CSE 215: Programming Language II Lab
## Lab – 18
JavaFX UI Controls

Objective:
- To learn about Label, CheckBox, RadioButton, TextArea, ListView, ImageView

In the last two lab we saw the designing aspect and action performed based on an event of JavaFX. Today we will be covering on how to develop comprehensive user interface using UI controls provided by JavaFX.

- Button - A button is a control that triggers an action event when clicked. JavaFX provides regular buttons, toggle buttons, check box buttons, and radio buttons. The Labeled class defines the common properties for labels and buttons. A button is just like a label except that the button has the onAction property defined in the ButtonBase class, which sets a handler for handling a button's action.
- CheckBox - A CheckBox is used for the user to make a selection. When a check box is clicked (checked or unchecked), it fires an ActionEvent. To see if a check box is selected, use the isSelected() method.
- RadioButton - Radio buttons, also known as option buttons, enable the user to choose a single item from a group of choices. In appearance radio buttons resemble check boxes, but check boxes display a square that is either checked or blank, whereas radio buttons display a circle that is either filled (if selected) or blank (if not selected).
- TextArea - f you want to let the user enter multiple lines of text; you may create several instances of TextField. A better alternative, however, is to use TextArea, which enables the user to enter multiple lines of text.
- ComcoBox - A combo box, also known as a choice list or drop-down list, contains a list of items from which the user can choose. A combo box is useful for limiting a user's range of choices and avoids the cumbersome validation of data input.
- ListView - A list view is a control that basically performs the same function as a combo box, but it enables the user to choose a single value or multiple values.

