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

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

COMP 102: Lab Assessment 2

2021, Dec 6

Instructions

- Time allowed: **90 minutes**
- Attempt **all** the questions. There are 50 marks in total.
- To write your answers in BlueJ, follow the instructions on page 2.
- If you think a question is unclear, ask for clarification.
- This lab assessment contributes 15% of your final grade
  (But your mark will be increased to your final lab assessment mark if that is higher.)
- You may access the online Java Documentation.
- You may use dictionaries and calculators.
- You may not access any other web sites or online help of any kind.
- You may write notes and working on this paper, but make sure your answers are clear.
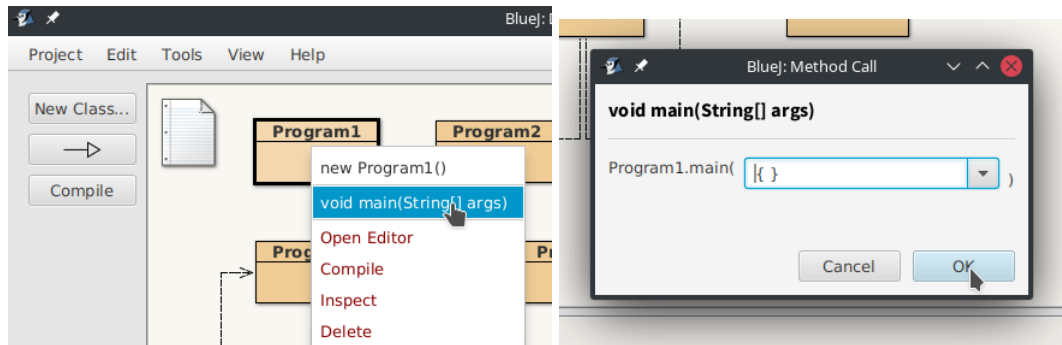
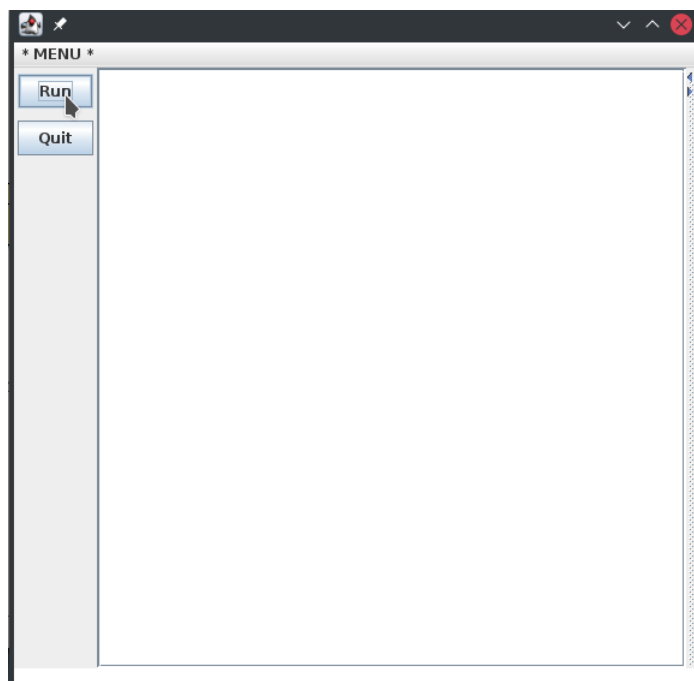| Questions	Marks	
1. Loops	[4]	
2. Scanners	[8]	
3. Defining Classes	[13]	
4. Event-Driven Input	[15]	
5. Files	[10]	
	TOTAL:	

Running the code:

1. Download the zip file from the url below:

   https://ecs.wgtn.ac.nz/Courses/COMP102_2021T3/LabAssessment2

2. Unzip the file, and open the project in BlueJ.

3. Each question has its own class. To run them, right-click and select 'void main(String[] args)'. Hit 'OK' when it prompts you for arguments.

