

Family Name: Other Names:

Student ID: Signature

COMP 102 : Test

2017, Apr 3 ** WITH SOLUTIONS **

Instructions

- Time allowed: **50 minutes**
- Answer **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 program syntax	[10]	<input type="text"/>
2. Understanding programs with graphical output	[6]	<input type="text"/>
3. Writing programs with text input and output	[8]	<input type="text"/>
4. Writing programs with graphical output	[6]	<input type="text"/>
5. Writing methods with if	[6]	<input type="text"/>
6. Writing methods that use objects	[8]	<input type="text"/>
7. Using String methods and if	[6]	<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 program syntax**[10 marks]**

The following doWordGame method contains at least six syntax errors. The first error has been identified and corrected. Circle at least FIVE further errors and show how you would correct them.

Should be `/**`, not `/**`

`/**` Asks the user to enter a word!

** Says "Yes, that fits" if the word starts with "p" and is 7 characters long,*

** and "Sorry, that word won't work" otherwise.*

**/ [1] should have () in method header*

`public void doWordGameX{`

[2] needs " after String

`String word = UI.askString("Enter a word: X);`

[3] needs == not =

`if (word.startsWith("p") && (word.length() =X 7)X{`

[4] needs) after condition

`UI.println("Yes, that fits")X`

[5] needs ; at end

`}`

[6] needs { after else to match closing curly bracket

`else X`

[7] needs UI. before println

`X println("Sorry, that word won't work");`

`}`

`}`

Question 2. Understanding programs with graphical output

[6 marks]

Sketch what the following drawIt method will draw in the graphics pane.

```

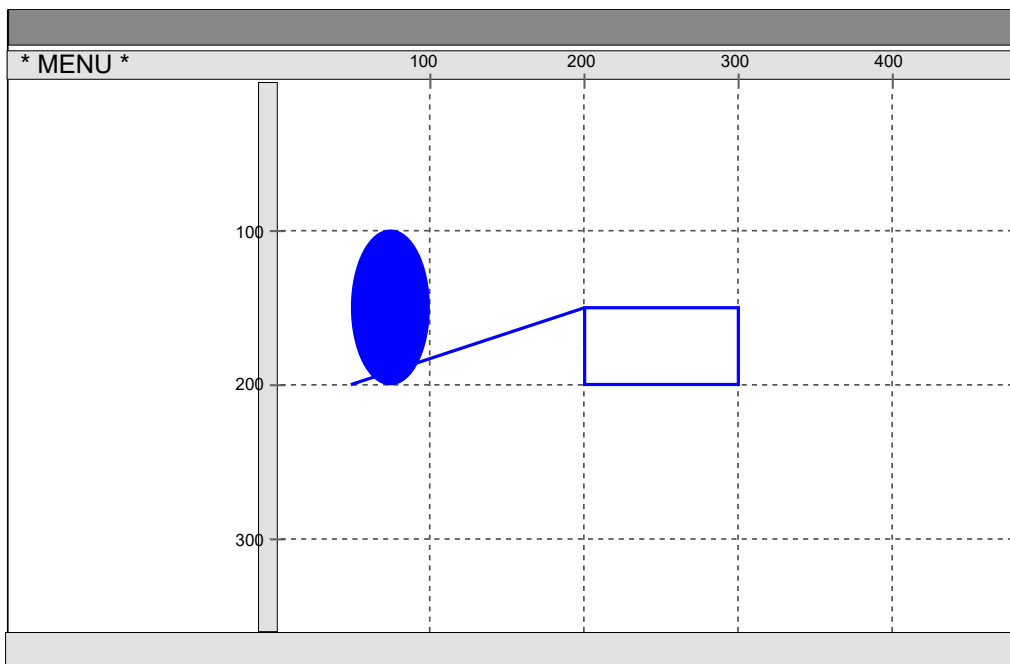
public static final double WIDTH = 100;
public static final double HEIGHT = 50;
:
public void drawIt(){
    UI.drawRect(200, 150, WIDTH, HEIGHT);
    UI.fillOval (50, 100, HEIGHT, WIDTH);
    double pos = WIDTH*2;
    double end = WIDTH+HEIGHT;
    UI.drawLine(50, pos, pos, end );
}
    
```