Example:
Write a JavaFX program that will let user to select a nation from a combo box and then display a country's flag and its description in a text area. When user selects the provided radio Button or checkbox appropriate action should have to be performed.
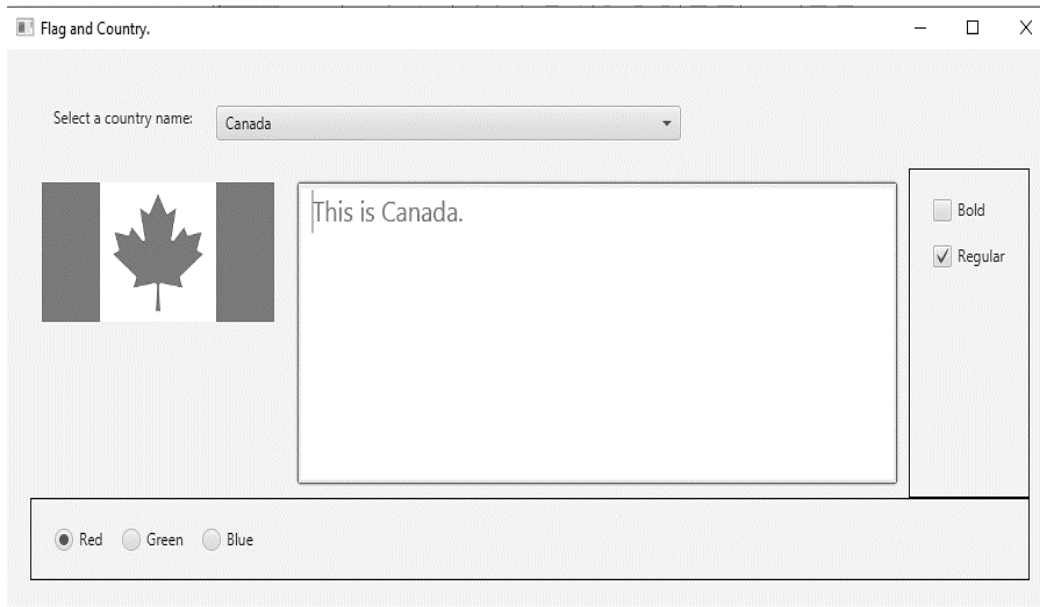
Code:

```
1   public class UIControl extends Application {
2     public static BorderPane bPane = new BorderPane();
3     public static String title = "Select a country name: ";
4     public static String[] item = { "Canada", "USA", "Bangladesh" };
5     public static int selectedItem = 0;
6     public       static       ImageView[]       imageofCountry       =       {       new
7   ImageView("file:\\C:\\Users\\Phantomhive\\Pictures\\739891.png"),
8         new ImageView("file:\\C:\\Users\\Phantomhive\\Pictures\\739891.png"),
9         new ImageView("file:\\C:\\Users\\Phantomhive\\Pictures\\739891.png") };
10    public static String[] countryDesList = { "This is Canada.", "This USA.", "This is
11  Bangladesh" };
12
13    public static Font boldFont = Font.font("Times New Roman", FontWeight.BOLD, 20);
14    public static Font italicFont = Font.font("Times New Roman", FontPosture.ITALIC,
15  20);
16    public static Font regularFont = Font.font("Times New Roman", FontPosture.REGULAR,
17  20);
18    public static Font boldAndItalicFont = Font.font("Times New Roman", FontWeight.BOLD,
19  FontPosture.ITALIC, 20);
20    public static TextArea textBody;
21
22    @Override
23    public void start(Stage primaryStage) {
24
25      bPane.setPadding(new Insets(20, 20, 20, 20));
26      setDisplay(selectedItem);
27      bPane.setTop(getTopofPane(title, item));
28      bPane.setRight(setPanelForCheckBox());
29      bPane.setBottom(setPanelForRadioButton());
30      BorderPane.setMargin(setPanelForRadioButton(), new Insets(10));
```

```java
31
32        Scene scene = new Scene(bPane, 900, 400);
33        primaryStage.setScene(scene);
34        primaryStage.setTitle("Flag and Country.");
35        primaryStage.show();
36
37    }
38
39    public static HBox setPanelForRadioButton() {
40        HBox panelForRadioButton = new HBox(16);
41
42        final ToggleGroup toggleGroupOfColorRadioButtons = new ToggleGroup();
43
44        RadioButton rbRed = new RadioButton("Red");
45        rbRed.setToggleGroup(toggleGroupOfColorRadioButtons);
46        RadioButton rbGreen = new RadioButton("Green");
47        rbGreen.setToggleGroup(toggleGroupOfColorRadioButtons);
48        RadioButton rbBlue = new RadioButton("Blue");
49        rbBlue.setToggleGroup(toggleGroupOfColorRadioButtons);
50
51        panelForRadioButton.setPadding(new Insets(20, 20, 20, 20));
52        panelForRadioButton.setStyle("-fx-border-color: black;");
53        panelForRadioButton.getChildren().addAll(rbRed, rbGreen, rbBlue);
54        EventHandler<ActionEvent> handler = e -> {
55          if (rbRed.isSelected()) {
56            textBody.setStyle("-fx-text-fill: red;");
57            bPane.setCenter(textBody);
58          } else if (rbGreen.isSelected()) {
59            textBody.setStyle("-fx-text-fill: green;");
60            bPane.setCenter(textBody);
61          } else if (rbBlue.isSelected()) {
62            textBody.setStyle("-fx-text-fill: blue;");
63            bPane.setCenter(textBody);
64          } else {
65            textBody.setStyle("-fx-text-fill: black;");
66            bPane.setCenter(textBody);
67          }
68        };
69
70        rbRed.setSelected(true);
71        rbRed.setOnAction(handler);
72        rbGreen.setOnAction(handler);
73        rbBlue.setOnAction(handler);
74
75        return panelForRadioButton;
76    }
77
78    public static VBox setPanelForCheckBox() {
79        VBox panelForCheckBox = new VBox(16);
80        panelForCheckBox.setPadding(new Insets(20, 20, 20, 20));
```

```java
81      panelForCheckBox.setStyle("-fx-border-color: black;");
82      CheckBox chkBold = new CheckBox("Bold");
83      CheckBox chkItalic = new CheckBox("Italic");
84      panelForCheckBox.getChildren().addAll(chkBold, chkItalic);
85
86      EventHandler<ActionEvent> handler = e -> {
87        if (chkBold.isSelected() && chkItalic.isSelected()) {
88          textBody.setFont(boldAndItalicFont);
89          bPane.setCenter(textBody);
90        } else if (chkItalic.isSelected()) {
91          textBody.setFont(italicFont);
92          bPane.setCenter(textBody);
93        } else if (chkBold.isSelected()) {
94          textBody.setFont(boldFont);
95          bPane.setCenter(textBody);
96        } else {
97          textBody.setFont(regularFont);
98          bPane.setCenter(textBody);
99        }
100     };
101
102     chkBold.setOnAction(handler);
103     chkItalic.setOnAction(handler);
104     return panelForCheckBox;
105   }
106
107   public static TextArea getTextArea(int index) {
108     TextArea taNote = new TextArea(countryDesList[selectedItem]);
109     taNote.setPrefColumnCount(20);
110     taNote.setPrefRowCount(5);
111     taNote.setWrapText(true);
112     taNote.setStyle("-fx-text-fill: red");
113     taNote.setFont(regularFont);
114     BorderPane.setMargin(taNote, new Insets(10));
115     return taNote;
116   }
117
118   public static void setDisplay(int index) {
119     ImageView selectedCountryImage = imageofCountry[selectedItem];
120     selectedCountryImage.setFitHeight(100);
121     selectedCountryImage.setFitWidth(200);
122     BorderPane.setMargin(selectedCountryImage, new Insets(10));
123     bPane.setLeft(selectedCountryImage);
124     textBody = getTextArea(selectedItem);
125     bPane.setCenter(textBody);
126   }
127
128   public static HBox getTopofPane(String title, String[] listItem) {
129
130     HBox horizontalBox = new HBox(16);
```

```
131        horizontalBox.setPadding(new Insets(20, 0, 20, 20));
132
133        Label titleLabel = new Label(title);
134
135        ComboBox<String> boxForItem = new ComboBox<>();
136
137        ObservableList<String> boxItem = FXCollections.observableArrayList(listItem);
138        boxForItem.getItems().addAll(boxItem);
139        boxForItem.setValue(listItem[0]);
140        boxForItem.setPrefWidth(400);
141        boxForItem.setOnAction(new EventHandler<ActionEvent>() {
142          @Override
143          public void handle(ActionEvent event) {
144            String value = boxForItem.getValue();
145            selectedItem = boxItem.indexOf(value);
146            setDisplay(selectedItem);
147          }
148        });
149
150        horizontalBox.getChildren().addAll(titleLabel, boxForItem);
151        return horizontalBox;
152    }
153
154    public static void main(String[] args) {
155        launch(args);
156    }
157 }
```

Home Task

Write a program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on. Only one light can be on at a time (see Figure). No light is on when the program starts.