

Family Name:

Other Names:

ID Number:

Signature.....

COMP102: Test 1

30 March, 2012

Instructions

Model Solutions

- Time allowed: **45 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.
- If you think some question is unclear, ask for clarification.
- Brief Java documentation is provided with the test
- This test contributes 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 your answers are clear.

Questions

Marks

1. Components of Java Programs

[7]

2. Understanding variables

[8]

3. Defining a Method

[10]

4. Calling methods

[6]

5. Using Objects

[6]

6. Extracting Methods

[8]

TOTAL:

Please answer the following question. (Your answer will not affect your mark in any way.)

How much programming had you done before starting the course?

Little or none

Some (used variables, if's, and loops)

Lots (eg, used arrays, defined methods/functions with parameters, used libraries)

Question 1. Components of Java Programs

[7 marks]

The questions on the facing page refer to the code below.

```
1 import comp102.*;
2 public class Question1 {
3
4     public void insuranceClaim(){
5         int vehicleCategory = UI.askInt("Vehicle Type: ");
6         double repairCost = this.computeCost(vehicleCategory);
7         double cover = UI.askDouble("Maximum coverage available: ");
8         this.reportClaim(repairCost, cover);
9     }
10
11    public double computeCost(int type){
12        double parts = UI.askDouble("Cost of parts: ");
13        double partsCost = 200 + parts * 1.5;
14        double labour = 500 * type;
15        return (partsCost + labour);
16    }
17
18    public void reportClaim(double cost, double cover){
19        UI.printf ("Estimated cost of repair is $%4.2f\n", cost);
20        UI.printf ("Payment will be $%4.2f\n", Math.min(cover, cost));
21    }
22
23 }
```

(Question 1 continued on next page)

ANSWERS

Student ID:

(Question 1 continued)

The following questions refer to the code on the facing page:

(a) [1 mark] Which line or lines contain the header of a method definition? (list the line numbers.)

4, 11, 18

(b) [1 mark] What type of value is returned by the computeCost method?

double

(c) [1 mark] Which line or lines contain a literal String?

5, 7, 12, 19, 20

(d) [1 mark] Which line or lines contain contain a literal integer value?

13, 14

(e) [1 mark] What types of arguments does the reportClaim method require?

doubles

(f) [1 mark] Which line or lines contain a call to a method that returns a value?.

5, 6, 7, 12, 20

(g) [1 mark] Which line or lines contain an declaration of a variable?

5, 6, 7, 12, 13, 14.
(the parameter declarations in lines 11 and 18 are also acceptable)

Question 2. Understanding programs with variables

[8 marks]

Consider the `sortNumbers` method below. In the box on the facing page (p5), write the output that `sortNumbers` would produce. Note that the first part of each line of output is provided for you.

Hint: draw a box for each variable and keep track of the value that is put into it.

```
public void sortNumbers(){
    String stage = "Start";
    int x = 2;
    int y = 5;
    int z = 9;

    Ul.println ("stage = " + stage);
    Ul.println ("largest = " + x);
    Ul.println ("middle = " + z);
    Ul.println ("smallest = " + y);

    x = y + z;
    y = x - 3;
    stage = stage + "-" + z;

    Ul.println ("stage = " + stage);
    Ul.println ("largest = " + y);
    Ul.println ("middle = " + z);
    Ul.println ("smallest = " + x);

    stage = "end" + z + x;
    x = y;
    y = z;
    z = x;

    Ul.println ("stage = " + stage);
    Ul.println ("largest = " + z);
    Ul.println ("middle = " + y);
    Ul.println ("smallest = " + x);
}
```

(Question 2 continued on next page)

(Question 2 continued)

```
stage    =    Start
largest  =    2
middle  =    9
smallest =    5
stage    =    Start-9
largest  =    11
middle  =    9
smallest =    14
stage    =    end914
largest  =    11
middle  =    9
smallest =    11
```

ANSWERS

Question 3. Defining a Method

[10 marks]

Ripoff Rentals rents short-term apartments by the week. They charge a \$100 signing fee when you start renting, \$150 base rent per week, plus \$90 per week per bedroom.

For example, for a 5 week rental of a 2 bedroom apartment, the total cost would be \$1750 (= \$100 + \$330 × 5).

Complete the following ripoffRent() method for calculating rent. The method should

- ask the user for the number of bedrooms
- calculate and print the rental cost per week
- ask for the number of weeks
- calculate and print total cost of renting the apartment.

