

Family Name: .....

Other Names: .....

ID Number: .....

Signature .....

## Model Solutions

### COMP 102: Test 2

5 May, 2011

#### Instructions

- Time allowed: **45 minutes**
- There are 45 marks in total.
- Answer **all** the questions.
- 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 paper translation dictionaries, and calculators without a full set of alphabet keys.
- You may write notes and working on this paper, but make sure it is clear where your answers are.

#### Questions

#### Marks

1. Basic Java

[17]

2. Event Driven Input

[8]

3. Defining Classes

[10]

4. File Processing

[10]

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. Basic Java**

[17 marks]

(a) [4 marks] Consider the following numbers method

```
public void numbers(){
    int num = 1;
    int ans = 0;
    while (num < 5){
        Ul.println (ans);
        ans = ans + num;
        num = num + 1;
    }
    Ul.println ("finally " + ans);
}
```

What will be printed if numbers() is called?

```
0
1
3
6
finally 10
```

(b) [3 marks] Consider the following choose method

```
public void choose(int a, int b, int c){
    if (a < b) { Ul.println ("Monday"); }
    else if (a > b && b == c) { Ul.println ("Wednesday"); }
    else if (c == a || c == b) { Ul.println ("Friday"); }
    else { Ul.println ("the weekend"); }
}
```

What will be printed if choose(4, 7, 7) is called?

```
Monday
```

What will be printed if choose(7, 7, 4) is called?

```
the weekend
```

What will be printed if choose(7, 4, 7) is called?

```
Friday
```

**(Question 1 continued)**

**(c)** [4 marks] Suppose the file called `WordsAndNumbers.txt` contains the following text:

```
Hi 3
Program 2
Apple 4
Good 1
```

What will the following `filePrint` method print out?

```
public void filePrint (){
    try{
        Scanner scan = new Scanner(new File("WordsAndNumbers.txt"));
        while (scan.hasNext()){
            String w = scan.next();
            int n = scan.nextInt ();
            while (n > 0){
                UI.print (w);
                n = n-1;
            }
            UI.println ();
        }
        scan.close();
    }
    catch(IOException e){UI.println("Fail: " + e);}
}
```

```
HiHiHi
ProgramProgram
AppleAppleAppleApple
Good
```

(Question 1 continued on next page)

**(Question 1 continued)**

**(d)** [6 marks] Consider the Fish class shown below.

What will the following fragment of code print out?

```
Fish f1 = new Fish("Nino");
f1.report();
Fish f2 = new Fish("Blote");
f1.meet("Debian");
f1.report();
f2.meet("Dori");
f1.teem(5);
f2.report();
f1.report();
```

```
Nino : 1 : -
Nino : 2 : Debian
Blote : 2 : Dori
Nino : 7 : Nino
```

---

```
public class Fish{
    private String name;
    private int count;
    private String friend = "-";

    public Fish(String nm){
        this.name = nm;
        this.count = 1;
    }
    public void report(){
        Ul.println (this.name + " : " + this.count + " : " + this.friend);
    }
    public void meet(String str){
        this.friend = str;
        this.count = this.count + 1;
    }
    public void teem(int n){
        this.friend = this.name;
        this.count = this.count + n;
    }
}
```