4. The UI that pops up will have a 'Run' button that will run the code for the question, once you've written it.

Question 1. While Loops   **[4 marks]**
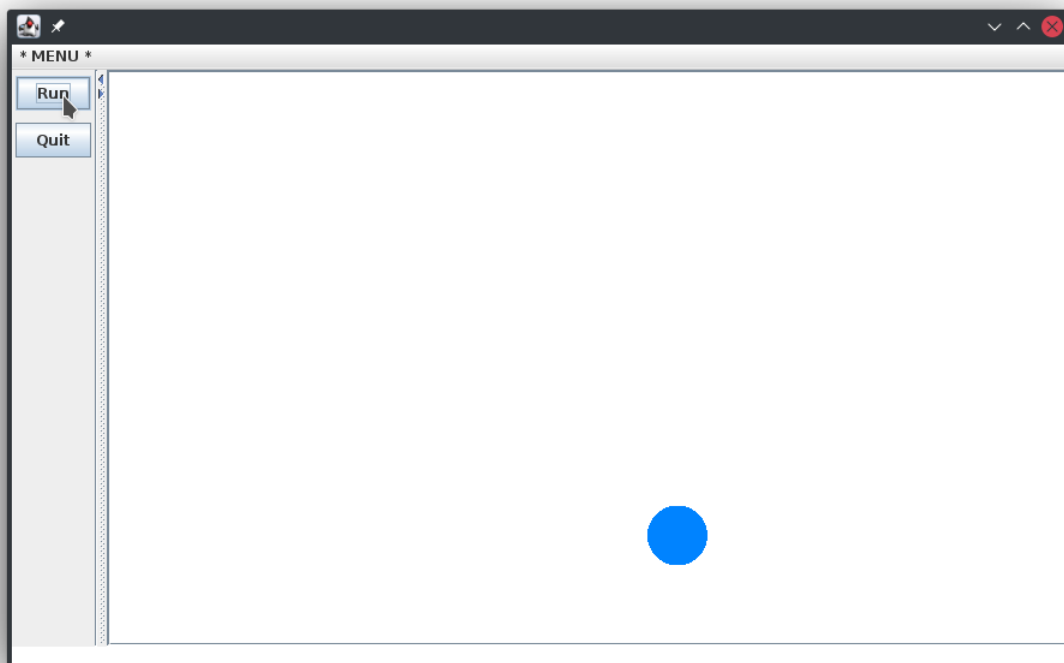
Answer this question in the Program1 class.

Use a while loop to complete the doBubble() method in Program1, which should make the bubble move to the top of the screen.

The distance the bubble should move upwards every step has been provided as the constant BUBBLE_STEP, as have methods to erase and draw the bubble.

Note:

- You must use a while loop
- You should use UI.sleep(50) to pause at the end of every step so that the animation doesn't move too quickly to see.

A bubble slowly rises to the top of the screen

Question 2. Scanners [8 marks]

Answer this question in the Program2 class.

The program in Program2 is an extremely simple text-based bot, like you might see in applications like discord or irc.

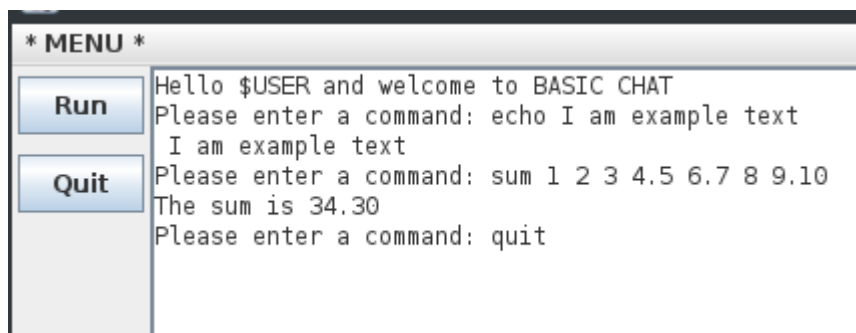It can support three commands: quit, echo, and sum.

quit exits the program

echo repeats text back to the user
> e.g. the input "echo hello my name is simplebot" would result in the program printing " hello my name is simplebot"

sum adds up the numbers and prints the sum
> e.g. the input "sum 1 3 5.8 2.2" would result in the program printing "The sum is 12.00"

```
* MENU *
        Hello $USER and welcome to BASIC CHAT
 Run    Please enter a command: echo I am example text
         I am example text
 Quit   Please enter a command: sum 1 2 3 4.5 6.7 8 9.10
        The sum is 34.30
        Please enter a command: quit
```

(a) **[4 marks]** Complete the run() method

The run() method asks the user for a command, and then uses a scanner to determine which command to run. The first token of the user's input will be the command, and the rest of the input will be any extra information.