For example, if the user entered 2 bedrooms and 5 weeks, the interaction should look like:

```
How many bedrooms: 2
Rental per week = $330
How many weeks: 5
Total cost = $1750
```

```
public void ripoffRent(){
    int bedrooms = UI.askInt("How many bedrooms: ");
    int perWeek = 150+bedrooms*90;
    UI.println ("Rental per week = $" + perWeek);
    int weeks = UI.askInt("How many weeks: ");
    int cost = 100 + (perWeek * weeks);
    UI.println ("Cost = $" + cost);

// OR (using doubles and printf)

    double bedrooms = UI.askDouble("How many bedrooms: ");
    double perWeek = 150+bedrooms*90;
    UI.printf ("Rental per week = $%.2f\n", perWeek);
    double weeks = UI.askDouble("How many weeks: ");
    double cost = 100 + (perWeek * weeks);
    UI.println ("Cost = $%.2f\n", cost);

}
```

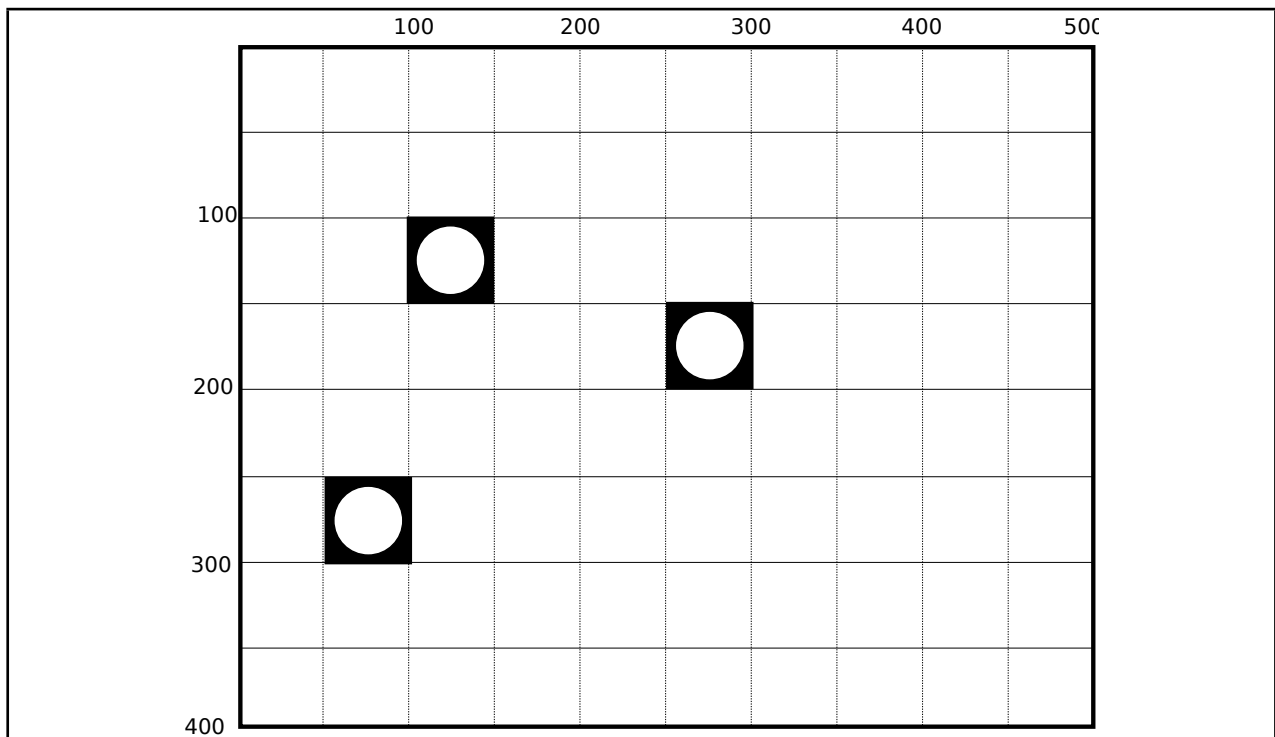
Question 4. Calling methods

[6 marks]

Sketch what the following drawGame method would draw on the graphics pane.
Note that drawGame calls drawPiece which is also defined below.

```
public void drawGame(){
    this.drawPiece(100, 100);
    double x = 250;
    this.drawPiece(x, x-100);
    double y = 50;
    this.drawPiece(y, x);
}

public void drawPiece(double x, double y){
    UI.setColor(Color.black);
    UI.fillRect(x, y, 50, 50);
    UI.setColor(Color.white);
    UI.fillOval(x+5, y+5, 40, 40);
}
```



The grid lines and coordinates are just to help you sketch your answer.

There is an extra copy of the grid on the next page if you need it.

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 5. Using objects

[6 marks]

Suppose the HummingBird class has one constructor and two methods:

Constructor:

```
public HummingBird(Color col)
// constructs a HummingBird object, of the specified color , and displays it
// on the window at a random place
```

Methods:

```
public void fly (double dist)
// If dist is positive , makes the HummingBird move forward by the specified distance
// If the argument is negative , then it moves backwards.
// Either movement will take 1 second.
```

```
public void hover()
// makes the HummingBird flap its wings very fast in the same position
// for 1 second.
```

Complete the following animate method, which should create a red HummingBird object and then make it move forward 30 units over 1 second, flap its wings for 2 seconds, then move backward to its original position over 3 seconds.

