

Family Name:..... Other Names:

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

Model Solutions

COMP 102: Test 1

31 March, 2014

Instructions

- Time allowed: **50 minutes** .
- Answer **all** the questions. There are 50 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	[6]	<input type="text"/>
2. Writing programs with input and output	[7]	<input type="text"/>
3. Understanding variables and expressions	[5]	<input type="text"/>
4. Writing methods with graphical output	[6]	<input type="text"/>
5. Writing methods that use objects	[6]	<input type="text"/>
6. Defining and Calling Methods	[10]	<input type="text"/>
7. Writing methods with if , return , and while	[10]	<input type="text"/>
	TOTAL:	<input type="text"/>

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. Understanding programs with input and output

[6 marks]

What will the following program do if the drawIt method is called, and the user types kitten when asked for input?

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{

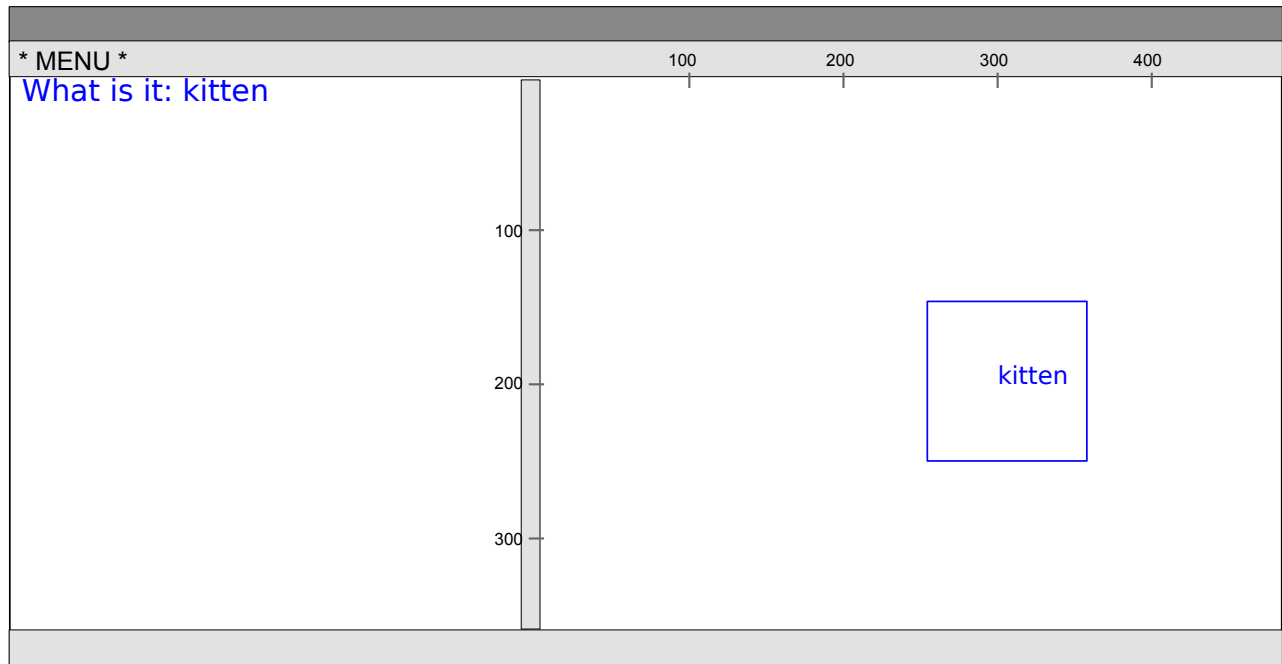
    public static final double Size = 100;
    public static final double CenterX = 300;
    public static final double CenterY = 200;

    public void drawIt(){
        String item = UI.askString("What label: ");
        double left = CenterX-Size/2;
        double top = CenterY-Size/2;
        UI.drawRect(left, top, Size, Size);
        UI.drawString(item, CenterX, CenterY);
    }
}
    
