

Family Name: Other Names:

Student ID: Signature

COMP 102/112 : Test

2021, March 25 ** WITH SOLUTIONS **

Instructions

- Time allowed: **50 minutes**
- Attempt **all** the questions. There are 50 marks in total.
- Write your answers in this test paper and hand in all sheets. (We have different instructions for distance students)
- If you think a 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 and calculators.
- You may write notes and working on this paper, but make sure your answers are clear.
- You may assume all the programs import the ecs100 library and other standard libraries.

Questions

Marks

1. Input & output of different types	[6]	<input type="text"/>
2. Writing programs with <u>if</u>	[10]	<input type="text"/>
3. Defining methods with parameters	[10]	<input type="text"/>
4. Writing methods that use objects	[10]	<input type="text"/>
5. Writing programs with <u>for</u>	[14]	<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. Input & output of different types**[6 marks]**

Complete the echo() method below so that it first asks the user to input an integer number, a float point number, a word, a true/false value, and then prints the four values out in a single line.

The input prompt messages should look like

```
integer:
number:
word:
is it true?
```

Suppose the user enters 10, 2.5, good, yes, the output should be

```
You entered: 10, 2.5, good, true
```

```
public void echo () {

    int first = UI.askInt("integer:");
    double second = UI.askDouble("number:");

    String word = UI.askString("word:");
    Boolean flag = UI.askBoolean("is it true?");

    UI.println ("You entered: "+ first +", "+ second +", "+ word +", "+ flag);
//OR
    UI.printf ("You entered: %d, %.1f, %s, %b\n", first, second, word, flag);

}
```

Question 2. Writing programs with if**[10 marks]**

The program on the next page has two methods. The `testMessages()` method reads a list of numbers and then uses a **for each** loop to print out a message for each number.

Your task is to complete the `getMessage(...)` method to **return** a String message for the number specified in the parameter, according to the following table.

<u>Number</u>	<u>Message</u>
from 10 up to 20	good
smaller than 5	too small
larger than 50	too big
other numbers	not sure

For example, if the user enters the list 40, 60, 12, 2, 90, 10, the `testMessages()` method should print out

```
not sure
too big
good
too small
too big
good
```

```
public class Program2{

    public void testMessages() {
        ArrayList<Double> nums = UI.askNumbers("Enter numbers:");
        for (double number : nums) {
            UI.println (this.getMessage(number));
        }
    }

    /** Returns the message associated with the given number */
    public String getMessage(double number){

        if ((number>=10) && (number<=20))
            return "good";
        else if (number>50)
            return "too big";
        else if (number<5)
            return "too small";
        else(
            return "not sure";

        }

    }
}
```

Question 3. Defining a method with parameters**[10 marks]**

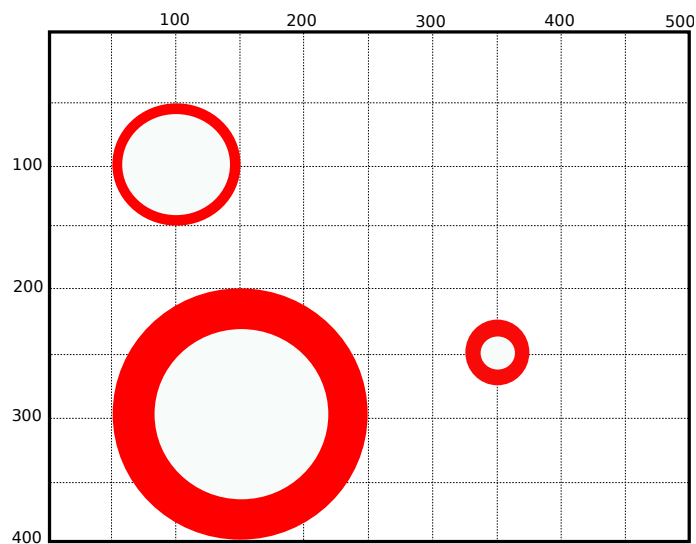
The program on the facing page has two methods. The `testDrawRing()` method calls the `drawRing(...)` method three times to draw three different rings.

