty tag string.

Complete the Food class below and on the facing page.

You need to define:

- Fields: to store the state information about the food item.
- The constructor: initialises the food item with a description and a price. Other fields should be initialised to default values.
- The like() method: Increases the number of likes by 1.
- The dislike() method: Increases the number of dislikes by 1.
- The isRecommended() method: Returns a boolean value that is true if the total number of votes is greater than 2 and there are more likes than dislikes.
- The addToTag(...) method: Parameter is a string which should be added to the end of the tag string. It only adds to the tag if the current tag string is not too long (less than 60 characters), otherwise it prints an error message.
- The display() method: Prints the description and price of the food item. If the food is recommended, then its description should be printed in Uppercase.

```
public class Food{
    private String description;
    private double price;
    private String tag ="";
    private int likes = 0;
    private int dislikes = 0;


    public Food(String d, double p){

        this.description = d;
        this.price = p;

    }

//methods on next page
```

(Question 6 continued)

```
//continued from previous page

    public void like (){
        this. likes ++;


    }
    public void dislike (){
        this. dislikes ++;


    }
    public boolean isRecommended(){
        return ((this. likes +this. dislikes >2)&& (this.likes>this. dislikes ));


    }
    public void addToTag(String f) {
        if (this.tag.length()<60)
            this.tag = this.tag + "  " + f;
        else
            UI. println ("sorry, no space");




    }
    public void display(){
        if (this.isRecommended()){
            UI. print (this. description .toUpperCase());
        }
        else
            UI. print (this. description );
        UI. printf (" @ $%.2f %s\n", this.price, this.tag);









    }
}
```

Question 7. More Loops [6 marks]

The following drawStuff method draws filled and empty circles on the graphics pane.

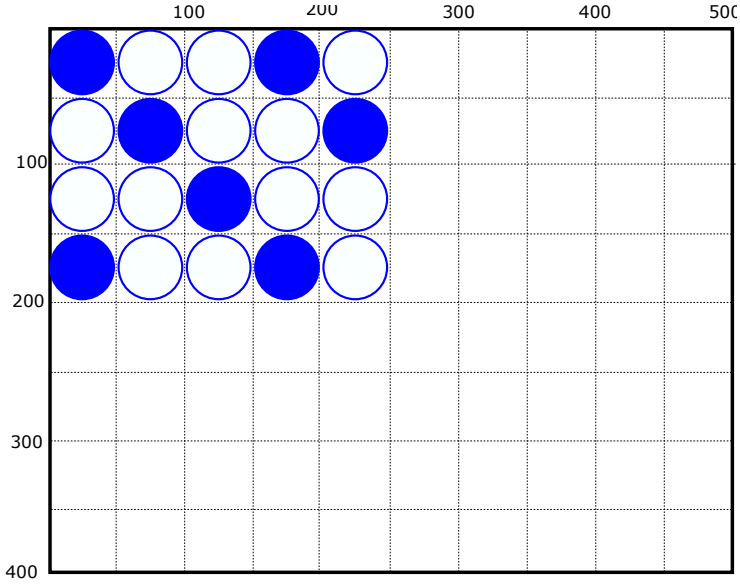
Sketch what it will draw on the grid on the facing page.

```
public void drawStuff(){
    int y = 0;
    boolean last = false;
    boolean prev = false;

    while ( y < 200 ){
        int x =0;
        while ( x <= 200 ){
            if ( !last ){
                UI. fillOval (x, y, 50, 50);
                last = true;
            }
            else {
                UI.drawOval(x, y, 50, 50);
                if (prev){
                    last = !last ;
                }
                prev = !prev;
            }
            x = x + 50;
        }
        y = y + 50;
    }
}
```

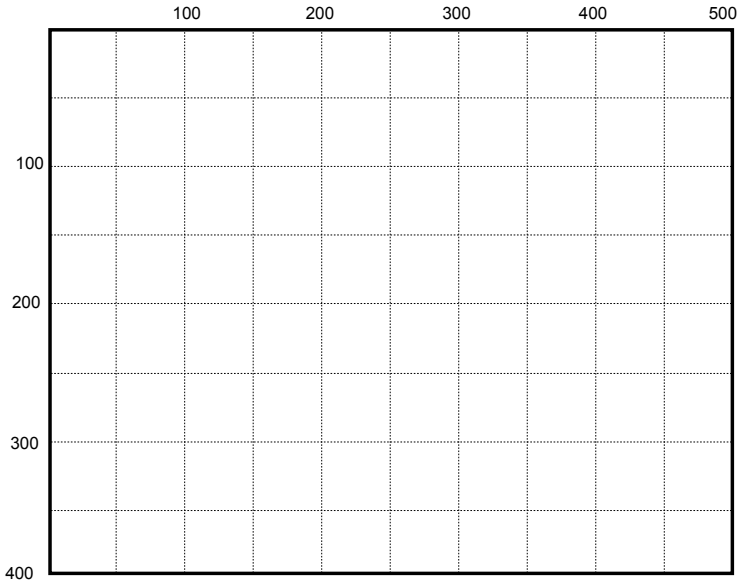
(Question 7 continued on next page)

(Question 7 continued)

Graphics pane grid for your answer:



Spare copy of grid (for if you mess up on the first grid)



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

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.