Complete the implementation so that it uses the provided scanner to:

- Read in the first **token** of the input string, which will be one of the commands: "quit", "sum", or "echo"
- Quit the program (using break) if the command is "quit"
- Call the doEcho(...) method if the command is "echo"
- Call the doSum(...) method if the command is "sum"

(b) **[2 marks]** Complete the doEcho(...) method

The doEcho(...) method accepts a scanner as a parameter.

This method should read in the next line of input from the scanner, and print it out.

(c) **[2 marks]** Complete the doSum(...) method

The doSum(...) method accepts a scanner as a paramter.

This method should read all of the numbers (doubles) from the scanner, and calculate the sum.

Question 3. Defining Classes   **[13 marks]**

Answer this question in the Program3 class.

This question is about an arcade-style game where a small green orb needs to cross a road.

(a) **[10 marks]** Design a Car class to represent Car objects.

A car is represented on the screen by a coloured rectangle. It moves from left to right, or from right to left.

The car class should contain:

- Fields to store the position (x and y) of the car, its speed, and its colour.
  - The position will determine where on the screen the car is drawn, and represents the **centre** of the car
  - The speed should be a double, and is a permitted to be negative
  - The colour of the car will determine the colour of the rectangle that is drawn.
- A constructor that initializes the fields.
- The draw() method that draws the car centred at the location stored in its fields..
  - A filled rectangle of the appropriate size and colour should be drawn first.
  - Then, a black unfilled rectangle should be drawn to provide an outline.
- The step() method that should just change the car's x position according to its speed.
- The getX() method that returns the X position of the car.
- The isInside(…) method that calculates if the given point is inside the car.

(b) **[6 marks]** Complete the makeCar() method in the Program3 class

Complete the method so that it will create and return a new car object according to the following specifications:

- If the fastLane parameter is TRUE, the car should be positioned at FAST_LANE_X, FAST_LANE_Y, with a speed of -15.0 (negative = moving left)
- If the fastLane parameter is FALSE, the car should be positioned at SLOW_LANE_X, SLOW_LANE_Y, with a speed of 10.0
- In both cases, the car should be a random colour

Don't get hit!

Question 4. Event-Driven Input [15 marks]

Answer this question in the Program4 class.

Complete the Program4 class so that it allows the user to create basic collages by drawing other images onto the screen (stamps), drawing lines, and adding text.

The program should have:

- A button for each tool (Stamp, Line, Text) that changes the currently active tool
- A slider that changes the size of the text (min: 8, max: 50, initial: 10)
- A text field that updates the current text that is drawn with the text tool

You will also need to write the mouse listener.

- When the mouse is **pressed**, the program should just record the location.
- When the mouse is **released**, the program should perform one of the following:
    - If the current tool is "Line", draw a line between where the mouse was pressed and where it was released
    - If the current tool is "Text", draw the current text on the screen at the location the mouse was released
    - If the current tool is "Stamp", draw the current stamp image on the screen between where the mouse was pressed and where it was released.

    Hint: The addSlider method requires a name, a minimum value, a maximum value, an initial value and a method to call.

