# CMPT 381 Assignment 0:
# Hello World in JavaFX and Android

**Date due**: Friday, January 5, 2017, 11:59pm
**NOTE**: unlike future assignments, the submission system will ***not*** close at the due date – you may hand in the assignment anytime up to the due date of the next assignment (but you should do it as soon as you can)

## Overview

You will build 'hello world' GUIs in JavaFX and Android, in order to test your setup of the development environments, and test the Moodle handin system.

This assignment is to be done individually; each student will hand in an assignment.

## Requirements

Build 'hello world' GUI systems as specified below. Both systems will show a label and a button; when the button is pressed, the label will display the text "Hello 381!".

For JavaFX:

- Use the following tutorial as your starting point: http://docs.oracle.com/javase/8/javafx/get-started-tutorial/hello_world.htm
- Change the layout container so that you can add multiple interface components:
    - Replace the StackPane container used in the tutorial with a VBox (see https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/VBox.html for details)
- Add a label to the interface:
    - Declare a variable of type Label and pass a default text string into the constructor (see https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Label.html for details)
    - Add the label to the StackPane in the same way that the existing button is added
- Change the button's event handler to alter the label's text:
    - In the **handle()** method of the button's EventHandler (look for the existing line that prints "Hello World"), add a line of code that changes the text of the label, using the **setText()** method of the Label class

For Android:

- Use the following tutorial as your starting point: http://developer.android.com/training/basics/firstapp/index.html
- This tutorial will help you set up your environment and an Android Emulator
- Using the default project in Android Studio, go to the layout editor by clicking on the XML file controlling the layout (e.g., "activity_main.xml" if you created the project with default settings)
- Click on the TextView in the layout and add the ID ***label1*** in the attributes panel (top item)
- Add a Button to the layout by dragging a button from the list to the left of the layout
- Select the button and add the ID ***button1*** in the attributes panel
- Change the button's text attribute to "Press me"
- Change the button's onClick attribute to **handleClick**

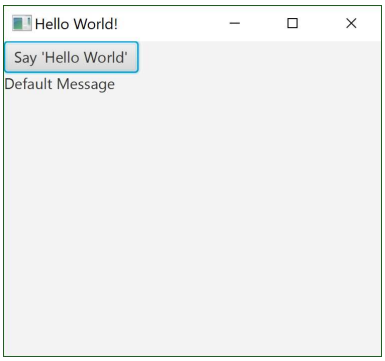
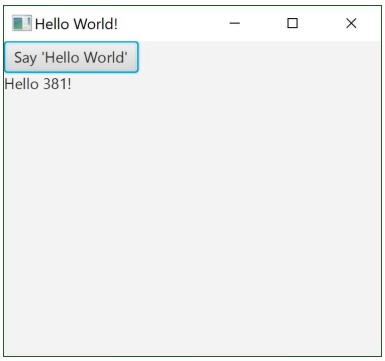
- In your main activity class (e.g., class MainActivity if you used the defaults, in file “MainActivity.java”), declare instance variables for the button (class Button) and the label (class TextView).
- In the onCreate() method of the file for your main activity, add the following lines to link the instance variables to the button and the label:

```
textView1 = (TextView) findViewById(R.id.textview1);
button1 = (Button) findViewById(R.id.button1);
```

- Add a new method to the class to handle the button click:

```
public void handleClick(View v) {
}
```

- Add code to the **handleClick()** method to change the text of the label when the button is pressed, using the **setText()** method of class TextView

| | | | |
|---|---|---|---|
|  |  |  |  |
| JavaFX before clicking | JavaFX after clicking | Android before clicking | Android after clicking |

# What to hand in

- JavaFX: hand in a zip file of your NetBeans project folder.
- Android: hand in a zip file of your Android Studio project folder
- A readme.txt file that indicates exactly what the marker needs to do to run your code. (Note that your systems for 381 should never require the marker to install external libraries or any new downloads).

# Where to hand in

Hand in your three files (two zips and one readme.txt) to the link on the course Moodle.

# Evaluation

Marks will be given for producing a system that meets the requirements above, and compiles and runs without errors. Note that unlike future assignments, late assignments ***will*** be allowed up to the deadline of the next assignment.