RIT | Golisano College of Computing and Information Sciences
School of Information

ISTE-121
Lab 01: GUI Applications with I/O

Exercise 1: Simple GUI Application

Overview
The purpose of this exercise is to practice writing Java GUI applications. Like most software, you will build this in stages starting with design, then testing each stage before proceeding to the next.

This lab will provide a description of the application requirements. Some of the programming requires you to apply your learning of the concepts in different ways than presented. You may need to review the class notes or JavaDocs to complete this lab. Some suggestions will be provided. If you really get stuck, ask the TA.

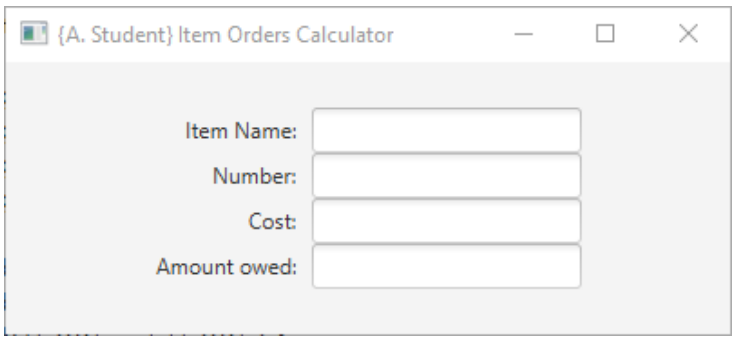
Part 1: Start simple labels and entry fields (2 Points)
NOTE: Answers to the questions for this exercise should be entered into a file called Exercise1.txt. Put the question # and the answer. You will be instructed when to submit this file.

Before creating the GUI shown below, answer these questions:

Q1: What layout (Pane) will you use?

Create a GUI class called **Orders** that contains right-aligned labels for text fields as shown. Place all of the components in an appropriate Pane, then add the Pane to the Scene. This is a common practice, easily allowing moving of the Pane to another Scene, should it be needed.

Use your name in place of "A. Student" in the title bar.



Q2: Write the full statement used to place the Pane.
HINT: If you use a GridPane, call GridPane.setHalignment(lblXX, HPos.RIGHT) for **each** of your labels - then they will be right aligned when in a GridPane.

Submit file Exercise1.txt with your answers to the Lab01 Assignment folder.

Part 2: Control Buttons (3 Points)
NOTE: Similar to Part1, put your answer into a file called Exercise2.txt.

Add buttons to control the actions that will take place on the screen. Your screen should now look similar to the one below.

The easy way to do this is to change the root Pane to a VBox. Create the GridPane from before, but call it something different from 'root' (e.g., gpTop).

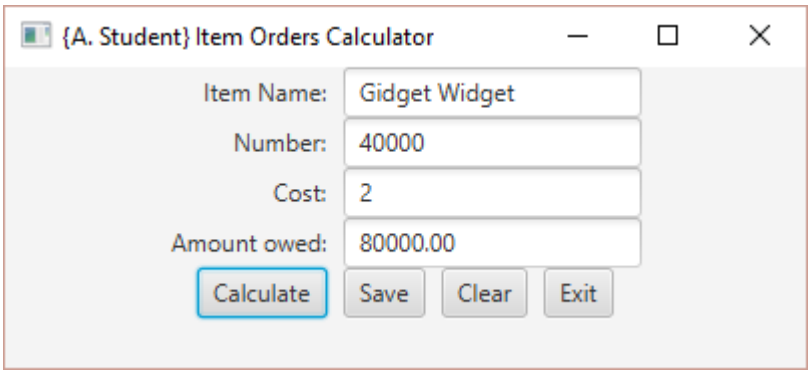
Also create a FlowPane (example fpBot) and place the buttons in it:

```
fpBot.getChildren().addAll(btn1, btn2 … );
```

Then add the GridPane and the FlowPane to the VBox:

```
root.getChildren().addAll(gpTop, fpBot);
```

- Change the "Amount owed" field to **not** allow input.
  Hint: Within Javadocs under TextField, search for the **inherited** setEditable method.



Q1: Name all layout managers used

Q2: In what area are the buttons placed?

Q3: In what class is setEditable defined?

Submit file Exercise2.txt with your answers to the Lab01 Assignment folder.

Part 3: Buttons control (5 Points)
Add controls for the buttons. Start with the easier ones first.

Exit	Exit the program
Clear	Clears the text fields. Can setText() to null or "" (blank).
Calculate	Multiply the "Number of" by the cost per item, place result in the Amount owed field. Format the amount owed result to have two decimal places. Hint: See String's "format" method.
Save	Open/append to a text file named **121Lab1.csv**. Write in comma-separated format, the item's name within quotes, the number of items, cost per item and the calculated amount owned. Each time the user clicks Save, first execute the Calculate code. The calculation code should only appear once in your program. Check the javadocs for constructors for FileOutputStream to see how to open a file for appending.

Open the CSV file to ensure it opens before submitting your code.

Submit your code (the .java file(s) only) to the Lab01 Assignment folder when Exercise 2 is working correctly.