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

ID Number: . . . . . .

COMP102: Test 2

Model Solutions

10 Apr, 2008

Instructions

- Time allowed: **90 minutes** ($1\frac{1}{2}$ hours).
- There are 90 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.
- At the end of the test paper, there is some Java documentation, and model code from Assignment 5, including MovingShape.java.
- This test will contribute 18% of your final grade, if it helps your grade.
- Non-electronic translation dictionaries and calculators without a full set of alphabet keys are permitted.

Questions	Marks	
1. Basic Java	[31]	
2. Loops with the DrawingCanvas	[15]	
3. Objects and Fields	[15]	
4. Debugging	[21]	
5. Event Driven Input	[8]	
	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 [31 marks]

(a) [3 marks] What will the following fragment of Java print out?

```
double width = 5.0;
double length = 8.0;
System.out.println("A: " + width + length);
System.out.println("B: " + (width + length));
System.out.println("C: " + width + "==" + length);
if (width == length){
  System.out.println("D: same");
}
else {
  System.out.println("D: different");
}
System.out.println("E: " + (width == length));

width = length + 2;
length = width / 2;
double ans = width * length;
System.out.printf("F: %4.2f * %4.2f -> %4.2f\n", width, length, ans);
```

```
A: 5.08.0
B: 13.0
C: 5.0==8.0
D: different
E: false
F: 10.00 * 5.00 -> 50.00
```

(Question 1 continued on next page)

(Question 1 continued)

(b) [4 marks] Consider the following printAns method.
[Marked out of 6!]

```
public void printAns(String arg){
    String ans = "Ans";
    if ( arg.length() > 4 && arg.length()<8 ) {
        ans = ans + " length";
    }
    if ( arg.equals("John") || arg.equals("Peter") ){
        ans = ans + " name";
      }
    else if ( arg.startsWith("J") ) {
        ans = ans + " just";
    }
    ans = ans + " done";
    System.out.println(ans);
}
```

(i) [2 marks] What will be printed if printAns("James") is called?

```
Ans length just done
```

(ii) [2 marks] What will be printed if printAns("John") is called?

```
Ans name done
```

(iii) [2 marks] What will be printed if printAns("Peter") is called?

```
Ans length name done
```

(Question 1 continued on next page)

(Question 1 continued)

(c) [4 marks] What will the following fragment of Java print?

```
int i = 0;
System.out.print("num: ");
while ( i < 20) {
  System.out.print(i + ", ");
  i = i + 4;
}
```

```
num:  0, 4, 8, 12, 16
```

(d) [4 marks] Write a fragment of Java that will print out all the even integers from 1000 down to 100 (including 100), one number per line.

```
int k = 1000;
while ( k>=100 ){
    System.out.println(k);
    k = k-2;
}
```

(e) [4 marks] Write a method named square which has one *double* parameter and returns a value of type *double*. square should return the square of its parameter (the number multiplied by itself). For example, square(2.5) should return the value 6.25.

```
public double square (double n){
    return n * n;




}
```

(Question 1 continued on next page)

(f) [6 marks]

The following repeater method has one int parameter n. It should ask the user (in the terminal window) to enter a line of text. It should then print out the line they entered n times, numbering each of the lines. For example, if it were called with the argument 4, and the user entered the text "`Write a new program`", it should then print out:

```
1 Write a new program
2 Write a new program
3 Write a new program
4 Write a new program
```

Complete the repeater method.

```
public void repeater(int n){
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter line: ");
    String line = scan.nextLine();
    int count = 1;
    while (count <= n){
        System.out.println(n + ": "+ line);
        count = count + 1;
    }
//OR
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter line: ");
    String line = scan.nextLine();
    int count = 0;
    while (count < n){
        count = count + 1;
        System.out.println(n + ": "+ line);
    }









}
```

(Question 1 continued on next page)

(Question 1 continued)

(g) [6 marks] Suppose the file called data.txt contains the text:

```
3 words have 9
7 are 2 4 1
8 seem like letters.
```

