Family Name:..... **Other Names:** .....

Student ID:..... **Signature**.....

COMP 102 : Test #2

2022, 2 December ** WITH SOLUTIONS **

Instructions

- Time allowed: **45 minutes**
- Attempt **all** the questions. There are 30 marks in total.
- Write your answers in this test paper and hand in all sheets. (Distance students: see instructions on OnlineTests webpage)
- If you think a question is unclear, ask for clarification.
- Brief Java documentation is provided with the test.
- This test contributes 10% of your final grade.
- 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. Using foreach loops with numbers	[8]	
2. Using Objects	[8]	
3. Using conditionals and foreach loops with Strings	[8]	
4. Debugging foreach Loops with variables	[6]	
	TOTAL:	

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. Using foreach loops with numbers **[8 marks]**

Complete the addLargeNumbers(...) method below so that it

- asks the user for an ArrayList of numbers
- adds up all the numbers in the list that are larger than 10.
- prints out the total

```
public void addLargeNumbers(){
    ArrayList<Double> numbers = UI.askNumbers("Enter numbers");
    double total = 0;
    for (double num : numbers){
        if (num > 10) {
            total = total + num;
        }
    }
    UI.println("Total of positive numbers = " + total);




}
```

Question 2. Using Objects **[8 marks]**

Suppose you are writing a program to keep track of all the vaccination centers in NZ, along with the number of vaccinations they have administered.

As part of this program, you have designed a VaccCenter class. Each VaccCenter object will contain information about one center, including its town, map coordinates, opening day, and the number of vaccinations adminstered so far.

It has a constructor and several methods:

```
class VaccCenter

// Constructor to make a new VaccCenter object

    public VaccCenter(String town, double latitude, double longitude, int openingDay)
    // Requires four arguments:
    //   the name of the town/suburb where it is located,
    //   the map coordinates, (latitude and longitude)
    //   the day of the current year that the center was opened (1..365)


// Methods that can be called on a VaccCenter object

    public void update(int numVaccines, int day)
      // Update the vaccination center with the number of vaccinations since the last update.
      // Parameters are the number of vaccinations and the day of this update

    public void report()
      // Prints out a report on the vaccination center, showing the current details

    public void display()
      // Draws the vaccination center at the right place on the map as a circle

    public String getTown()
      // Returns the name of the town of the center.

    public int getTotal()
      // Returns the total number of vaccinations at this center so far
```

(Question 2 continued on next page)

(Question 2 continued)

To test your class, complete the following testVaccCenters() method so that it creates two VaccCenter objects and uses methods in the VaccCenter class to run through the following scenario.

1. Open a center in Wellington on day 20 (20 Jan); coordinates are (lat: -41.28, long: 174.78)
2. Update the Wellington center with 300 vaccines by day 35
3. Print a report on the Wellington center.
4. Open a center in Waikanae on day 46 (15 Feb); coordinates are (lat: -40.87, long: , 175.06)
5. Update the Waikanae center with 71 vaccines by day 50
6. Update the Wellington center with 460 more vaccines by day 50
7. Display both centers on the map.
8. Print out the total of all the vaccinations given in both centers.

```
public void testVaccCenters(){
        VaccCenter wgtn = new VaccCenter("Wgtn Central", -41.25, 174.78, 20);
        wgtn.update(300,35);
        wgtn.report();
        VaccCenter waik = new VaccCenter("Waikanae", -40.87, 175.06, 46);
        waik.update(71,50);
        wgtn.update(460,50);
        waik.display();
        wgtn.display();
        int total = wgtn.getTotal() + waik.getTotal();
        UI.println("Total vaccinations = "+total);









}
```

Question 3. Using conditionals and foreach loops with Strings **[8 marks]**

The classifyContacts(..) method on the next page is passed an ArrayList of Strings.

Each String is the description of a contact, eg a phone number, an email address, a physical address, or a postal address.

For each String in the list, the method should print out the classification of the contact, according to the following (simplified) rules, which should be checked in order.

- A contact containing "P.O. Box" is a **postal** address;
- A contact containing an "@" and at least one "." is an **email** address;
- A contact containing "Road" or "Street" or "Avenue" is a **physical** address;
- Anything else is **unclassified**

For example, given the list:

{"chef@GreenStreet.com", "74 Aro Avenue, Waikanae", "+64 27 3981 3256",
"FastFood, P.O. Box 53, Kelburn", "12 High Street, Petone"}

The method should print out:

```
chef@GreenStreet.com:  email
74 Aro Avenue, Waikanae:  physical
+64 27 3981 3256:  unclassified
FastFood, P.O. Box 53, Kelburn:  postal
12 High Street, Petone:  physical
```

(Question 3 continued on next page)

(Question 3 continued)

Complete the classifyContacts(..) method:

```
public void classifyContacts (ArrayList<String> contacts){
    for (String contact : contacts){
        if (contact.contains("P.O. Box")) {
            UI.println(contact + ": postal");
        }
        else if (contact.contains("@") && contact.contains(".")) {
            UI.println(contact + ": email");
        }
        else if (contact.contains("Road") || contact.contains("Street") ||
                 contact.contains("Avenue")) {
            UI.println(contact + ": physical");
        }
        else {
            UI.println(contact + ": unclassified");
        }
    }









}
```

Question 4. Debugging foreach loops with variables **[6 marks]**

The findClosest(...) method is passed an ArrayList of numbers and a target number and should then find and return the number in the list that is closed to the target.

For example, if it is passed the list {35.0, 52.0, 84.0, 14.1} and the target 49.0, it should return 52.0

The following version of findClosest(...) does not work correctly.

```
public double findClosestBad( ArrayList<Double> numbers, double target){
    double closest = 0;
    for (double num : numbers){
        if ((target - num) < (target - closest)){
            closest = num;
        }
    }
    return closest ;
}
```

(a) **[1 mark]** What does the above version of findClosestBad(...) return if it is passed the list [35.0, 52.0, 84.0, 14.1] and the target 49.0?

```
84.0
```

(Question 4 continued on next page)

(Question 4 continued)

(b) **[5 marks]** Write a correct version of findClosest(...) which will work with any list containing numbers and any target value.

```
public double findClosest ( ArrayList<Double> numbers, double target){
    double closest = Double.POSITIVE_INFINITY;
    for (double num : numbers){
        if (Math.abs(num − target) < Math.abs(closest − target)){
            closest = num;
        }
    }
    return closest ;
//OR
    double closest = 0;
    boolean firstTime = true;
    for (double num : numbers){
        if (firstTime) {
            closest = num;
            firstTime = false;
        }
        else if (Math.abs(num − target) < Math.abs(closest − target)){
            closest = num;
        }
    }
    return closest ;



}
```

* * * * * * * * * * * * * * *