

Family Name: Other Names:

Student ID: Signature

COMP 102: Test 2

2017, December ?? ** WITH SOLUTIONS **

Instructions

- Time allowed: **90 minutes**
- Attempt **all** the questions. There are 50 marks in total.
- Write your answers 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 increased to your exam mark if that is higher.)
- You may use dictionaries.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions

Marks

1. Understanding while	[6]	<input type="text"/>
2. Writing with while	[8]	<input type="text"/>
3. Defining Classes	[10]	<input type="text"/>
4. Event Driven Input	[10]	<input type="text"/>
4. Files	[16]	<input type="text"/>
	TOTAL:	<input type="text"/>

Question 1. Understanding while**[6 marks]**

Consider the following printOut method that has one parameter.

```

public void printOut(double num) {
    double x = 0;
    while (x <= num) {
        Ul.println(x);
        x = x + 5;
    }
    Ul.println("x: " + x);
    Ul.println("End");
}

```

x:

(a) **[3 marks]** What will be printed if printOut(12) is called?**Hint:** Show your working using the box for x.

Your answer:

```

0.0
5.0
10.0
x: 15.0
End

```

(b) **[3 marks]** What will be printed if printOut(5) is called?

Your answer:

```

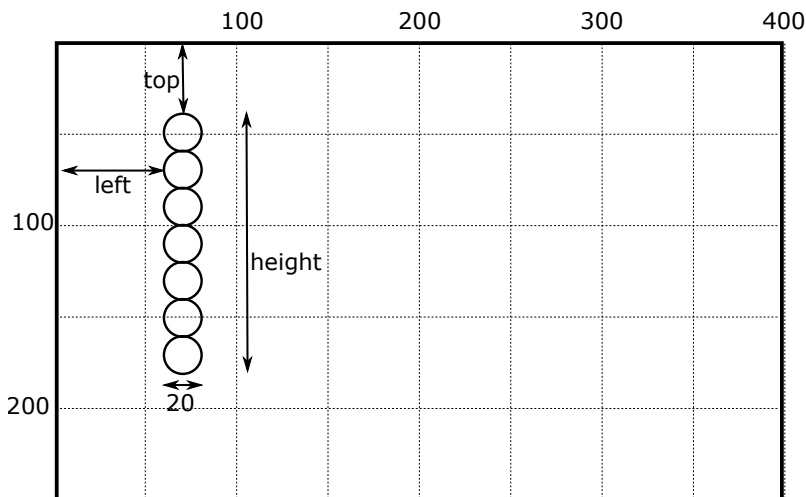
0.0
5.0
x: 10.0
End

```

Question 2. Writing with while**[8 marks]**

Complete the following drawCircles method so that it uses a loop to draw a tower of circles, as shown below. You may use a **for** loop or a **while** loop.

- left is the left side of all the circles.
- top is the top of the top circle.
- height is the height of the tower of circles.
- The size of each circle is 20.



For example, the pattern above was generated by calling drawCircles(60, 40, 140).

```
public void drawCircles(double left, double top, double height) {
    for (double y = top; y <= top + height - 20; y = y + 20) {
        UI.drawOval(left, y, 20, 20);
    }
```

OR

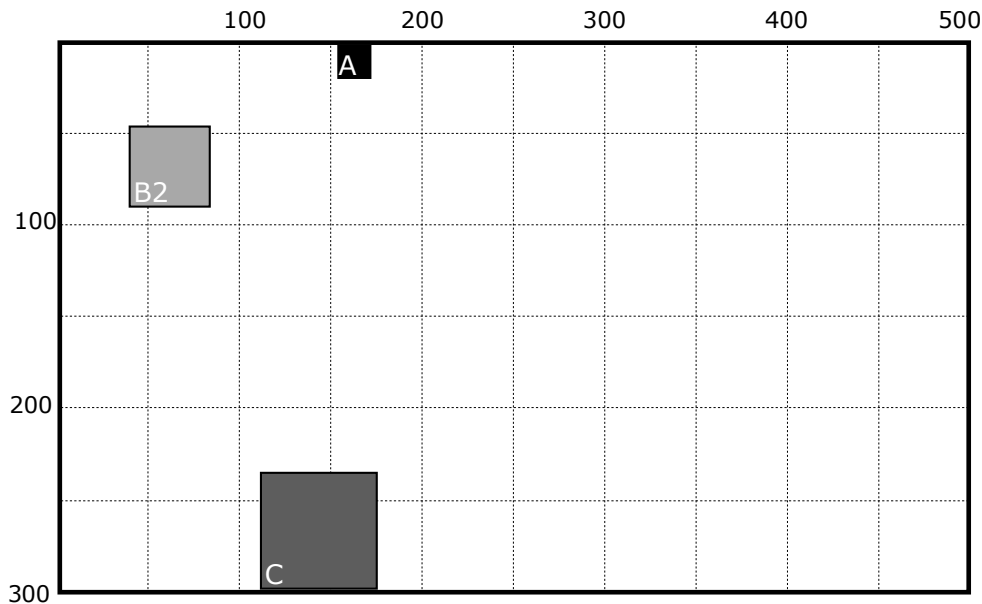
```
double y = top;
while (y <= top + height - 20) {
    UI.drawOval(left, y, 20, 20);
    y = y + 20;
}
```