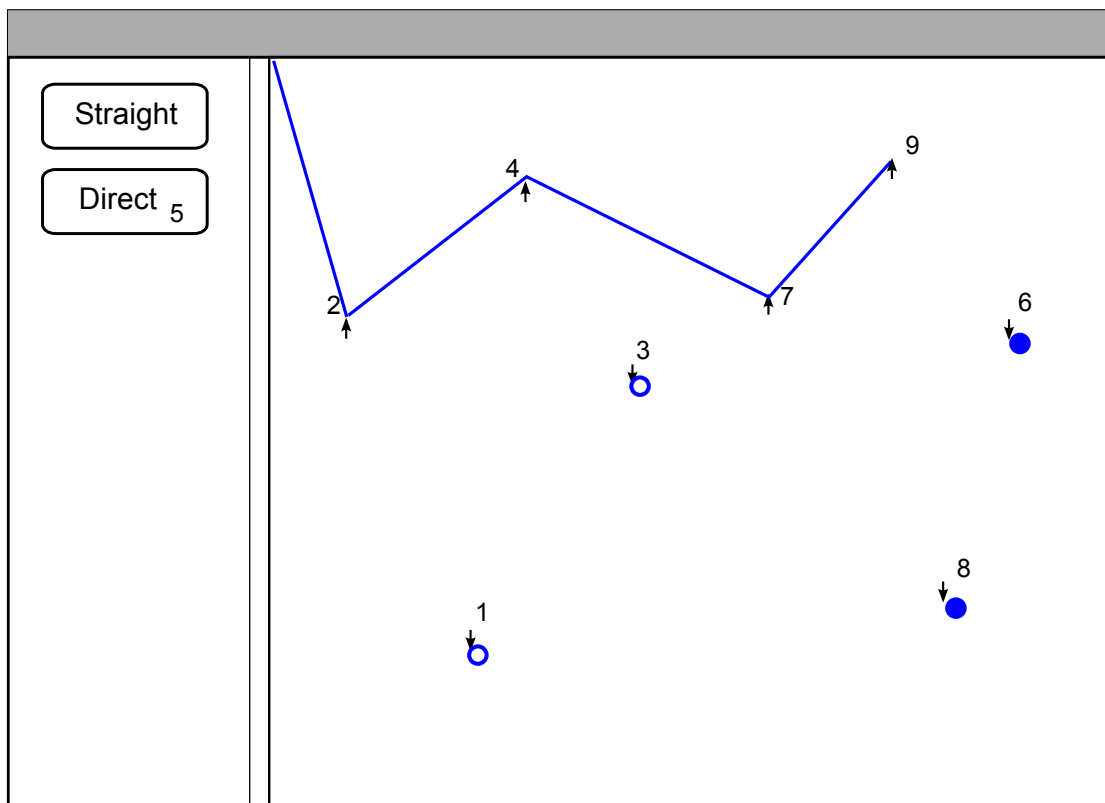
## Question 2. Event-Driven Input

[8 marks]

Consider the PatternMaker class on the facing page.

Sketch below what the program would draw on the graphics pane if the user took the following mouse actions in sequence:

1. press at point 1 and release mouse at point 2
2. press at point 3 and release mouse at point 4
3. press the "Direct" button
4. press at point 6 and release mouse at point 7
5. press at point 8 and release mouse at point 9



(Question 2 continued on next page)

## (Question 2 continued)

```

public class PatternMaker implements UIButtonListener, UIMouseListener{
    // Fields
    private String what = "straight";
    private double lastX = 0;
    private double lastY = 0;

    // Set up interface
    public PatternMaker(){
        UI.setMouseListener(this);
        UI.addButton("Straight", this);
        UI.addButton("Direct", this);
    }

    // Respond to button presses
    public void buttonPerformed(String button){
        if (button.equals("Straight")){
            this.what = "straight";
        }
        else if (button.equals("Direct")){
            this.what = "direct";
        }
    }

    public void mousePerformed(String action, double x, double y) {
        // Respond to "pressed" events
        if (action.equals("pressed")) {
            if (this.what.equals("straight")){
                UI.drawOval(x, y, 10, 10);
            }
            else if (this.what.equals("direct")){
                UI.fillOval (x, y, 10, 10);
            }
        }
        // Respond to "released" events
        else if (action.equals("released")) {
            UI.drawLine(this.lastX, this.lastY, x, y);
            this.lastX = x;
            this.lastY = y;
        }
    }

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

```

### Question 3. Defining Classes

[10 marks]

Consider the FuelGauge class on the facing page.

A FuelGauge object keeps track of the capacity of a fuel tank and the amount of fuel left, and displays it as a simple red vertical bar in a rectangle, as in the example below, which shows a gauge one third full (eg, 10 litres of fuel when the tank capacity is 30 litres).