Hint: write the values of the variables in the boxes

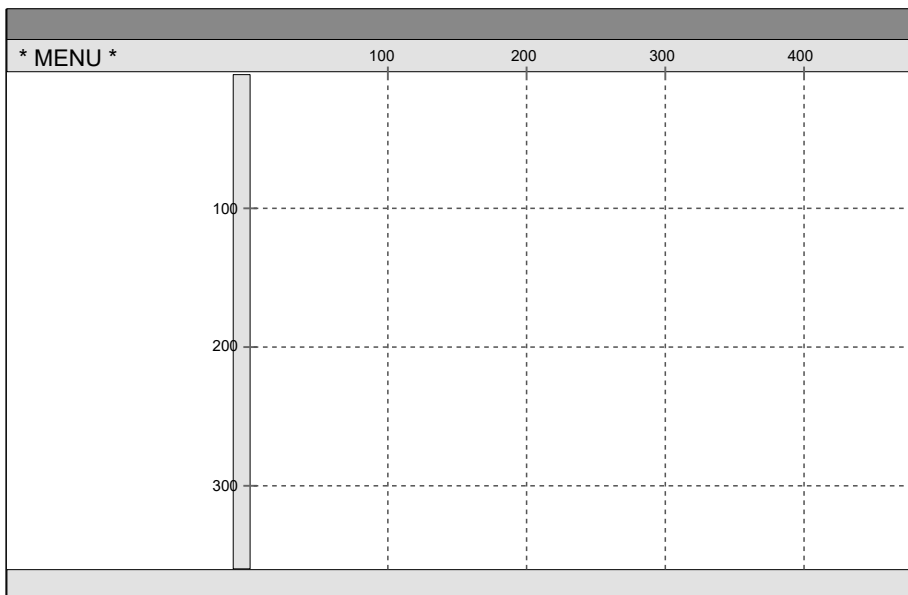
pos:

end:

Hint: The documentation page specifies the parameters of the drawing methods.



Extra copy (in case you made a mistake):



Question 3. Writing programs with text input and output**[8 marks]**

Complete the following `doCalculateEnergy` and `calculateEnergy` methods to calculate the kinetic energy of a moving vehicle.

- `doCalculateEnergy` should ask the user for the mass and speed of the vehicle and then call `calculateEnergy`.
- `calculateEnergy` should calculate the kinetic energy and print it out. The kinetic energy is given by:

$$\text{energy} = \frac{\text{mass} \times \text{speed} \times \text{speed}}{2}$$

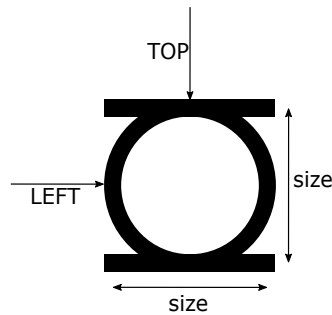
```
public void doCalculateEnergy () {
    double mass = UI.askDouble("Mass:");
    double speed = UI.askDouble("Speed:");

    this.calculateEnergy (mass, speed);
}

public void calculateEnergy (double m, double s){
    double energy = m * s * s / 2.0;
    UI.println ("Kinetic energy: " + energy);
}
```

Question 4. Writing programs with graphical output**[6 marks]**

Complete the following drawSign method to draw the following sign consisting of a circle and two lines.



The drawSign method should

- Compute values for the right and bottom of the sign, and put them in variables.
- Draw the circle and the two lines, using the variables.
- The position of the sign is defined by the constants LEFT and TOP.
- The parameter of drawSign specifies the size (both width and height) of the sign.

```

public static final double LEFT = 100;
public static final double TOP = 100;
    :
public void drawSign(double size){
    UI.setLineWidth(size /10.0);

    double right = LEFT + size;
    double bottom = TOP + size;

    UI.drawOval(LEFT, TOP, size, size);
    UI.drawLine(LEFT, TOP, right, TOP);
    UI.drawLine(LEFT, bottom, right, bottom);

}

```

Question 5. Writing methods with if**[6 marks]**

Complete the printSmallest method below.

- If the two arguments are the same, it should print "equal".
- If the arguments are different, it should print the smallest of its two arguments.

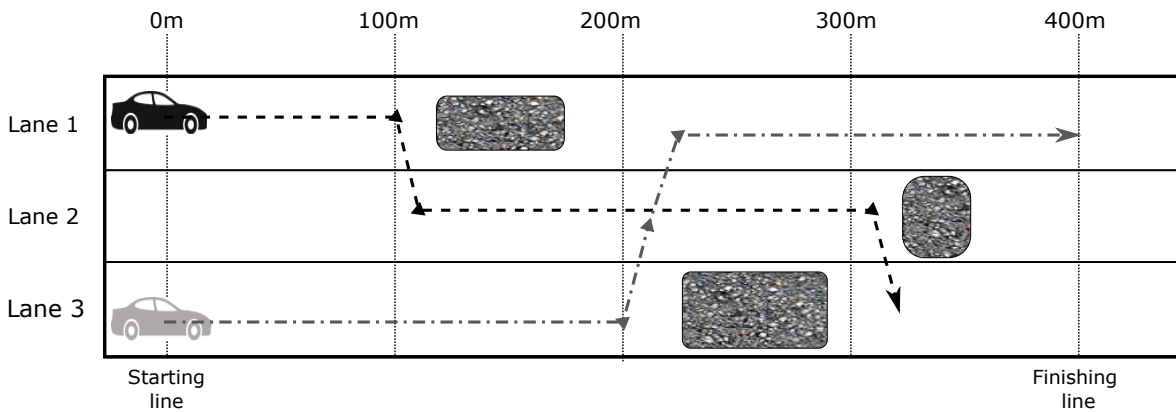
For example:

- this.printSmallest(10,50) should print 10,
- this.printSmallest(10,2) should print 2,
- this.printSmallest(5,5) should print equal

```
public void printSmallest (double first , double second) {  
    if ( first == second) {  
        Ul. println ("equal");  
    }  
    else if ( first < second) {  
        Ul. println ( first );  
    }  
    else {  
        Ul. println (second);  
    }  
  
}
```

Question 6. Writing methods that use objects**[8 marks]**

This question is about a program that makes two cars move along a racetrack. The racetrack has three lanes and some obstacles:



Complete the race method on the facing page so that it

1. Puts two cars at the start of the racetrack:
 - a black car in lane 1
 - a gray car in lane 3
2. Calls a sequence of methods on the two cars to make them take turns moving along the dotted paths on the diagram. Each car should make FOUR moves.

Note: You don't have to draw the racetrack.

The documentation for the Car class is given below.
It has one constructor and three methods:

```
// Constructor:
public Car(Color color, int lane)
  /** Creates a new Car object of the specified colour, placing it
      at the starting line in the specified lane. */

//Methods:
public void forward(double distance)
  /** Moves the Car forward the specified distance in its current lane. */

public void upLane()
  /** Moves the Car to the next lane up while moving 10m forward. */

public void downLane()
  /** Moves the Car to the next lane down while moving 10m forward. */
```

Hint: Possible color values are listed in the documentation.


```
/** Creates two cars then makes them take turns moving */
```

```
public void race(){  
    Car r1 = new Car(Color.black, 1);  
    Car r2 = new Car(Color.gray, 3);  
    r1.forward(100);  
    r2.forward(200);  
    r1.downLane();  
    r2.upLane();  
    r1.forward(200);  
    r2.upLane();  
    r1.downLane();  
    r2.forward(180);
```

```
}
```

Question 7. Using String methods and if.**[6 marks]**

The `checkCode(String code)` method on the facing page should check whether its argument satisfies the following constraints and prints out

- “Bad”, if it fails any constraints or
- “Good”, if it satisfies all the constraints.

A string is a valid code if

- its length is between 6 and 20 characters, *and*
- it does not contain two ';' characters in a row, *and*
- it starts with either '0' or '1', *and*
- the letters in the code are not all lowercase, *and*
- it does not consist of just a substring followed by a copy of the substring (eg, "1Day1Day").

Your method will need to use methods from the String class on the last page of the documentation sheets.

// Alternative answer:

```
public void checkCode(String code){
    String firstHalf = code.substring(0,code.length()/2);
    String secndHalf = code.substring(code.length()/2, code.length ());

    if (( code.length() >= 6 ) &&
        ( code.length() <= 20 ) &&
        ( code.startsWith("0") || code.startsWith("1") ) &&
        ! ( code.contains(";;") ) &&
        ! ( code.toLowerCase().equals(code) ) &&
        ( code.length()% 2 !=0 || ! ( firstHalf .equals(secndHalf)) ) ) {
        UI.println ("GOOD");
    }
    else {
        UI.println ("BAD");
    }
}
```

(Question 7 continued)

```

public void checkCode(String code){
    if ( (code.length() > 20) || (code.length() < 6) ) {
        Ul.println ("BAD");
    }
    else if ( ! code.startsWith("0") && ! code.startsWith("1") ){
        Ul.println ("BAD");
    }
    else if ( code.contains(";;") ){
        Ul.println ("BAD");
    }
    else if ( code.toLowerCase().equals(code) ){
        Ul.println ("BAD");
    }
    else if ( code.length()% 2 ==0 &&
        code.substring (0,code.length ()/2). equals( code.substring (code.length ()/2, code.length ()) ) )
        Ul.println ("BAD");
    }
    else {
        Ul.println ("GOOD");
    }
}

```

ALTERNATIVE:

```

int len = code.length();
String firstHalf = code.substring (0, len /2);
String secondHalf = code.substring (len /2, len );
if ( ( len >=6 && len <= 20 ) &&
    ( code.startsWith("0") || code.startsWith("1") ) &&
    ! code.contains(";;") &&
    ! code.toLowerCase().equals(code) &&
    ! firstHalf .equals(secondHalf) ) {
    Ul.println ("GOOD");
}
else {
    Ul.println ("BAD");
}
}

```

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.