

Family Name:.....

Other Names: .....

ID Number: .....

Signature .....

## COMP 102: Test

14 August, 2014

### 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 alpha-bet keys.
- You may write notes and working on this paper, but make sure your answers are clear.

### Questions

### Marks

1. Understanding programs with input and output

[5]

2. Writing programs with input and output

[5]

3. Understanding variables, if, while

[8]

4. Understanding method calls

[7]

5. Defining and calling methods

[10]

6. Writing methods that use objects

[10]

TOTAL:

### Question 1. Understanding programs with input and output

[5 marks]

What will the following program do if the drawlt method is called, and the user types 200 when asked for the width and 100 for the height?

Write and sketch your answer on the diagram below, showing what will be in the text pane and the graphics pane of the UI window.

```
import ecs100.*;
public class Drawer{
    public static final double centerX=200;
    public static final double centerY=200;

    public void drawlt(){
        double width = UI.askDouble("enter the width: ");
        double height = UI.askDouble("enter the height: ");
        double area = height * width;
        UI.println ("The area is " + area);
        double left = centerX - width/2;
        double top = centerY - height/2;
        UI.drawRect(left , top, width, height );
        UI.drawString("area="+area, centerX, centerY);
    }
}
```

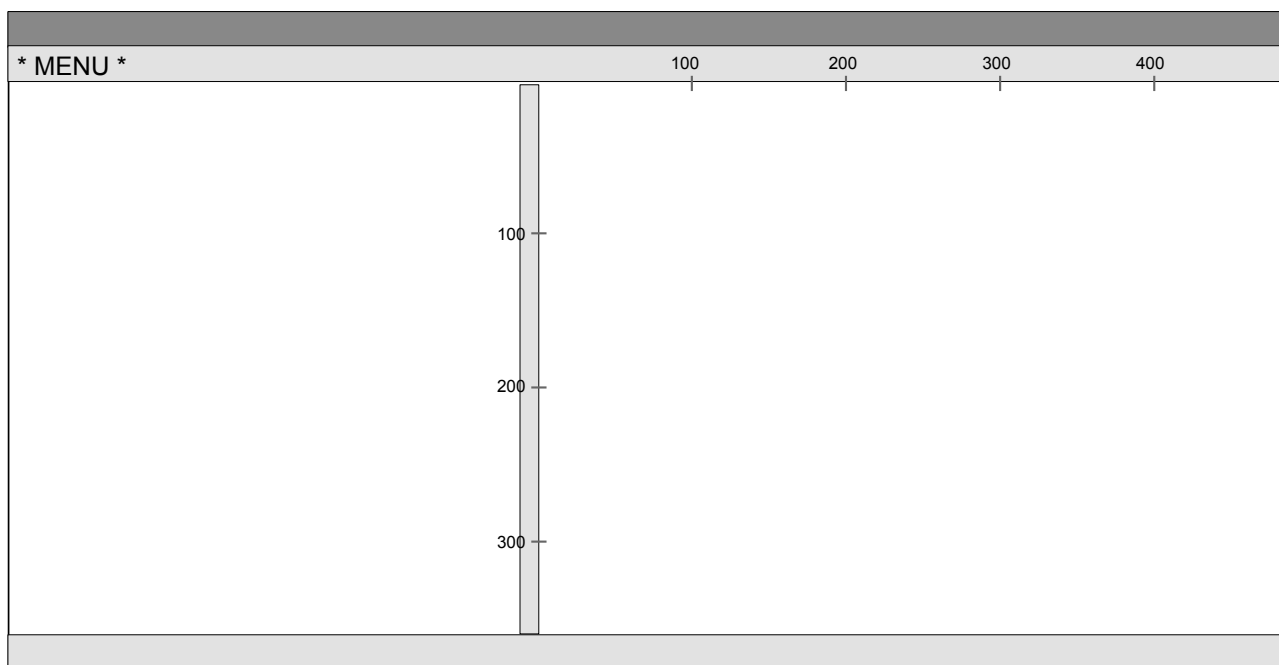
width:

height:

left:

top:

area:



**Question 2. Writing programs with input and output**

[5 marks]

Complete the following ticketCost method to calculate the ticket price for a group with adults and children. An adult ticket is \$10 and a child ticket is half price. The method should ask the user for the number of adults and the number of children, and then calculate and print out the cost.

