College of Computer Science & Engineering

Department of Computer Science and Artificial Intelligence

جامعة جدة
University of Jeddah

CCCS214: Object-Oriented Programming II

Lab 6: Creating GUI Applications with JavaFX and Scene Builder

Lab Objectives

- Be able to create a basic JavaFX application with a closable window.
- Be able to create Label, TextField, and Button components.
- Be able to arrange controls in a layout container.
- Be able to construct a scene graph and set it to the stage.
- Be able to write an event handler that executes when a Button is clicked.

Introduction

In this lab you will create a tip calculator application with a JavaFX user interface. The application will let you enter the amount of a restaurant charge, and it will display the amount of a 20% tip. The completed user interface will look like this:
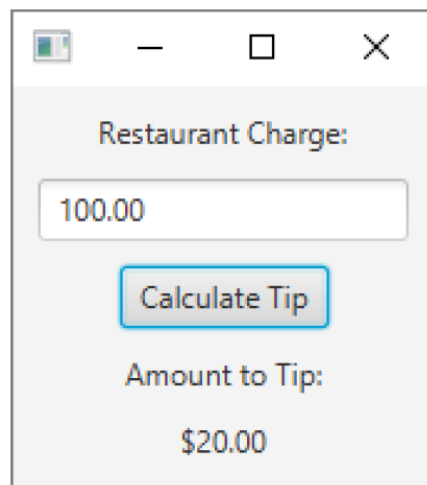
In your code, you will create a Scene containing the following controls:

- Various Label controls to display text.
- A TextField control to read the user's input.
- A Button control to calculate and display the tip.

Once you have created a GUI with Scene Builder, and saved it to an FXML file, you need to write a Java class that performs the following:

- Loads the FXML file

- Builds the scene graph in memory
- Displays the GUI

To do that:

1- Start a new Java source code file.
2- Write `import` statements for the following JavaFX classes:
   - `javafx.application.Application`
   - `javafx.stage.Stage`
3- Create a class named `TipCalculator` that inherits from the `Application` class.
4- Inside the `TipCalculator` class, write a `main` method that calls the `launch` method (which is inherited from the `Application` class).
5- Inside the `TipCalculator` class, write a `start` method that accepts a `Stage` object as an argument. The method should call the `Stage` object's `show()` method.
6- Save the file, and then compile and execute it.

The main application class is responsible for building the scene graph and displaying the GUI. The *controller class* is responsible for handling events that occur while the application is running. The controller class contains the following items:

- The necessary `import` statements.
- Private variables to reference the components that have an `fx:id` in the scene graph.
- An optional `initialize` method that is automatically called after the FXML file is loaded
- Event listener methods

Once you have written and compiled the controller class, you must go back into Scene Builder and register the controller class to the application's GUI.