}

Question 3. Defining Classes**[10 marks]**

Complete part of a program for a simple video game. The game has Block objects representing colored squares that fall down from the top of the screen.

Here is an example showing three Blocks. "A" is at the top, "B2" has fallen 90 units, "C" has fallen 235 units, and is now at the bottom:



Complete the Block class on the facing page. You must include:

- Fields to store the relevant information about a Block:
 - the position of the block, (*already completed for you*)
 - the size of the block,
 - the color of the block, and
 - name of the block.
- A constructor with three parameters: the color of the block, the name on the block, and the size. A new block always starts at the top of the screen, at a random x-position between 0 and 400.
- The fall method, which will make the block move down by some distance.
- The atBottom method which will return true if the block is at the bottom of the screen (300 units below the top), and false otherwise. For example, atBottom should return true for block C above.
- The draw method, which draws the Block at its current position (as a colored square, with the name in the bottom left corner).

```

public class Block{
    // Fields
    private double x = Math.random()*400;
    private double y;
    private double size ;
    private Color col;
    private String name;

    /** Creates a new Block object with the specified name, color , and size . */
    public Block (String name, Color color , double size){
        this.name = name;
        this.col = color;
        this.size = size;

    }
    /** Makes the Block fall by the distance . Does not redraw the Block */
    public void fall (double distance) {
        this.y = this.y + distance;

    }
    /** Returns true if the bottom of the Block is at or below 300
     * Otherwise, returns false . */
    public boolean atBottom() {
        return (this.y+this.size >= 300);
//OR
        if (this.y+this.size >= 300) {
            return true;
        }
        else {
            return false ;
        }

    }
    /** Draws a solid coloured square with the name in white at the bottom left . */
    public void draw(){
        UI.setColor (this.col );
        UI.fillRect (this.x, this.y, this.size , this.size );
        UI.setColor (Color.white);
        UI.drawString(this.name, this.x, this.y+this.size );

    }
}

```

Question 4. Event Driven Input

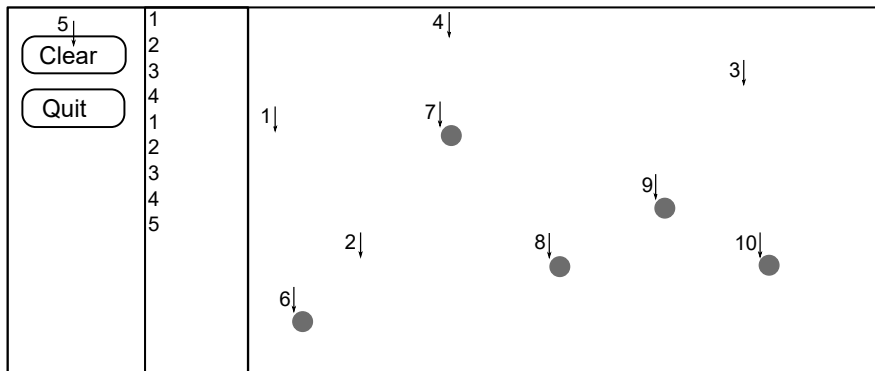
[10 marks]

Complete the DotMaker program on the facing page so that it allows the user to draw dots (small circles) on the drawing pane using the mouse.

- Every time the user releases the mouse, the program should draw a small circle at that position and print out the total number of dots in the text pane.
- The "Clear" button should clear the graphics pane and reset the count.
- The "Quit" button should quit the program.