For example, if you called the ticketCost method and entered 2 when asked for the number of adults and 4 for the number of the children, the interaction would be:

number of adults: 2

number of children: 4

The tickets for 2 adults and 4 children cost \$40.00

```
public void ticketCost(){
```

```
}
```

### Question 3. Understanding variables, if, while

[8 marks]

Consider the following testVariables method. In the box below, write the text that testVariables would output. Note that the first part of some lines of output is provided for you.

```
public void testVariables (){
    int n1 = 10;
    int n2 = 5;
    n1 = n1 + 1;
    n2 = n1 + n2;
    Ul.println ("n1 is "+ n1);
    Ul.println ("n2 is "+ n2);
    Ul.println ("a new value is "+ 2*n1);
    Ul.println ("one more "+ n1);
    int i = 0;
    while ( i < n2 ){
        if ( i > n1) {
            Ul.println (i);
        }
        i++;
    }
}
```

n1:	<input type="text"/>
n2:	<input type="text"/>
i:	<input type="text"/>

```
n1 is
n2 is
a new value is
one more
(---more numbers should be printed, write them below this line---)
```

**Question 4. Understanding method calls**

[7 marks]

Consider the following testMethods method. In the box below, write the text that testMethods would output. Note that the first part of each line of output is provided for you.

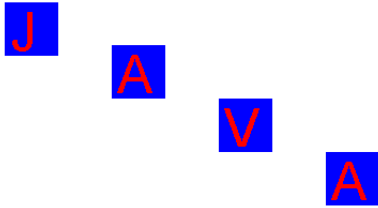
```
public void testMethods(){
    int m = 10;
    int n = 2;
    this.method1(m);
    int t = this.method2(m,n);
    Ul.println ("the result is " + t);
    Ul.println ("the result is " + this.method2(n,m));
}
public void method1(int x){
    x = x+1;
    Ul.println ("x is " +x);
}
public int method2(int x, int y){
    Ul.println ("x is " + x);
    Ul.println ("y is " + y);
    return (2*x + y);
}
```

```
x is
x is
y is
the result is
x is
y is
the result is
```

## Question 5. Defining and calling methods

[10 marks]

The following drawCharacters method prints out a number of characters on tiles at different places. Each tile has a blue background and a red letter in the middle. The tiles look like this:



```
public static final int TileSize = 50;

public void drawCharacters(){
    UI.setFontSize( TileSize );

    UI.setColor( Color.blue );
    UI.fillRect (100, 10, TileSize , TileSize );
    UI.setColor( Color.red );
    UI.drawString("J", 100+5 , 10+TileSize-5);

    UI.setColor( Color.blue );
    UI.fillRect (200, 50, TileSize , TileSize );
    UI.setColor( Color.red );
    UI.drawString("A", 200+5 , 50+TileSize-5);

    UI.setColor( Color.blue );
    UI.fillRect (300, 100, TileSize , TileSize );
    UI.setColor( Color.red );
    UI.drawString("V", 300+5 , 100+TileSize-5);

    UI.setColor( Color.blue );
    UI.fillRect (400, 150, TileSize , TileSize );
    UI.setColor( Color.red );
    UI.drawString("A", 400+5 , 150+TileSize-5);
}
```

drawCharacters is not well designed: it has quite a bit of repetition. It would be better design to define another method called drawTile which prints out a single tile, and make drawCharacters call the drawTile method four times, as in the version of drawCharacters on the facing page.

**(Question 5 continued)**

Complete the definitions of `drawCharacters` and `drawTile` below so that they do the same as the original version. You will need to determine the appropriate arguments for the calls to `drawTile`, the appropriate parameters in the header of `drawTile`, and the statements in the body of `drawTile`.

```
public static final int tileSize = 50;
```

```
public void drawCharactersFixed(){
```

```
    UI.setFontSize( tileSize );
```

```
    this.drawTile(
```

```
    this.drawTile(
```

```
    this.drawTile(
```

```
    this.drawTile(
```

```
}
```

```
public void drawTile(
```

```
}
```

## Question 6. Writing methods that use objects

[10 marks]

Suppose the Car class has one constructor and three methods:

Constructor:

```
public Car(double x, double y, Color color)
// constructs a Car object of the specified location (x,y) and color ,
// displays it at the location , facing right .
```

Methods:

```
public void drive (int time)
// makes the car move for the specified number of seconds in the specified direction .
//The direction is either right or left .
// Please note the car can only drive along a horizontal line , that is ,
// only its x location changes and its y location does not change.
// You can see the car moving step by step when the car at old position is erased ,
// and the car at a new position (some pixels head) is drawn, but you do not need to know
// these details .

public void turn()
// changes the direction from right to left , or from left to right

public double getLocation()
// returns the value of its x location
```

Complete the following carGame method, which should first create a red car at (100,200), and make the red car drive for 5 seconds, turn, and then drive for 3 seconds.

Then it should create a blue car at a random location, so the blue car may appear at any locations with x range from 100 to 500 and y range from 100 to 400 . It should then make the blue car drive to one side of the window, turn and then drive to the other side of the window. Make sure that the car does not drive out of the window, for example, you can set up two boundaries: x location at 50 and x location at 800. Keep the car moving within these boundaries and stop it when it has turned 5 times.



(Question 6 continued)

```
public void carGame(){
```

```
}
```

\*\*\*\*\*

**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.

**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.