The University of the West Indies, St. Augustine
COMP 2603 Object Oriented Programming 1

Practice 2 Week 7

This lab focuses on adding functionality to a GUI using Action Listeners.

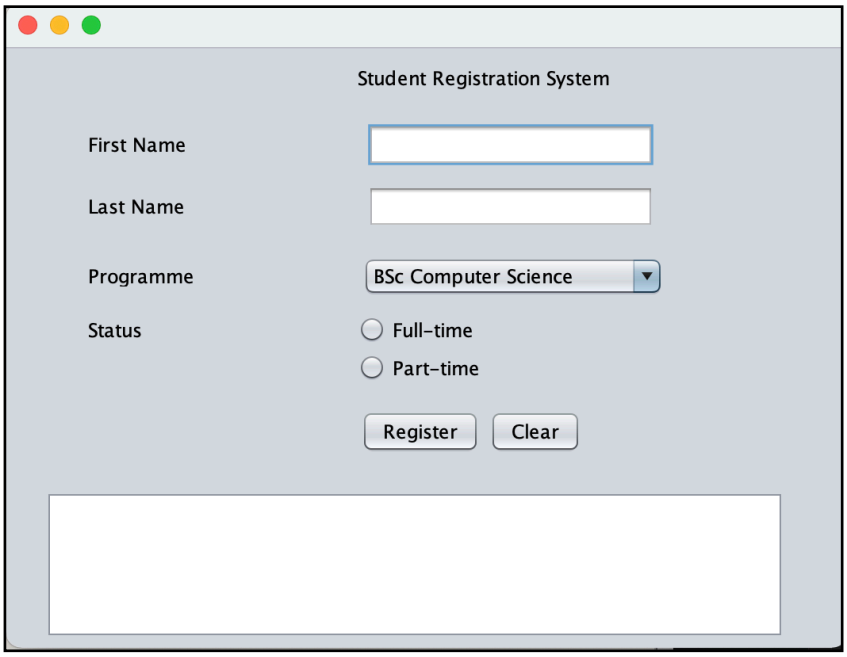

Figure 1

Download the Week7-GUI zip file from myElearning/temporary site. Extract and open the BlueJ project in your BlueJ editor. Run the StudentWindow class and observe the GUI as shown in Figure 1.

Part 1: GUI Elements

The StudentWindow class was created using the Matisse tool in Netbeans. Examine the StudentWindow class and observe the autogenerated code. The private init( ) method sets up the GUI components. Make the following changes in the init( ) method (ignore the warning message since it applies only if using the Netbeans editor):

1. Modify the labels First Name and Last Name to be Student First Name and Student Last Name.
2. Modify the existing programmes in the combo box list to be: BSc Computer Science (Special) and BSc Information Technology (Special)
3. Add two new programmes to the combo box list: BSc Computer Science (Major), BSc Information Technology (Major).
4. Set the default display value in the combo box to BSc Computer Science Major.
5. Use the ButtonGroup object (buttonGroup1) to group the radio buttons so that either the Full-time or the Part-time option is selected, but not both at once. Neither is selected by default.

Part 2: Simple Event Handling

Some GUI components are used to collect and display data (textfields, text area, combo box, radio button) and others respond to user initiated actions (buttons).

1. In the StudentWindow class, write a method called **clearData( )** that essentially clears all data in the textfields and text area, and resets the values in the combo box and radio buttons to their defaults
2. Create a new class **ClearButtonListener** that implements the **ActionListener** interface. An object reference to the **StudentWindow** must be passed into the ClearButtonListener class (e.g via the constructor). The **actionPerformed(..)** method must be overridden and it should invoke the **clearData( )** method.
3. In the StudentWindow class, associate the **clearButton** with an instance of the **ClearButtonListener** so that the button's functionality is implemented.

Part 3: Data Collection via the GUI

Four pieces of data should be collected when the user presses the Register button: first name, last name, programme and status.

1. In the StudentWindow class, write 5 methods that wrap the GUI component accessor/mutator methods:
    - public String getFirstName( )
    - public String getLastName( )
    - public String getProgramme( )
    - public String getStatus( )
    - public void setOutputText(String output)

    The first four methods return the data entered by the user on the GUI and the fifth one displays the data collected.
2. Create a new class **RegisterButtonListener** that implements the **ActionListener** interface. An object reference to the **StudentWindow** must be passed into the RegisterButtonListener class (e.g via the constructor). The **actionPerformed(..)** method must be overridden and it should invoke the **four** methods created in Step 1 to collect data, clear the existing data (use the clearData( ) method) and then display the data collected (fifth method from Step 1).
3. In the StudentWindow class, associate the **registerButton** with an instance of the **RegisterButtonListener** so that the button's functionality is implemented.

Additional Exercise

Make the **Register** button work as follows:

- Keep a record of the first and last names of every student registered using the GUI.
- Allow a maximum of 3 registrations
- Display the messages shown in the figures below when the Register button is clicked for the first time, when a student has been registered successfully, and when registration is full.
- The clear button should work as before (clearing data from all fields)