What will the following fragment of Java print?

```
try{
    Scanner sc = new Scanner(new File("data.txt"));
    int num=0;
    while (sc.hasNext()){
        if (sc.hasNextInt()){
            num = sc.nextInt() - num;
        }
        String token = sc.next();
        System.out.print(token+" ");
    }
    System.out.println();
    System.out.println("num = " + num);
    sc.close();
} catch(IOException e){System.out.println("Error while scanning"+e);}
```

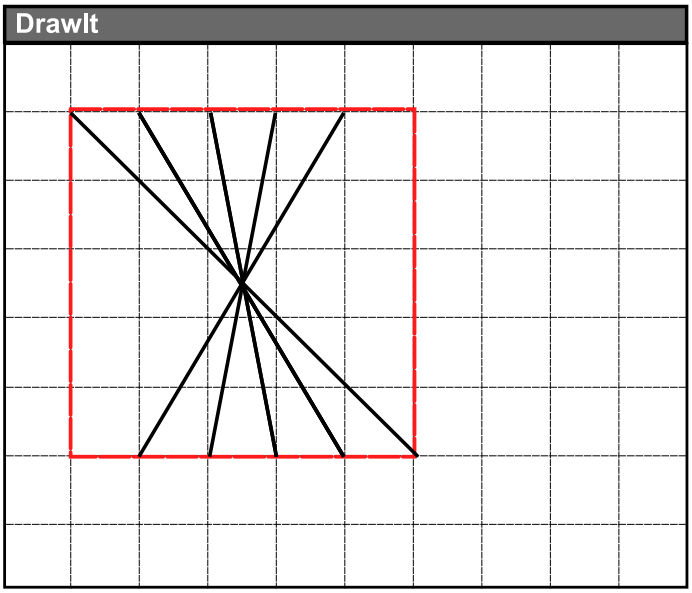
```
words have 7 are 4 8 seem like letters.
num = 5
```

Question 2. Loops with the DrawingCanvas [15 marks]

(a) [5 marks] Sketch what the following drawIt method will draw on the DrawingCanvas below. (Assume the grid lines on the canvas are spaced 20 units apart.)

```
public void drawIt(DrawingCanvas canvas){
    int a = 20;
    int b = 20;
    int c = 100;
    canvas.setColor(Color.red);
    canvas.drawRect(a, b, c, c);

    canvas.setColor(Color.black);
    int x = 0;
    while (x < c){
        canvas.drawLine(a+x, b, a+c−x, b+c);
        x = x + 20;
    }
}
```

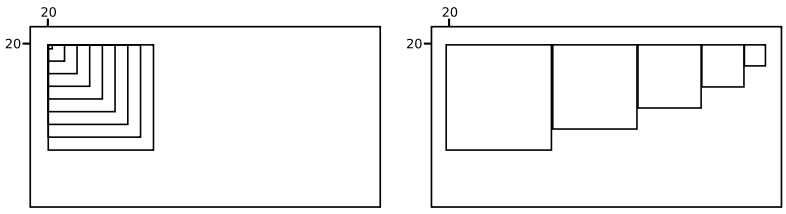


(Question 2 continued on next page)

(Question 2 continued)

(b) [10 marks] Complete the following drawSquares method so that it draws a set of squares of decreasing size. The first parameter of the method (size) specifies the size of the first square. The second parameter (diff) specifies the difference in size of successive squares (each square should be diff units smaller than the previous square). The third parameter specifies whether each square should be drawn inside the previous one or to its right. The fourth parameter is the DrawingCanvas to draw on. The first square should be placed at the position (20,20).

For example, drawSquares(100, 12, "inside", canvas) would draw the squares shown in the window on the left; drawSquares(100, 20, "right", canvas) would draw the squares shown in the window on the right.



There is documentation on the DrawingCanvas class at the end of the test paper.

```
public void drawSquares(int size, int diff, String how, DrawingCanvas canvas){
  int x = 20;
  int y = 20;
  while ( size > 0 ){
    canvas.drawRect(x, y, size, size);
    if ( how.equals("right") ){
      x = x + size;
    }
    size = size - diff;
  }




}
```

Question 3. Objects and Fields [15 marks]

This question concerns a program for a simple game in which a number of "balloons" (drawn as coloured circles) are placed at random positions on the screen. The player has to blow up the balloons, in small steps, in a way that maximises the sum of the areas of all the balloons. However, if the player blows up a balloon too large, and it touches another ballon, then the game is over.