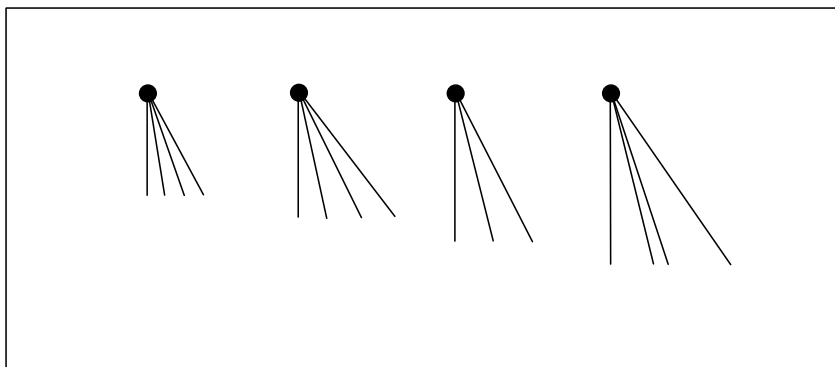
```
public void animate(){
    HummingBird hb = new HummingBird(Color.red);
    hb.fly (30);
    hb.hover();
    hb.hover();
    hb.fly (-10);
    hb.fly (-10);
    hb.fly (-10);
//OR (a bit nicer)
    HummingBird hb = new HummingBird(Color.red);
    double d = 30;
    hb.fly (d);
    hb.hover();
    hb.hover();
    hb.fly (-d/3);
    hb.fly (-d/3);
    hb.fly (-d/3);
}
```

Question 6. Extracting Methods

[8 marks]

The following `drawLightRow` method draws a pattern of circles with light rays coming from them (as shown in the figure below).

```
public void drawLightRow(){
    double ceiling = UI.askDouble("Level of ceiling");
    UI.fillOval(95, ceiling - 5, 10, 10);
    UI.drawLine(100, ceiling, 100, ceiling + 50);
    UI.drawLine(100, ceiling, 110, ceiling + 50);
    UI.drawLine(100, ceiling, 120, ceiling + 50);
    UI.drawLine(100, ceiling, 130, ceiling + 50);
    UI.fillOval(195, ceiling - 5, 10, 10);
    UI.drawLine(200, ceiling, 200, ceiling + 70);
    UI.drawLine(200, ceiling, 220, ceiling + 70);
    UI.drawLine(200, ceiling, 240, ceiling + 70);
    UI.drawLine(200, ceiling, 260, ceiling + 70);
    UI.fillOval(295, ceiling - 5, 10, 10);
    UI.drawLine(300, ceiling, 300, ceiling + 90);
    UI.drawLine(300, ceiling, 330, ceiling + 90);
    UI.drawLine(300, ceiling, 360, ceiling + 90);
    UI.drawLine(300, ceiling, 300, ceiling + 90);
    UI.fillOval(395, ceiling - 5, 10, 10);
    UI.drawLine(400, ceiling, 400, ceiling + 110);
    UI.drawLine(400, ceiling, 440, ceiling + 110);
    UI.drawLine(400, ceiling, 480, ceiling + 110);
    UI.drawLine(400, ceiling, 430, ceiling + 110);
}
```



(Question 6 continued on next page)

(Question 6 continued)

The `drawLightRow` method is not well designed: it has quite a bit of repetition, and repeated literal constants specifying various dimensions. It would be better design to define another method called `drawLight` which draws a single "light" (circle with light rays) and make `drawLightRow` call the `drawLight` method four times, as in the version of `drawLightRow` below.

Complete the definitions of `drawLightRow` and `drawLight` below. You will need to determine the appropriate arguments for the calls to `drawLight`, the appropriate parameters in the definition of `drawLight`, as well as the statements in `drawLight`.

```

public void drawLightRow(){
    double ceiling = UI.askDouble("Level of ceiling");

    this.drawLight( 100, ceiling , 50, 10 );
    this.drawLight( 200, ceiling , 70, 20 );
    this.drawLight( 300, ceiling , 90, 30 );
    this.drawLight( 400, ceiling , 110, 40 );
}

public void drawLight( double x, double y, double depth, int step ){
    UI.fillOval (x-5, y-5, 10, 10);
    double bot = y + depth;
    UI.drawLine(x, y, x, bot);
    UI.drawLine(x, y, x + step%90, bot);
    UI.drawLine(x, y, x + (2*step)%90, bot);
    UI.drawLine(x, y, x + (3*step)%90, bot);

    // the %90 was just plain nasty! there were other correct ways of doing it .
    // There were also some other patterns that you could have taken advantage of
    // - in the examples, the step size was always 1/10 of the x position .
    // - the depth was 30 + 0.2 * x

}

```

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.