```

item:

left:

top:



Question 2. Writing programs with input and output

[7 marks]

Complete the following printOctagonArea method to calculate the area of a regular octagon. The method should ask the user for the length of the side and then calculate and print out the area.

Note, the area of a regular octagon is six times the square of the length of the side. [After Test Note: this is not the correct formula!!!]

For example, if you called the printOctagonArea method and entered 10 when asked for the length, the interaction would be:

```
Length of side: 10
Area = 600
```

```
public void printOctagonArea(){
    double length = UI.askInt("Length of side:");
    double area = 6 * length * length;
    UI.println("Area = " + area);
}
```

Question 3. Understanding variables and expressions

[5 marks]

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

Hint: Keep track of the values of the variables in the boxes.

```
public void mixUp(){
    String name = "Jim";
    int big = 5;
    int small = 10;
    Ul.println ("1 name: "+name);
    Ul.println ("1 big: "+big);
    Ul.println ("1 small: "+small);
    name = name + big + small;
    big = small + big;
    small = big * 2;
    Ul.println ("2 small: "+small);
    Ul.println ("2 big: "+big);
    Ul.println ("2 name: "+name);
}
```

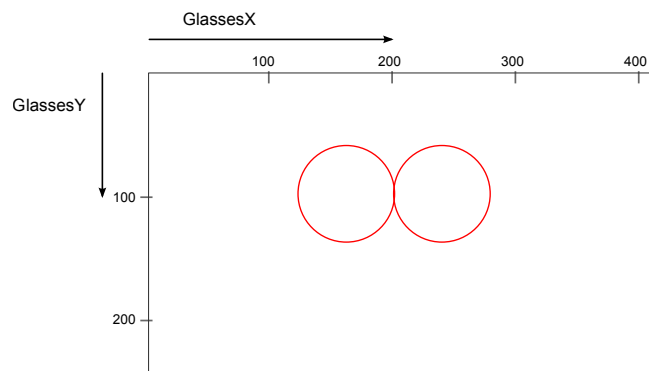
name: big: small:

```
1 name:    Jim
1 big:     5
1 small:   10
2 small:   30
2 big:     15
2 name:    Jim510
```

Question 4. Writing methods with graphical output

[6 marks]

Complete the drawGlasses method below, so that it draws a “pair of glasses” in the graphics pane, as shown here:



Result of calling `this.drawGlasses(80);`

drawGlasses has one parameter (diam), and should draw the outline of two red circles side-by-side touching each other on the graphics pane. The diameter of the circles should be the argument that was passed to the method. The place the circles touch should be at position (200,100). Your method should use the constants.

```
import ecs100.*;
import java.awt.Color;
public class GlassesDrawer{
    public static final double GlassesX = 200;
    public static final double GlassesY = 100;

    public void drawGlasses(double diam){
        double top = GlassesY - diam/2;
        double leftLeft = GlassesX - diam;
        UI.setColor(Color.red);
        UI.drawOval(leftLeft, top, diam, diam);
        UI.drawOval(GlassesX, top, diam, diam);
    }
}
```

Question 5. Writing methods that use objects

[6 marks]

Suppose the Snail class has one constructor and two methods:

Constructor:

```
public Snail(double size , Color color )  
    // constructs a Snail object , of the specified size and color ,  
    // displays it at a random position along the bottom of the graphics pane, facing up.
```

Methods:

```
public void crawl(int time)  
    // makes the snail move up the screen for the specified number of seconds ,  
    // leaving a trail behind it .
```

```
public void eat()  
    // makes the Snail eat some grass , leaving bits of grass around it .
```

Complete the following snailRace method, which should first create two Snail objects, of different sizes and different colors. It should then make the first snail crawl for 5 seconds, then make the second snail crawl for 4 seconds, then make each snail eat, and then the first snail crawl for 10 seconds and the second snail eat again.