One part of the program is the Balloon class that represents the individual balloons. You are to complete the Balloon class on the facing page.

- Each Balloon must have fields to store the DrawingCanvas it is drawn on, its position (the center of the balloon), and its size. Each balloon starts with a radius of 5 pixels.
- The constructor for the balloon should have three parameters: the DrawingCanvas, and its position (two ints). The constructor should initialise the fields and draw the balloon.
- The draw method should draw the balloon as a red coloured circle.
- The expand method should make the balloon larger by adding its int parameter to the current radius, and then drawing the Balloon again at the same position.
- The touches method has a parameter which will be another Balloon. It should return true if this Balloon is touching the other Balloon (ie, if the centers are closer than the sum of the two radiuses), and false if they are not touching.

Some parts of a possible program using the Balloon class are given below, to show how the Ballon class could be used.

```
public class BalloonGame {

        :
// make several new balloons
Balloon b1 = new Balloon(this.canvas, 57, 320);
Balloon b2 = new Balloon(this.canvas, 290, 144);
Balloon b3 = new Balloon(this.canvas, 157, 233);
        :
//blow up a balloon
b1.expand(5);
b1.expand(3);
        :
// check if touching
if ( b1.touches(b2) || b1.touches(b3) ){
   System.out.println("Game over");
   this.canvas.clear();
}
        :
```

(Question 3 continued on next page)

(Question 3 continued)

```
public class Balloon{
  // Fields
  private DrawingCanvas canvas;
  private int radius = 5;
  private int centerX, centerY;


  // Constructor: initialises fields and draws the Balloon
  public Balloon(DrawingCanvas c, int x, int y){
    this.centerX = x;
    this.centerY = y;
    this.canvas = c;
    this.draw();


  }
  // draw the balloon at its current size and position as a red circle.
  public void draw(){
    this.canvas.setColor(Color.red);
    this.canvas.fillOval (this.centerX-this.radius, this.centerY-this.radius,
                          this.radius*2, this.radius*2);

  }
  // Increase the radius by the step, and draw the Balloon again
  public void expand(int step){
    this.radius = this.radius + step;
    this.draw();


  }
  // Returns true if this Balloon is touching the other Balloon, and
  // false if it is not touching.
  public boolean touches(Balloon other){
     int dx = other.centerX - this.centerX;
     int dy = other.centerY - this.centerY;
     int dist = other.radius + this.radius;
    return ((dx*dx + dy*dy) < dist*dist);



  }
}
```

Question 4. Debugging [21 marks]

The projectCol method is intended to read a file containing a table of words, and print one line containing the words in just one of the columns of the table, separated by commas. The first argument specifies which column (which must be between 1 and the number of columns); the second argument is the name of the file.

For example, if the file `test.txt` contains the text on the left, then projectCol(3, "test.txt") should print out the values in the 3rd column, as on the right:

contents of `test.txt`

```
a b c d e
f g h i j
k l m n o
p q r s t
u v w x y
```

projectCol(3, "test.txt") ⇒

```
c, h, m, r, w
```

The following version of projectCol has errors:

```
public void projectCol(int n, String fname){     // This version of projectCol has errors
  try {
    Scanner scan = new Scanner(new File(fname));
    while (scan.hasNext()) {
      int col = 1;
      while (col <= n){                  // Ans (b) counts too far
        String junk = scan.next();
        col = col+1;
      }
      String word = scan.next();         // Ans (b) need to also read rest of line.
      System.out.println(word+",");      // Ans (b) print, not println.
    }                                    // Ans (b) should be no comma after last word.
    scan.close();
  } catch(Exception e){System.out.println("Error while scanning"+e);}
}
```

(a) [7 marks] If you called this version of projectCol with the statement
(projectCol(3, "test.txt");
what would it print out?

```
d,
h,
l,
p,
t,
x,
Error
```

(Question 4 continued on next page)

(Question 4 continued)

(b) [5 marks] On the code for projectCol in (a), circle and briefly explain at least four errors in the code.

Identify the errors on the code on the facing page!

