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

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

COMP102: Test 1

24 March, 2011

Model Solutions

Instructions

- 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	[5]	
5. Using Objects	[5]	
6. Extracting Methods	[10]	
	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)

ANSWERS

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 planWall(){
5          int doors = UI.askInt("Number of doors:");
6          double width = UI.askDouble("Width of wall");
7          double paint = this.computePaint(width*2.5, doors);
8          UI.println("Paint Required: " + paint + " litres");
9          this.planFraming(width, 2.5);
10     }
11
12     public double computePaint(double area, double doors){
13         double sqMetersPerLitre = 8.5;
14         double paintArea = area - doors * 1.9;
15         return (paintArea / sqMetersPerLitre);
16     }
17
18     public void planFraming(double length, double height){
19         double vertical = height * (length/0.7);
20         double dwangs = length * 3;
21         UI.printf("Framing Required: %4.2f meters\n", (vertical+dwangs));
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 an assignment statement? (list the line numbers.)

5, 6, 7, 13, 14, 19, 20

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

5, 6, 8, 21

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

double

(d) [1 mark] Which line or lines contain a method call?

5, 6, 7, 8, 9, 21

(e) [1 mark] How many arguments does the planFraming method require?

2

(f) [1 mark] Which line or lines specify the value that is returned by computePaint.

15

(g) [1 mark] List three operators used in the program

+, *, −

ANSWERS

Question 2. Understanding programs with variables [8 marks]

What will the following game method print out?

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

```
public void game(){
    int  games = 0;
    String  frst  = "Adam";
    String  secnd = "Brad";
    String  thrd  = "Cara";
    UI. println ("After  " + games + "  games:");
    UI. println ( frst  + "  " + secnd + "  " + thrd);

    games = games + 1;
    frst  = "Alex";
    UI. println ("After  " + games + "  games,  winner  is  " + frst);

    games = games + 1;
    secnd = frst ;
    frst  = secnd;
    thrd  = thrd  + thrd ;
    UI. println ("After  " + games + "  games:");
    UI. println ( frst  + "  " + secnd + "  " + thrd);

    games = (games + 3) / 4;
    secnd = thrd;
    thrd  =  frst ;
    frst  = secnd;
    UI. println ("After  " + games + "  games:");
    UI. println ( frst  + "  " + secnd + "  " + thrd);
}
```

```
After 0 games:
Adam Brad Cara
After 1 games, winner is Alex
After 2 games:
Alex Alex CaraCara
After 1 games:
CaraCara CaraCara Alex
```

Question 3. Defining a Method [10 marks]

The MagicTurf company sells rolls of ready-made grass for making lawns. Each roll covers 2 square meters, and costs $11 dollars. The company charges $35 for delivery, regardless of the number of rolls. You are to complete the following lawnCost method for calculating the cost of a lawn. The method should

- ask the user for the length and width of the lawn in meters
- calculate the area
- calculate the cost of the lawn (including delivery)
- print out the area of the lawn
- print out the number of rolls and the total cost of of the lawn.

You may assume that the length and width will be a whole number of meters.

For example, if the user entered a length of 3 and width of 13, the method should print out

```
Area = 39 square meters
Number of rolls = 20
Total cost = $255
```

```
public void lawnCost(){
    int length = UI.askInt("Length");
    int width = UI.askInt("Width");
    int area = length * width;
    int numRolls = (area+1) / 2;
    int cost = numRolls * 11 + 35;
    UI.println("Area = "+area+" square meters");
    UI.println("Number of Rolls = "+numRolls);
    UI.println("Total cost = $"+cost);









}
```

Note, You will still get most of the marks if your program would print out 19.5 rolls with the input above.

ANSWERS

Question 4. Calling methods [5 marks]

What will the following printGame method print out? Note that printGame calls printPiece which is also defined below.

Hint: the %3d in the printf statements means to print an integer, padding it with spaces to be at least 3 characters wide.

```
public void printGame(){
    int x = 50;
    int y = 100;
    this.printPiece(x, y);
    this.printPiece(x-10, y-20);
    this.printPiece(y, x);
}

public void printPiece(int x, int y){
    UI.printf("+-------+\n");
    UI.printf("| %3d |\n", x );
    UI.printf("| %3d |\n", y );
    UI.printf("+-------+\n");
    UI.println();
}
```

```
+-------+
|  50   |
| 100   |
+-------+

+-------+
|  40   |
|  80   |
+-------+

+-------+
| 100   |
|  50   |
+-------+
```

ANSWERS

Student ID: .....