(a) **[7 marks]** Complete the `drawRing(...)` method to draw a ring (draw a large red circle then a smaller white circle).

You will need to define four parameters for `drawRing(...)`:

- the x, y position of the center of the two circles.
- the diameter of the big circle.
- the width of the ring (radius of the big circle minus the radius of small circle)

(b) **[3 marks]** Complete the `testDrawRing()` method to draw three rings like the ones in the example picture below. (The widths of the rings do not need to be exactly the same as the picture).



```
public class Drawer {  
  
    // Draws one ring centered at given position and of the specified diameter and width.  
    public void drawRing( double x, double y, double diameter, double width ) {  
  
        double diameterSmall = diameter - 2* width;  
        UI.setColor( Color.red );  
        UI.fillOval (x-diameter/2,y-diameter/2,diameter, diameter);  
        UI.setColor( Color.white);  
        UI.fillOval (x-diameterSmall/2,y-diameterSmall/2, diameterSmall, diameterSmall);  
  
    }  
    // Draws three rings.  
    public void testDrawRing() {  
        UI.clearGraphics ();  
        this.drawRing(          100,  100, 100, 10          );  
  
        this.drawRing(          350, 250,  50, 15          );  
  
        this.drawRing(          150, 300, 200, 30          );  
  
    }  
}
```

Question 4. Writing methods that use objects**[10 marks]**

This question is about a program that stores, updates and displays information about students, including the courses they are taking. It uses a Student class to store, update, and display information about an individual student. The Student class has the following constructor and methods:

```
class Student {  
  
    // Constructor to make a new Student object  
  
    public Student(String name, int id, int year)  
        // Creates a new Student object with specified name, id, and the year they started their study  
  
    // Methods that can be called on a Student object  
  
    public void addCourse(String crseCode)  
        // add crseCode to the list of courses.  
  
    public void printIdentity ()  
        // Prints the student's name and ID.  
  
    public void printCourses ()  
        // Prints all the courses.  
  
    public int getYearOfStudy(int currentYear) {  
        // Returns the student's current year of study.  
        // Parameter is the current year.  
        // If the students started their study 2 years before the current year,  
        // then they are in their 3rd year of study.  
    }  
}
```

Complete the following twoStudents() method in the StudentAdmin class so that it

1. Creates two Student objects with the following information
 - Name: Alex; ID: 3020; Started in year: 2021
 - Name: Sean; ID: 3005; Started in year: 2018
2. Adds COMP102 and COMP103 to Alex's record
3. Adds MATH324 to Sean's record
4. Prints Sean's name, ID and courses
5. Prints the courses for Alex
6. Adds MATH335 to Sean's record
7. For the first student (Alex), print out a welcome message saying "Welcome to University!" if 2021 is the first year of study, and "Welcome back!" if the student is in the second or later year of study.

You **must** use methods in the Student class to do all the steps above.

```

public class StudentAdmin{
    public void twoStudents(){

        Student st1 = new Student("Alex", 3020, 2021);
        Student st2 = new Student("Sean", 3005, 2018);

        st1.addCourse("COMP102");
        st1.addCourse("COMP103");
        st2.addCourse("MATH324");
        st2.printIdentity ();
        st2.printCourses ();

        st1.printCourses ();
        st2.addCourse("MATH335");

        if (st1.getYearOfStudy(2021)>0){
            Ui.println ("Welcome back!");
        }
        else {
            Ui.println ("Welcome to University!");
        }

    }
}

```

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 5. Writing methods with for**[14 marks]**

The program for this question has three methods:

- analyseNumbers(), which asks the user for a sequence of numbers and then shows the result of calling the other two methods.
- findAverage(...)
- plotNumbers(...)

(a) **[7 marks]** Complete the findAverage(...) method to find and return the average of a list of numbers.

Note: This method does **not** print out the average value; it just returns the value.

