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

ID Number:.....

COMP102: Test 2

Model Solutions

3 Sep, 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	[12]	
2. if	[14]	
4. while	[26]	
4. Objects and Fields	[20]	
5. Files and Loops	[18]	
	TOTAL:	

Question 1. Basic Java [12 marks]

For each of the following terms or actions, find a corresponding element of the program below, draw a box around the element and label it with its number. The first one is done as an example.

1. A class name
2. A data field
3. The name of a constructor
4. A Boolean expression
5. An argument
6. A parameter
7. A cast (Casting is a way of conversion in Java.)
8. Call a method on the same object
9. Call a method on another object
10. Call a static method
11. Call a method that returns a value
12. Update/access a data field
13. Create an object

```
public class MShape{
    private DrawingCanvas canvas;
    private double x;
    private double y;
    private int size = 50;
    private int limit ;

    public MShape(double x, double y, int limit , DrawingCanvas c){
        this.x = x;
        this.y = y;
        this. limit = limit ;
        this.canvas = c;
        this.draw();
    }

    public void move(double stepX, double stepY){
        this.erase();
        this.x = this.x + stepX;
        this.y = this.y + stepY;

        if (this.x < 0) this.x = 0;
        if (this.x > (this. limit - this.size)) this.x = this. limit - this.size;
        if (this.y < 0) this.y = 0;
        if (this.y > (this. limit - this.size - 20)) this.y = this. limit - this.size - 20;

        this.draw();
    }


   public void erase(){
        this.canvas.clearRect((int)this.x, (int)this.y, this.size+1, this.size+1, false);
    }

    public void draw(){
        Color c = new Color((int)(Math.random()*255), 0, (int)(Math.random()*255));
        this.canvas.setColor(c);
        this.canvas. fillOval ((int)this.x, (int)this.y, this.size, this.size,false);
    }
}
```

Question 2. if [14 marks]

(a) [6 marks] Consider the following printMessage method.

```
public void printMessage(int arg){
    String ans = "Ans";
    if ( arg > 4 && arg < 8 ) {
        ans = ans + " small";
    }
    if ( arg == 4 || arg == 6 ) {
        ans = ans + " big";
      }
    else if ( arg < 4 ) {
        ans = ans + " just";
    }
    ans = ans + " done";
    System.out.println(ans);
}
```

(i) [2 marks] What will be printed if printMessage(10) is called?

```
Ans done
```

(ii) [2 marks] What will be printed if printMessge(4) is called?

```
Ans big done
```

(iii) [2 marks] What will be printed if printMeaasge(6) is called?

```
Ans small big done
```

(b) [8 marks] Write a method named myFunction which has one *double* parameter and returns a value of type *double*. myFunction is defined as follows:

$$myFunction(x) = \begin{cases} 2 * x + 1 & \text{if } x > 0 \\ 0.0 & \text{if } x = 0 \\ 2 * x - 1 & \text{otherwise} \end{cases}$$

For example, myFunction(2.5) should return the value 6.0.

```
public
        double myFunction(double x) {
    double d;
    if (x >0)
        d = 2 * x +1;
    else if (x == 0)
        d = 0;
    else
        d = 2 * x -1;
    return d;
 }


}
```

Question 3. Loops [26 marks]

(a) [6 marks] What will the following fragment of Java print?

```
int i = 1;
System.out.print("num: ");
while ( i < 15) {
  System.out.print(i + ", ");
  i = i + 3;
}
```

```
num:  1, 4, 7, 10, 13,
```

(b) [10 marks] Write a method named printNum which has one *int* parameter n and prints out all the integers from 1 to n, five numbers per line. For example, printNum(12) should print:

1 2 3 4 5
6 7 8 9 10
11 12

```
public void printNum(int n) {
    int i =1;
    int count = 1;
    while(i <=n) {

        if (count > 5) {
            System.out.println ();
            count =1;
        }
        count++;
        System.out.print(i + " ");
        i++;
    }

}
```

