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

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

Intro to Computer Programming: Final Exam

18 January, 2018

Instructions

- Time allowed: **2 Hours**
- Answer **all** the questions. There are 120 marks in total.
- Write your answers in the boxes in this test paper and hand in all sheets.
- Brief Java documentation is provided with the test
- You may use chinese-english translation dictionaries, and calculators without a full set of alphabet keys.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions	**Marks**	
1. Understanding Java	[35]	
2. Writing Basic Java	[30]	
3. Defining objects	[15]	
4. User Interfaces	[20]	
5. Arraylist	[20]	
	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 Java **[35 marks]**

(a) **[7 marks]** Variables.

What will be printed out by this printValues method?

```
public void printValues(){
    double num = 10;
    UI.println("num = " + num);

    double ten = num + 10;
    double twen = num + 25;
    UI.println("ten = " + ten);
    UI.println("twen = " + twen);

    if (num==ten){
        UI.println("same");
    }
    else {
        UI.println("different");
    }

    ten = twen;
    twen = ten;
    UI.println("ten is now " + ten);
    UI.println("twen is now " + twen);
}
```

num:

ten:

twen:

(Question 1 continued on next page)

(Question 1 continued)

(b) **[7 marks]** Parameters.

What will be printed out if this.doStuff(5, "yes") is called?

```
public void doStuff(double size, String ans) {
    UI.println("size = " + size);

    if (ans.equals("no")) {
        this.printTwo(size-1);
    }
    else if (ans.equals("yes")) {
        this.printTwo(size);
        this.printTwo(size+2);
    }

    UI.println("Done");
}

public void printTwo(double x) {
    UI.println("x is " + x);

    x = x * 2;
    UI.println("answer is " + x);
}
```

size: ☐

ans: ☐

x: ☐

(Question 1 continued on next page)

(Question 1 continued)

(c) **[7 marks]** Understanding **for** loops.

What will be printed if the following doForLoop method is called?

```
public void doForLoop() {
    double value = 3;
    UI.println("Start at " + value);

    for(int i = 0; i< 4; i++) {
        value = value + 2;
        UI.println(i + " -> " + value);
    }

    UI.println("Finally " + value);
}
```

```
Start at
```

(Question 1 continued on next page)

(Question 1 continued)

(d) **[7 marks] While** loops for reading files.

What will the readFile method print out if the file `"data.txt"` contains:

```
5 rice
3 fish
10 beans
2 cabbage
```

```
public void readFile () {
    try {
        int count = 1;
        String last = "none";
        Scanner scan = new Scanner(new File("data.txt"));
        while (scan.hasNext()){
            UI. println ( last );

            int num = scan.nextInt();
            last = scan.next();

            count = count + num;
        }

        UI. println (count + " : " + last);
    } catch(IOException e){UI. println ("error: " + e);}
}
```

(Question 1 continued on next page)

(Question 1 continued)

(e) **[7 marks] While** loops with ArrayLists.

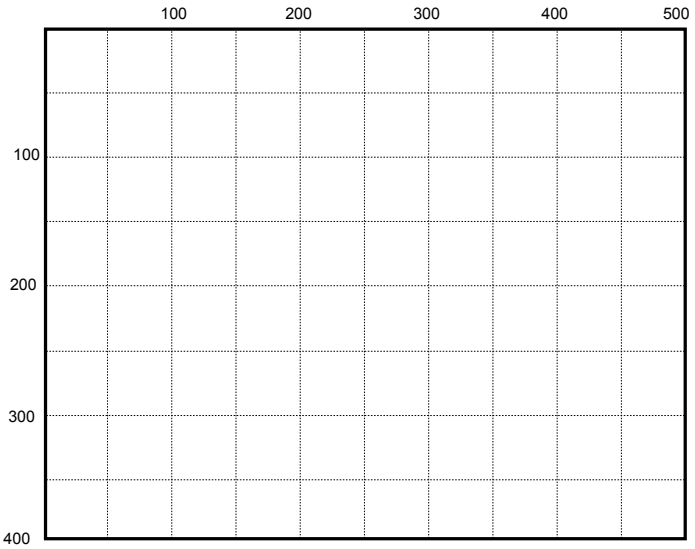
What will the processNumbers method draw if the user enters the following values:

```
Enter numbers:
>:  100
>:  250
>:  20
>:  350
>:  done
```