(c) [9 marks] Write a correct version of the projectCol method.

```
public void projectCol(int n, String fname){
  try {
    Scanner scan = new Scanner(new File(fname));

    while (scan.hasNext()) {
      int col = 1;
      while (col < n){
        String junk = scan.next();
        col = col+1;
      }
      String word = scan.next();
      String rest = scan.nextLine();
      System.out.print(word);
      if (scan.hasNext()) System.out.print(",");
    }
    System.out.println ();
//or
    String word = null;
    while (scan.hasNext()){
      if (word!=null){ System.out.print(","); }
      int col = 0;
      while (col < n){
        word = scan.next();
        col++;
      }
      String rest = scan.nextLine();
      System.out.print(word);
    }
    System.out.println ();

    scan.close();
  } catch(Exception e){System.out.println("Error while scanning"+e);}
}
```

```
while (scan.hasNext()) {
  Scanner lineSc = new Scanner(scan.nextLine());
  String word = lineSc.next();
  int col = 2;
  while (col <= n){
    word = lineSc.next();
    col++;
  }
  System.out.print(word);
  if (scan.hasNext()){ System.out.print(","); }
}
System.out.println ();
```

Question 5. Event Driven Input [8 marks]

Consider the version of the assignment 5 ScreenSaver on the facing page that has a GUI with two buttons. The code for the MovingShape class is provided at the end of the test paper.

(a) [3 marks] Suppose a user runs the program by calling the main method, and then clicks the "Reset" button. The program will then display a red circle on the screen. Explain how the Java Virtual Machine makes this happen, particularly explaining why ScreenSaver **implements** ActionListener, the role of the listener, and the steps that resulted the MovingShape constructor being called.

When main is called, it will call the ScreenSaver constructor, which will set up the user interface, including the two buttons. Both buttons will have the ScreenSaver object as a listener. ScreenSaver must implement ActionListener for this to be allowed. When the user clicks on the "Reset" button, the JVM will look up the listener of the Reset button, and call its actionPerformed method. This will then call the reset method, which calls the MovingShape constructor, which draws itself on the canvas.

(b) [3 marks] What would happen if the user clicked the "Run" button first, and how could the bug be fixed?

The program will crash with an error because the currentShape field will contain null, and the run method will therefore attempt to call the move method on null. It could be fixed by initialising the currentShape field in the constructor, eg, by having the constructor call reset().

(c) [2 marks] Why will the "Reset" button have no effect after the "Run" button has been clicked?

Because the Run button will call the run method [in the GUI thread], which goes into an infinite loop, so the JVM will never get round to responding to the "Reset" button.

(Question 5 continued)

```
/* Code for COMP 102 test 2008 T1 */
import comp100.*;
import javax.swing.*;
import java.awt.event.*;
import java.awt.BorderLayout;

public class ScreenSaver implements ActionListener{
    private JFrame frame = new JFrame("ScreenSaver");
    private DrawingCanvas canvas = new DrawingCanvas();
    private MovingShape currentShape;

    public ScreenSaver(){
        this.frame.setSize(600, 650);
        this.frame.getContentPane().add(this.canvas, BorderLayout.CENTER);
        JPanel panel = new JPanel();
        this.frame.getContentPane().add(panel, BorderLayout.NORTH);
        JButton but1 = new JButton("Reset"); panel.add(but1);
        but1.addActionListener(this);
        JButton but2 = new JButton("Run");   panel.add(but2);
        but2.addActionListener(this);
        this.frame.setVisible(true);
    }
    public void actionPerformed(ActionEvent e){
        String cmd = e.getActionCommand();
        if (cmd.equals("Reset") ){ this.reset(); }
        else if (cmd.equals("Run") ){this.run();}
    }
    public void reset(){
        this.canvas.clear();
        this.currentShape = new MovingShape(100, 100, 600, this.canvas);
    }
    public void run(){
        int count = 50;
        double stepX, stepY;
        while (true){
            if (count == 50){
                stepX = (Math.random()-0.5)*4;
                stepY = (Math.random()-0.5)*4;
                count = 0;}
            this.currentShape.move(stepX, stepY);
            count++;
            canvas.display();
            try{Thread.sleep(20);}catch(Exception e){}
        }
    }
    public static void main(String[] args){
        ScreenSaver S = new ScreenSaver();
    }
}
```

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.

****