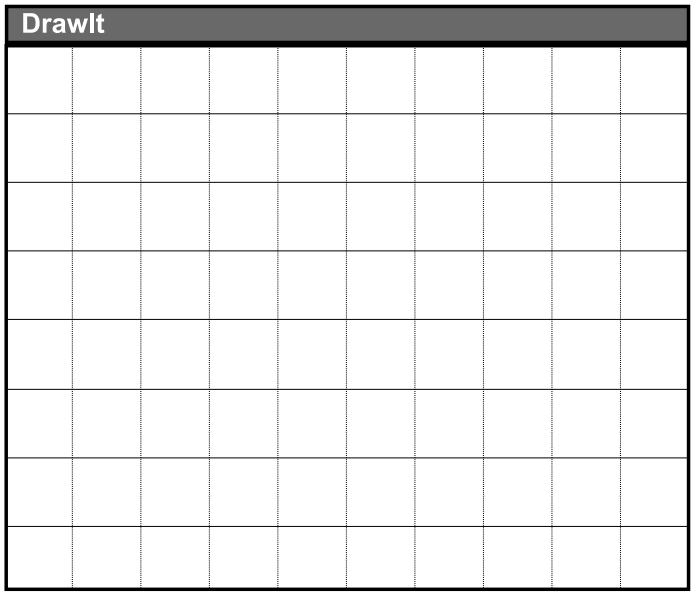
(c) [10 marks] Sketch what the following drawIt method will draw on the DrawingCanvas below. Assume the grid lines on the canvas are spaced 20 units apart.

```
public void drawIt(DrawingCanvas canvas) {
        int y = 20;
        for( int i = 0; i < 4; i++){
          this.drawPattern(canvas, 20, y, 100);
          y = y + 20;
        }
}

public void drawPattern(DrawingCanvas canvas, int a, int b, int c){

      canvas.drawRect(a, b, c, 20);

      while (a <= c){
          canvas.drawOval(a, b, 20, 20);
          a = a + 20;
      }
 }
```



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 4. Objects and Fields [20 marks]

(a) [6 marks] What will the following program print?

```
public class testObjects {
  public static void main(String[] args) {
    Item a = new Item(2, 6);
    Item b = new Item(3, 7);
    Item c = new Item(5, 4);

    System.out.println(a.f (2));
    System.out.println(c.g ());
    System.out.println(a.g() + b.f (1));
   }
  }


class Item {
  private int x, y;
  public Item(int xx, int yy) {
    this.x = xx;
    this.y = yy;
  }
  public int f( int m) {
      return this.x + m;
  }
  public int g() {
    return this.x * y;
  }
}
```

```
4 | 20 | 16
```

(b) [14 marks]

This question concerns a program for a simple game in which a number of "stones" (drawn as coloured circles) are placed on the screen and the stones can drop to the ground in small steps.

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

- Each stone object must have fields to store the DrawingCanvas it is drawn on, its position (the center of the stone), and its size.
- The Stone constructor should have four parameters: the DrawingCanvas, its position and its size (three *int*s). The constructor should initialise the fields and draw the stone.
- The draw method should draw the stone as a red coloured circle.
- The drop method should make the stone drop some distance towards the ground by updating its position using its *int* parameter, erasing the old stone, and drawing the stone at the new position.

Some parts of the program using the stone class are given below, to show how the Stone class could be used.

```
public class stoneGame {
  private DrawingCanvas canvas = new DrawingCanvas();

  public void testStoneGame() {
    JFrame frame = new JFrame("ScreenSaver2");
    frame.setSize(600, 600);

            :
    // make several new stones
    Stone b1 = new Stone(this.canvas, 57, 320, 10);
    Stone b2 = new Stone(this.canvas, 290, 144, 50);
    Stone b3 = new Stone(this.canvas, 157, 233, 20);
          :
    //drop a stone
    b1.drop(80);
    b3.drop(70);
          :
```

(Question 4 continued on next page)

(Question 4 continued)

```
public class Stone{
  // Fields
   private DrawingCanvas canvas;
   private double x;
   private double y;
   private int size;
  // Constructor: initialises fields and draws the stone
  public Stone(DrawingCanvas c, int x, int y, int s){



        this.x = x - s/2;
        this.y = y - s/2;
        this.canvas = c;
        this.size = s;
        this.draw();

  }
  // draw the stone at its size and current position as a red circle .
  public void draw(){

        this.canvas.setColor(Color.RED);
        this.canvas.fillOval ((int)this.x, (int)this.y, this.size, this.size,false);


  }
  // drop the stone slowly
  public void drop(int n){
        this.canvas.clearRect((int)this.x, (int)this.y, this.size+1, this.size+1, false);
        this.y = this.y + n;
        this.draw();









  }
```

Question 5. Files [18 marks]

Suppose the file lectureTimes.txt contains data about lectures during a week. Here is an example file:

```
Monday 1000 1210 1510
Tuesday 1210
Wednesday 900 1410
Thursday
Fri 1210
```

The first line says that there are three lectures on Monday and they start at 10:00, 12:10 and 15:10 respectively. In general,

- The file has exactly five lines and each line starts with the days of the week.
- The start times are given in 24-hour time and each lecture lasts 50 minutes. The lectures start on the hour in the mornings and 10 minutes after in the afternoons.
- The number of lectures on each day varies and there are days that have no lectures.