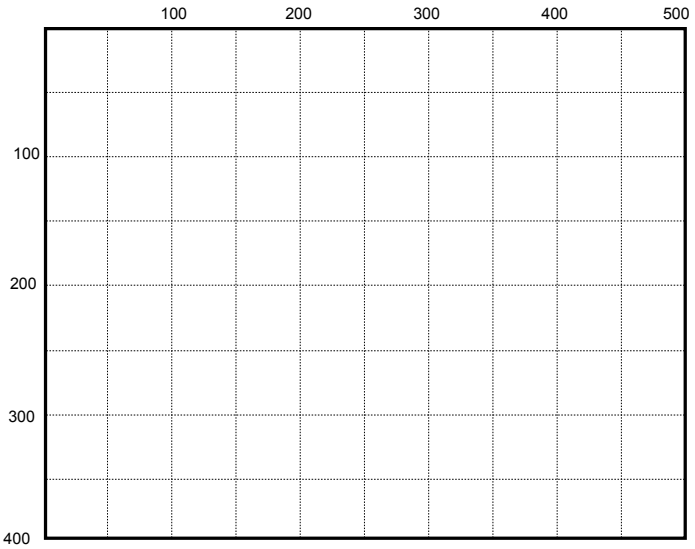
```
public void processNumbers() {
    ArrayList numbers = UI.askNumbers("Enter numbers: ");
    for (double x : numbers){
        UI.drawLine(0, 0, x, 200);
    }
}
```

x:

Draw your answer on this grid:



Spare copy of grid



Question 2. Writing basic Java **[30 marks]**

(a) **[10 marks]** Complete the dayNumber method below so that it returns the number (1 to 5) of a day. The parameter is a String which is the name of the day: `"Mon"`, `"Tue"`, `"Wed"`, `"Thu"`, `"Fri"`.

For example, dayNumber("Mon") should return 1 and dayNumber("Wed") should return 3.

dayNumber returns -1 if the parameter is not a valid day name.

You do not need to worry about Saturday or Sunday.

```
public int dayNumber(String day){









        else {
            return -1; // return -1 if not a valid day name.
        }
}
```

(Question 2 continued on next page)

(Question 2 continued)

(b) **[10 marks]** Complete the printSeries method below that will print the values of the series

1 3 9 27 81 243 729 2187 ...

where each number is 3 times the previous number.

printSeries should not print numbers larger than max.

For example, printSeries(200) should print:

1 3 9 27 81

printSeries(888) should print:

1 3 9 27 81 243 729

Hint: use a **while** loop with a variable that starts at 1.

```
public void printSeries ( int max){




}
```

(Question 2 continued on next page)

(Question 2 continued)

(c) **[10 marks]** Complete the printLongNames method below that will read city names from a file, and print out all the names that have more than 7 characters in the name.

For example, if the file `"fujian-cities"` contains

```
Changle Fuan Fuding Fuqing Fuzhou Jian'ou Jianyang Jinjiang
Longhai Longyan Nan'an Nanping Ningde Putian Quanzhou Sanming
Shaowu Shishi Wuyishan Xiamen Yongan Zhangping Zhangzhou
```

then printLongNames("fujian-cities") should print out:

```
Jianyang Jinjiang Quanzhou Wuyishan Zhangping Zhangzhou
```

Hint: You may use the length() method to find the number of characters in a String.

```
public void printLongNames(String fname){
    try{
        Scanner scan = new Scanner(new File(fname));









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

Question 3. Defining objects **[15 marks]**

Complete the RedEnvelope class that describes RedEnvelope objects.

ARedEnvelope object should contain:

- one field to store the amount of money in the RedEnvelope
- one field to store the name of the person it is given to.
- a constructor with one String parameter - the name of the person. The constructor should store the value in the name field.
- an addMoney method with one double parameter - the amount to add. The amount should be added to the amount field.
- a give method, which should print out a message with the name and the amount.

For example, the following code

```
RedEnvelope redEnv = new RedEnvelope("Lu WeiMei");
redEnv.add(100);
redEnv.add(500);
redEnv.give ();
```

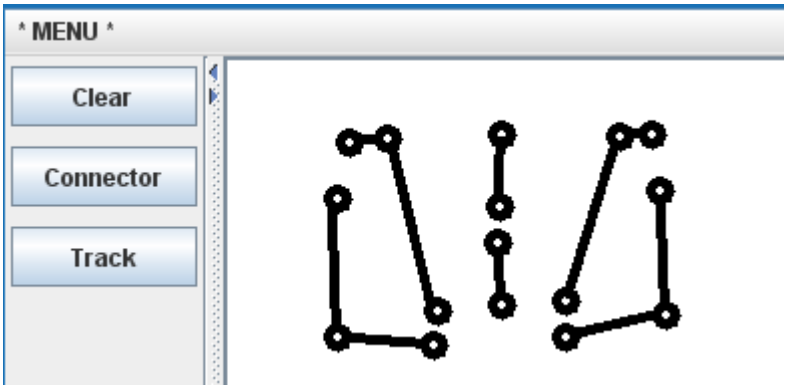
should print out