You may assume the ArrayList is not empty

```
public void analyseNumbers () {
    ArrayList<Double> nums = UI.askNumbers("Enter numbers [0, 400]");
    UI.println ("The average is " + this.findAverage(nums));
    this.plotNumbers(nums);
}

public double findAverage( ArrayList<Double> nums) {

    int count = 0;
    double runningTotal = 0;

    for (double score: nums) {
        runningTotal = runningTotal + score;
        count = count + 1;
    }

    double average = runningTotal/count;
    return average;

}
```

(Question 5 continued)

(b) [7 marks] Complete the `plotNumbers(...)` method to plot the numbers relative to the average value.

- Draw a blue horizontal line (at $y = 200$) to represent the average value.
- Draw a small dot (radius = 5) to show the value of each number relative to the average.
- Draw a vertical line from each dot to the blue line.
- If the number is bigger than the average, draw it green, otherwise draw it red.

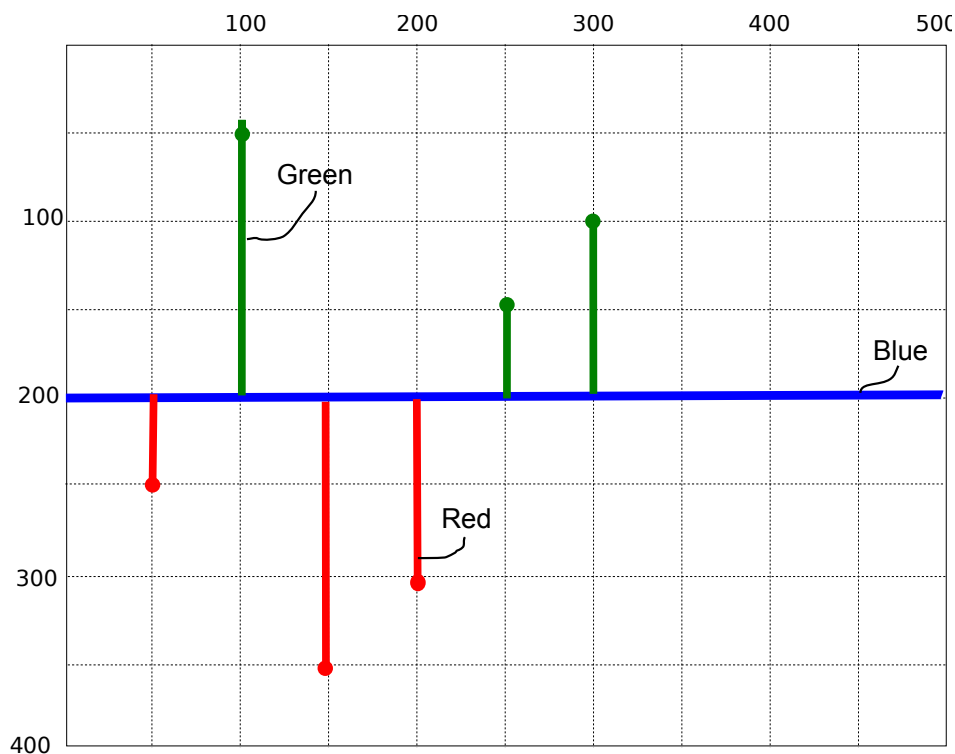
Draw the vertical lines every 50 units apart on the x-axis.

You may assume the user enters at least one and at most ten numbers and all numbers are in the range of $[0, 400]$.

For example, if the user enters the following numbers:

```
100
300
0
50
200
250
done
```

The average is 150 and the plot should look like this:



(Question 5 continued)

```
public void plotNumbers(ArrayList<Double> nums){
    UI.clearGraphics ();

    UI.setColor ( Color.blue );
    UI.drawLine(0, 200, 450, 200);

    double ave = findAverage(nums);
    int i = 0;
    for( double d : nums ) {
        if ( d >= ave ) {
            UI.setColor( Color.green );
        }
        else{
            UI.setColor( Color.red );
        }
        UI.fillOval ( 50 + i * 50 -5, 200-(d-ave)-5, 10, 10);
        UI.drawLine(50 + i*50 , 200-(d-ave), 50 + i*50, 200);
        i++;
    }
}
```

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.