Question 5. Using objects [5 marks]

Suppose the Surprise class has one constructor and one method:

Constructor:

```
public Surprise(String word)
// constructs a Surprise object, which stores the word that was the
// argument to the constructor
```

Methods:

```
public void sayBoo()
// prints out "Boo" followed by the word that was stored in the object
// when it was constructed.
```

If you call the following clownGame method, what will it print out?

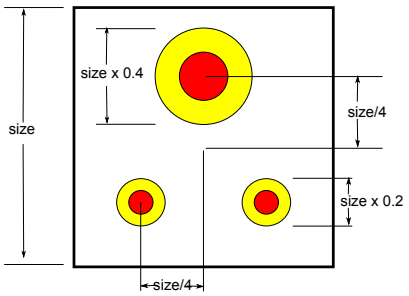
```
public void clownGame(){
    Surprise one = new Surprise("pumpkin");
    Surprise two = new Surprise("squash");
    one.sayBoo();
    two.sayBoo();
    this.together(two, one);
}

public void together(Surprise a, Surprise b){
    UI.println("Now together:");
    a.sayBoo();
    b.sayBoo();
}
```

```
Boo pumpkin
Boo squash
Now together:
Boo squash
Boo pumpkin
```

Question 6. Extracting Methods [10 marks]

The following drawArcheryFlag method draws a square flag on the Graphics pane containing three "targets" - red circles inside yellow circles:




```
public void drawArcheryFlag(){
    double size = UI.askDouble("Flag size");
    UI.setColor(Color.black);
    UI.drawRect(200−size/2, 150−size/2, size, size);

    UI.setColor(Color.yellow);
    UI. fillOval  (200−size∗0.2, (150−size/4)−size∗0.2, size∗0.4, size∗0.4);
    UI.setColor(Color.red);
    UI. fillOval  (200−size∗0.1, (150−size/4)−size∗0.1, size∗0.2, size∗0.2);

    UI.setColor(Color.yellow);
    UI. fillOval  ((200−size/4)−size∗0.1, (150+size/4)−size∗0.1, size∗0.2, size∗0.2);
    UI.setColor(Color.red);
    UI. fillOval  ((200−size/4)−size∗0.05, (150+size/4)−size∗0.05, size∗0.1, size∗0.1);

    UI.setColor(Color.yellow);
    UI. fillOval  ((200+size/4)−size∗0.1,  (150+size/4)−size∗0.1,  size∗0.2,  size∗0.2);
    UI.setColor(Color.red);
    UI. fillOval  ((200+size/4)−size∗0.05, (150+size/4)−size∗0.05, size∗0.1, size∗0.1);
}
```

The drawArcheryFlag method is not well designed: it has quite a bit of repetition, repeated literal constants (200, 150) specifying dimensions of the flag, and complicated expressions. It would be better design to define and use some variables and to define another method called drawTarget which draws a single target and make drawArcheryFlag call the drawTarget method three times, as in the version of drawArcheryFlag on the facing page.

Complete the definitions of drawArcheryFlag and drawTarget on the facing page. You will need to determine the appropriate arguments for the calls to drawTarget, the appropriate parameters in the definition of drawTarget, as well as the statements in drawTarget.

(Question 6 continued on next page)

(Question 6 continued)

```
public void drawArcheryFlag(){
    double size = UI.askDouble("Flag size");
    double x = 200;    // center of the flag
    double y = 150;
    UI.setColor(Color.black);
    UI.drawRect(x-size/2, y-size/2, size, size);

    double offset = size/4;

    this.drawTarget(x, y-offset, size*0.4                    );

    this.drawTarget(x-offset, y+offset, size*0.2             );

    this.drawTarget(x+offset, y+offset, size*0.2             );
}

public void drawTarget( double x, double y, double size                  ){
    UI.setColor(Color.yellow);
    UI. fillOval (x-size/2, y-size/2, size, size);
    UI.setColor(Color.red);
    UI. fillOval (x-size/4, y-size/4, size/2, size/2);









}
```

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.

****