```
Hello Lu Weimei, this is a gift of 600.0 RMB
```

```
public class RedEnvelope {




    public RedEnvelope(String nm) {



    }
    public void add(double amt) {



    }

    public void give() {



    }
}
```

Question 4. User interfaces. **[20 marks]**

The LayoutDrawer program lets a user draw simple printed-circuit layout diagrams, such as:



These diagrams contain

- **Connector**s: small circles (diameter 10) representing holes for components
- **Track**s: straight lines representing connections.

After clicking the `Connector` button, the user can draw connectors by releasing the mouse on the graphics pane.

After clicking the `Track` button, the user can draw tracks by pressing the mouse at the beginning and releasing the mouse at the end.

Complete

- the LayoutDrawer constructor
- doSetConnector,
- doSetTrack, and
- doMouse.

(Question 4 continued on next page)

(Question 4 continued)

```
public class LayoutDrawer{
    private String tool = "connector"; // or "track"
    private double pressedX;
    private double pressedY;

    public LayoutDrawer(){
        UI.setLineWidth(5);
        UI.setMouseListener( this::doMouse );
        UI.addButton("Clear", UI::clearGraphics);




    }
    public void doSetConnector(){    // Remember user is drawing connectors


    }
    public void doSetTrack(){        // Remember user is drawing tracks


    }
    public void doMouse(String action, double x, double y) {
        if (action.equals("pressed")){
            //remember pressed position



        }
        else if (action.equals("released")){
            //draw a connector (circle) or a track (line)









        }
    }
}
```

Question 5. Arraylists. **[20 marks]**

The ProcessExperiment program reads numbers from a file into an ArrayList stored in a field called data.

```
private ArrayList<Double> data = new ArrayList<Double>();
```

It then processes and displays the numbers.

(a) **[7 marks]** Complete the readExpData method below. The parameter fileName is the name of a file that contains a sequence of numbers.

readExpData should make sure the data field contains an empty ArrayList. It then reads all the numbers from the file into the ArrayList.

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









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

(Question 5 continued on next page)

(Question 5 continued)

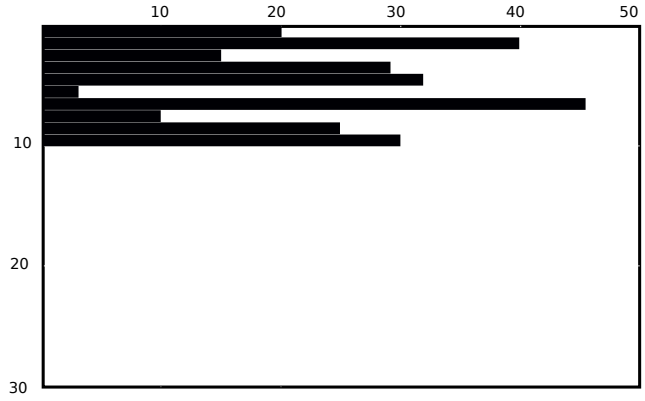
(b) **[6 marks]** Complete the displayData method below.

displayData should plot all the values in the data field on the graphics pane. For each value, it draws a horizontal line from the left side of the window. The length of the line should be the value.

For example, if the values in the ArrayList were:

20 40 15 29 31 2 45 10 25 30

then displayData should draw a plot like this (but no numbers)




```
public void displayData(){




}
```

(Question 5 continued on next page)

(Question 5 continued)

(c) **[7 marks]** Complete the processData method below. The parameter max is the largest value that should be allowed in the data.

processData should remove all the negative values in the ArrayList in the data field, and should replace any value larger than max by max.

For example, if the values in the ArrayList were:

365 –95 –1 256 45 –140 210 313 126

- processData(300) should change the ArrayList to contain

  300 256 45 210 300 126

- processData(200) should change the ArrayList to contain

  200 200 45 200 200 126

```
public void processData(double max){




}
```

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