(a) [6 marks] Using the example file given above, what will the following method print to System.out?

```
public void testFiles () {
    try{
        Scanner fileScan = new Scanner(new File("lectureTimes.txt"));
        while (fileScan.hasNext()) {
          String day = fileScan.next();
          int time = fileScan.nextInt();
          System.out.println(day + " - " + time);
        }
        fileScan.close();
    }
    catch(Exception e){System.out.println("File reading error.");}
}
```

```
Monday - 1000
1210 - 1510
Tuesday - 1210
Wednesday - 900
File reading error.
```

(Question 5 continued on next page)

(Question 5 continued)

(b) [12 marks] Write a method that reads data from a file in the format described above and calculates the total lecture hours in a week. Please note each lecture lasts 50 minutes. On the previous example file, it should return 5.833.

```
public double lectureHours() {

    double d = 0.0;
    try {
        Scanner f = new Scanner(new File("lectureTimes.txt"));
        while(f.hasNext()) {
            String day = f.next();

            while(f.hasNextInt()){
                f.nextInt();
                d = d + 50.0/60.0;
            }
        }
    }
    catch(Exception e) {System.out.println("File reading error.");}
    return d;









}
```

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.

****

(You may detach this page)

Brief and partial documentation of some classes and methods

```
PrintStream:                                    // Note, System.out is a PrintStream object
 public PrintStream (File f);                   // Constructor, for printing to a file
 public void close();                           // Close the file (if it is wrapping a File object)
 public void print (String s);                  // Prints s with no newline
 public void print (int i);                     // Prints i with no newline
 public void print (double d);                  // Prints d with no newline
 public void println ();                        // Prints a newline
 public void println (String s);                // Prints s followed by newline
 public void println (int i);                   // Prints i followed by newline
 public void println (double d);                // Prints d followed by newline
 public void printf (String format, ...);       // Prints the format string, inserting the remaining
                                                // arguments at the %'s in the format string:
                                                //   %s for Strings.
                                                //   %d for ints, (%3d: use at least 3 characters),
                                                //   %.2f for 2dp doubles,
                                                //   (%6.2f: at least 6 characters and 2dp),
                                                // Use \n for newline

Scanner
   public Scanner (InputStream i);              // Constructor. Note: System.in is an InputStream
   public Scanner (File f);                     // Constructor, for reading from a file
   public Scanner (String s);                   // Constructor, for reading from a string
   public boolean hasNext();                    // Returns true if there is more to read
   public boolean hasNextInt();                 // Returns true if the next token is an integer
   public boolean hasNextDouble();              // Returns true if the next token is a number
   public String next();                        // Returns the next token (chars up to a space/line)
   public String nextLine();                    // Returns the next line
   public int nextInt();                        // Returns the integer value of the next token
                                                //  (throws exception if next token is not an integer)
   public double nextDouble();                  // Returns the double value of the next token
                                                //  (throws exception if next token is not a number)
   public void close();                         // Closes the file (if it is wrapping a File object)

File
   public File (String fname);                  // Constructor. Creates a File object attached to the
                                                //  file with the name fname

Integer
   public static final int MAX_VALUE;           // The largest possible int: 2^(31−1)
   public static final int MIN_VALUE;           // The smallest possible int: −2^(31)

Double
   public static final double MAX_VALUE;        // The largest possible double: just under 2^(10
   public static final double MIN_VALUE;        // The smallest possible positive nonzero double
   public static final double POSITIVE_INFINITY; // positive infinity (greater than any number)
```

```
public static final double NEGATIVE_INFINITY; // negative infinity (less than any number)
public static final double NaN;               // The Double that is Not a Number (''undefined'
```

String

```
public int length();                          // Returns the length (number of characters) of the
public boolean equals(String s);              // String has same characters as s
public boolean equalsIgnoreCase(String s);    // String has same characters as s, ignoring their
public boolean startsWith(String s);          // First part of string matches s
public boolean contains(String s);            // s matches some part of the string
public String substring(int j, int k)         // Returns substring from index j to index k−1
public int indexOf(String s);                 // Returns −1 if it does not contain s anywhere
                                              // otherwise, returns the index of where s first m
```

Math

```
public static double sqrt(double x);          // Returns the square root of x
public static double min(double x, double y); // Returns the smaller of x and y
public static double max(double x, double y); // Returns the larger of x and y
public static double abs(double x);           // Returns the absolute value of x
public static int min(int x, int y);          // Returns the smaller of x and y
public static int max(int x, int y);          // Returns the larger of x and y
public static int abs(int x);                 // Returns the absolute value of x
```

DrawingCanvas

```
public void clear();                          // Clears the drawing canvas
public void setColor(Color c);                // Change the colour for later commands
public void drawLine(int x, int y, int u, int v);  // Draws line from cd{(x, y) to cd{(u, v)
public void drawRect(int x, int y, int wd, int ht); // Draws outline of rectangle
public void fillRect(int x, int y, int wd, int ht); // Draws solid rectangle
public void clearRect(int x, int y, int wd, int ht); // Draws clear rectangle
public void drawOval(int x, int y, int wd, int ht); // Draws outline of oval
public void fillOval(int x, int y, int wd, int ht); // Draws solid oval
```

Color

```
public Color(int red, int green, int blue);   // Make a colour; arguments must be 0..255
public Color(float red, float green, float blue); // Make a colour; arguments must be 0..1.0
Color.gray, Color.blue, Color.red,            // Some of the predefined colours
Color.green, Color.black, Color.white
```