    Hint: To draw the stamp in the correct place, use the left, top, width, and height variables that have already been provided in the doMouse method.
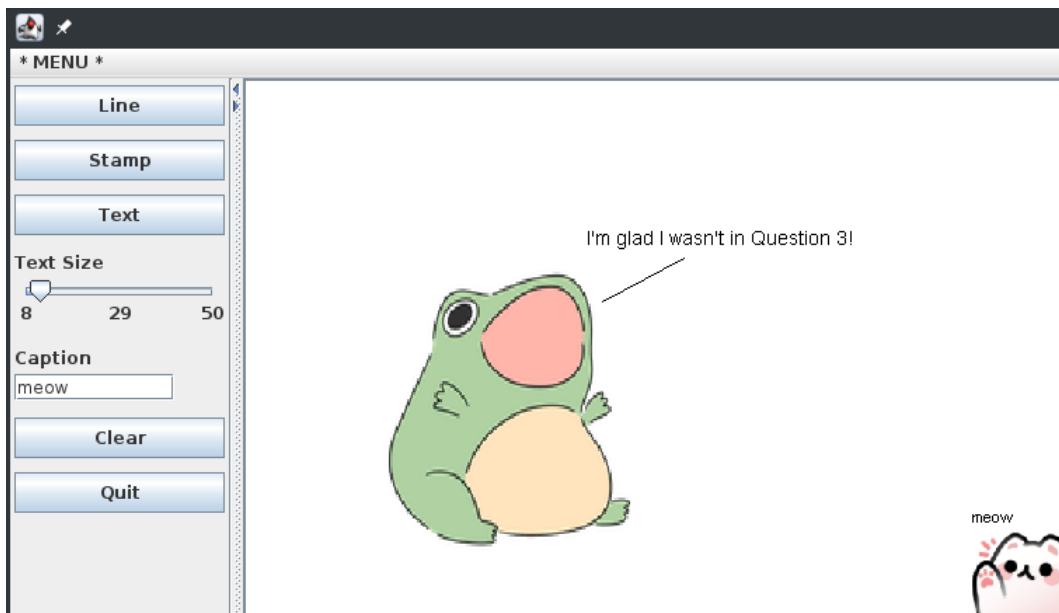
Question 5. Files [10 marks]

An excerpt of text file is shown below.

```
The sun had far descended, and I still sat on the shore,
satisfying my appetite, which had become ravenous, with
an oaten cake, when I saw a fishing-boat land close to
me, and one of the men brought me a packet; it contained
letters from Geneva, and one from Clerval entreating me
to join him.
```

Complete the run() method in Program5 that will ask the user for a letter, and then:

- count the number of words (tokens) in the file that **begin** with that letter, *and*
- calculate the average length of those words, *and*
- print out the longest word that starts with that letter

For example, if you enter the letter "W", your program should print:
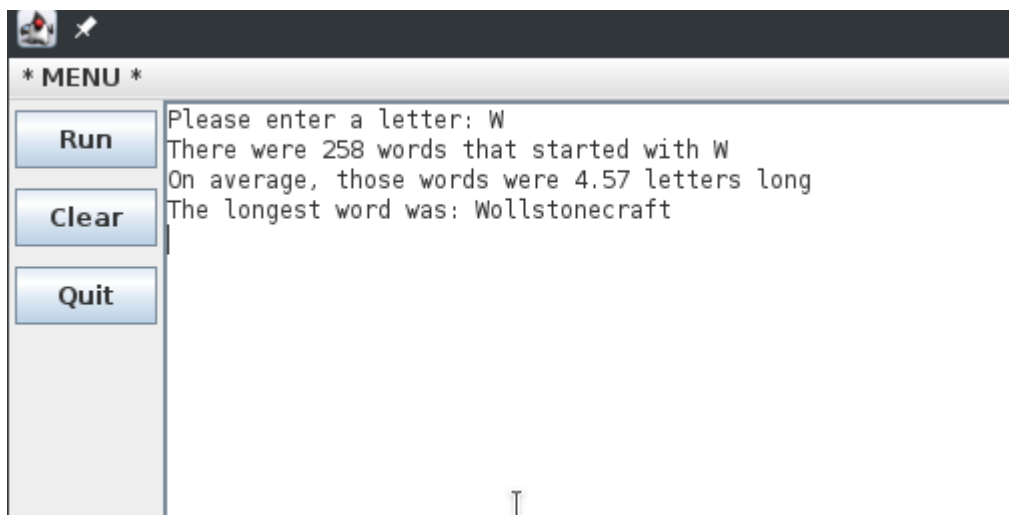
Hint: You do not need to worry about case.

Hint: You can use the String method startsWith(…) to check if a string starts with a specific letter.

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