For example, the diagram shows what the program should show if the user

- clicked at positions 1↓ to 4↓
- pressed the clear button, 5↓
- clicked at positions 7↓ to 10↓



Hints:

- The size of the dots should be 4 pixels.
- Your program should have a field to remember the count.

(Question 4 continued)

```
public class DotMaker {
    // Fields
    private int count;

    /** Constructor: sets up the interface . */
    public DotMaker(){
        Ul.addMouseListener( this :: doMouse);
        Ul.addButton("Clear", this::doClear);

        Ul.addButton("Quit", Ul::quit);
    }

    /** Respond to the Clear button */
    public void doClear(){
        Ul.clearGraphics ();
        this.count = 0;

    }

    /** Respond to the Mouse */
    public void doMouse(String action, double x, double y){
        if (action.equals("released")){
            Ul.fillOval (x, y, 4, 4);
            this.count = this.count + 1;
            Ul.println (this.count);
        }

    }

}
```

Question 5. Files**[16 marks]**

This question is about files which contain information about a players in a basketball team. Each line has a player's first name, second name, age, and position on the team.

For example:

Pan	Siwen	18	point-guard
Jiang	Minping	22	small-forward
Zhou	Xiaotian	20	center
Zhang	Qi	19	point-guard
Bai	Zichen	19	small-forward
Chang	Yixuan	21	shooting-guard
Sun	Baohua	24	power-forward
Zhao	Ying	18	center
Li	Haochen	20	point-guard
Zhang	Mingxuan	18	shooting-guard
Liu	Yijuan	23	small-forward

(a) [4 marks] Consider the following printData method.

```
public void printSomePlayers(){
    try{
        Scanner scan = new Scanner (new File("BasketballTeam.txt"));
        String line = scan.nextLine();
        while (scan.hasNext()){
            if (line.endsWith("forward")){
                Ul.println ( line );
            }
            line = scan.nextLine();
        }
        Ul.println ("done");
    } catch(IOException e){Ul.println ("Fail: " + e);}
}
```

What will be printed if printSomePlayers is called, assuming "BasketballTeam.txt" is the example file shown above?

Hint: Notice that there are two instructions that read the next line of text from the file.

```
Jiang Minping 22 small-forward
Bai Zichen 19 small-forward
Sun Baohua 24 power-forward
done
```


(b) [6 marks] Complete the following printYoungPlayers method.

Its parameters are

- the name of a file of basketball players (a String)
- a maximum age (a number)

The printYoungPlayers method should read through the file and print the name of every player in the file whose age is less than or equal to the maximum age.

For example, using the example file above, printLimitedItems("BasketballTeam.txt", 19) should print:

```
Pan Siwen
Zhang Qi
Bai Zichen
Zhao Ying
Zhang Mingxuan
```

Hint: Each time round the loop, read all four tokens on the line.

```
public void printYoungPlayers( String fileName, int maxAge){
    try{
        Scanner scan = new Scanner (new File(fileName));
        while(scan.hasNext()){
            String name1 = scan.next();
            String name2 = scan.next();
            int age = scan.nextInt ();
            String position = scan.next();
            if (age <= maxAge){
                UI.println (name1 + " " + name2);
            }
        }
        scan.close ();
    }
} catch(IOException e){UI.println ("Fail: " + e);}
}
```

(c) [6 marks] There are two “forward” positions in basketball: “power-forward” and “small-forward”.

Complete the following countForwards method.

- Its parameter is the name of a file of basketball players.
- The countForwards method should count and print out the number of players for each for the two “forward” positions in a file

For example, using the example file above, countPositions(“BasketballTeam.txt”) should print:

```
power-forward 1
small-forward 3
```

```
public void countPositions( String fileName){
    try{
        Scanner scan = new Scanner (new File(fileName));
        int countPower = 0;
        int countSmall = 0;
        while(scan.hasNext()){
            String name1 = scan.next();
            String name2 = scan.next();
            int age = scan.nextInt ();
            String position = scan.next ();
            if ( position .equals("power-forward")) {
                countPower++;
            }
            if ( position .equals("small-forward")) {
                countSmall++;
            }
        }
        Ul. println ("power-forward: " + countPower);
        Ul. println ("small-forward: " + countSmall);
        scan. close ();

    } catch(IOException e){Ul. println ("Fail: " + e);}
}
```