You need to complete the FuelGauge class:

- Define two fields to store the capacity of the tank (maximum number of litres in the tank) and the current number of litres of fuel remaining.
- Complete the constructor, which should have one parameter which specifies the capacity of the FuelGauge object, and should draw the gauge on the graphics pane.
- Complete the fill method, which has no parameters. It should fill the tank to capacity, then re-draw the fuel gauge on the screen.
- Complete the use method, which should have one parameter—the amount of fuel to use up. It should reduce the current amount of fuel by the value of the parameter, and then re-draw the gauge. If all the fuel in the tank is used, the fuel should be zero—it should never be negative.
- Complete the reDraw method, which has no parameters. It should erase the FuelGauge and redraw it with the bar showing how full the tank is.  
Note that fields for the position and size of the gauge are provided, as is the part of the reDraw method that erases and draws the outline of the gauge.

As an example of using the FuelGauge class, the testGauge method below creates a gauge, fills it, and then uses 40 litres of fuel, leaving it one third full.

```
public void testGauge(){
    FuelGauge fg = new FuelGauge(60);
    fg.fill ();
    fg.use(15);
    fg.use(25);
}
```

(Question 3 continued on next page)



## (Question 3 continued)

```
public class FuelGauge{
    private double capacity;
    private double fuel = 0;

    //dimensions
    private double left = 40;
    private double top = 120;
    private double base = 220;
    private final double width = 10;
    private final double height = 100;

    public FuelGauge( double cap){
        this.capacity = cap;
        this.reDraw();
    }
    public void fill ( ){
        this.fuel = this.capacity;
        this.reDraw();
    }
    public void use( double amount){
        this.fuel = this.fuel - amount;
        if (this.fuel < 0) {
            this.fuel = 0;
        }
        this.reDraw();
    }
    public void reDraw(){
        UI.eraseRect(this.left , this.top, this.width, this.height);
        //draw the bar
        UI.setColor(Color.red);

        double fuelHeight = this.fuel / this.capacity * this.height;
        UI.fillRect (this.left , this.base - fuelHeight, this.width, fuelHeight);

        //draw the outline
        UI.setColor(Color.black);
        UI.drawRect(this.left , this.top, this.width, this.height);
    }
}
```

#### Question 4. File Processing

[10 marks]

Suppose the file “`timetable.txt`” contains information about the departure times, platform number, and destination of trains from Wellington station. For example, part of the file might be:

```
0930 1 Johnsonville
0949 3 Upper Hutt
0958 2 Melling
1002 4 Johnsonville
1010 2 Upper Hutt
1024 3 Melling
1038 1 Johnsonville
```

where each line contains a time (in 24 hour format), a platform number, and a destination.

Complete the `chooseTrains` method on the facing page. `chooseTrains` has two parameters — a destination station and a time. The method should read the “`timetable.txt`” file, and print out the platform and departure time of all trains to the specified destination which leave at or before the specified time. It should also print out the number of trains to that destination that leave after the specified time.

For example, if the file contained just the data above, then calling `chooseTrains("Johnsonville", 1030);` should print out

```
Trains to Johnsonville before 1030:
  platform 1 at 930
  platform 4 at 1002
There are 1 later trains
-----
```

Note: the method should **not** ask the user for the destination and time.

(Question 4 continued on next page)

## (Question 4 continued)

```
public void chooseTrains(String dest, int time){
    try{
        Scanner scan = new Scanner(new File("timetable.txt"));
        UI.printf ("Trains to %s before %d:\n", dest, time);
        int countLate = 0;
        while (scan.hasNext()){
            int t = scan.nextInt ();
            int platform = scan.nextInt ();
            String station = scan.nextLine();
            if (station .contains(dest)){
                if (t <= time){
                    UI.printf (" platform %d at %d\n", platform, t);
                }
                else{
                    countLate++;
                }
            }
        }
        UI.printf ("There are %d later trains\n", countLate);
        UI.println ("-----");

    }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.