```
public void snailRace(){  
  
    Snail s1 = new Snail(30, Color.red);  
    Snail s2 = new Snail(50, Color.blue);  
    s1.crawl(5);  
    s2.crawl(4);  
    s1.eat();  
    s2.eat();  
    s1.crawl(10);  
    s2.eat();  
  
}
```

Question 6. Defining and Calling Methods

[10 marks]

The following printLabels method prints out labels for items for a bus company. The labels show the destination city, the weight (with a note if the item is over 20kg), the type of the item (luggage or parcel), and a loading instruction depending on the item type.

```
public void printLabels (){
    Ul.println ("*****");
    Ul.println ("* City: Taupo");
    Ul.println ("* Weight: 17.5 kg");
    Ul.println ("* Type: Luggage");
    Ul.println ("* Loading: Stow forward");
    Ul.println ("*****");
    Ul.println ();
    Ul.println ("*****");
    Ul.println ("* City: Auckland");
    Ul.println ("* Weight: 33.0 kg");
    Ul.println ("* HEAVY!");
    Ul.println ("* Type: Parcel");
    Ul.println ("* Loading: stow at rear");
    Ul.println ("*****");
    Ul.println ();
    Ul.println ("*****");
    Ul.println ("* City: Hamilton");
    Ul.println ("* Weight: 19.1 kg");
    Ul.println ("* Type: Parcel");
    Ul.println ("* Loading: stow at rear");
    Ul.println ("*****");
    Ul.println ();
    Ul.println ("*****");
    Ul.println ("* City: Auckland");
    Ul.println ("* Weight: 24.8 kg");
    Ul.println ("* HEAVY!");
    Ul.println ("* Type: Luggage");
    Ul.println ("* Loading: Stow forward");
    Ul.println ("*****");
    Ul.println ();
}
```


(Question 6 continued)

printLabels is not well designed: it has quite a bit of repetition. It would be better design to define another method called printOneLabel which prints out a single label, and make printLabels call the printOneLabel method four times, as in the version of printLabels below.

Complete the definitions of printLabels and printOneLabel below so that they print out the same labels as the original version. You will need to determine the appropriate arguments for the calls to printOneLabel, the appropriate parameters in the header of printOneLabel, and the statements in the body of printLabels. You should not use more than three parameters.

```

public void printLabelsFixed (){

    this.printOneLabel("Taupo", 17.5, true);

    this.printOneLabel("Auckland", 33.0, false);

    this.printOneLabel("Hamilton", 19.1, false);

    this.printOneLabel("Auckland", 24.8, true);
}
public void printOneLabel( String city , double wt, boolean isLuggage){
    Ul.println ("*****");
    Ul.println ("* City: " + city);
    Ul.printf ("* Weight: %5.2f kg%n", wt);
    if (wt > 20){
        Ul.println ("* HEAVY!");
    }
    if (isLuggage){
        Ul.println ("* Type: Luggage");
        Ul.println ("* Loading: Stow forward");
    }
    else {
        Ul.println ("* Type: Parcel");
        Ul.println ("* Loading: Stow at rear");
    }
}

    Ul.println ("*****");
    Ul.println ();
}

```

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 7. Writing methods with if, return, and while

[10 marks]

Complete the following checkCode method which should check whether the given String is an acceptable code name. It should return true if the String is acceptable, and false otherwise.

An acceptable code name must

1. have a length between 6 and 12 characters, (eg, "ABC" is bad),
2. start with "A" or "Z", (eg, "Bcdefghi" and "abcdefg" are bad),
3. not contain the character "\$" (eg, "ABC\$DEFG" is bad),
4. not contain any character twice. (eg, "ZbcdebA" is bad),

You can get 5 marks if your method checks the first three conditions.

You can get a further 5 marks if your method also checks the final condition (requires a loop to check each character against the rest of the string).

Hint: look at the provided documentation for the String class.

```

public boolean checkCode(String name){
    if (name.length() < 6 || name.length() > 12 ) { return false; }
    if (name.contains("$")) { return false; }
    if (!name.startsWith("A") && !name.startsWith("Z")) {
        return false;
    }
    int i=0;
    while (i<name.length()){
        if (name.substring(i+1, name.length()).contains(name.substring(i, i+1))){
            return false;
        }
        i = i+1;
    }
    return true;
}

```

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.