

## COMPUTER SCIENCE DEPARTMENT

**CS0053**

(PROGRAMMING TOOLS AND TECHNIQUES)

EXERCISE

2

GUI-Based Registration Module

|  |  |
| --- | --- |
| Name: **John Paul L. Besagas** | Professor: **Dr. Beau Habal** |
| Date Performed : **9/23/2024** | Date Submitted: **9/23/2024** |

1. **OBJECTIVES**

At the end of this exercise, students must be able to:

Cognitive

1. Understand the topics they have learned from lesson 2.

.

Psychomotor:

1. Apply coding convention and standard.
2. Create Java GUI-based program with Javadoc, comment, indention and naming convention.

Affective

1. Appreciate the concept behind this exercise.
2. **BACKGROUND INFORMATION**

In order to accomplish this exercise, the student must have a clear understanding of the following topics:

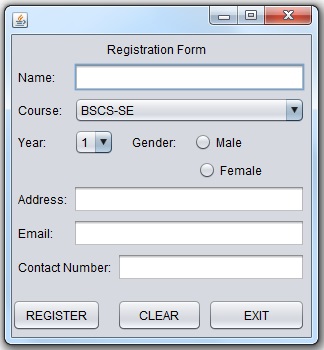
* Coding Convention and Standard
* Javadoc
* Import packages or libraries

1. **LABORATORY PROCEDURE**

1. Create a new program.

Program Name: Registration.java

2. Design your layout as shown below



3. Requirements

- Must validate input.

* Apply coding conventions.
* The program must be free from any errors.

|  |
| --- |
| import javax.swing.\*;  import java.awt.\*;  /\*\*   \* The Registration class creates a GUI for the user registration form.   \* With a simple GUI, the registration form also can validate user input   \* and ensure that theres no missing field before register   \*/  public class Registration {      JFrame frameRegistration;      JTextField fieldDisplayName, fieldDisplayAddress, fieldDisplayEmail, fieldDisplayContact;      JComboBox<String> fieldCourse, fieldYear;      JRadioButton radioMale, radioFemale;      JButton buttonRegister, buttonClear, buttonExit;      ButtonGroup buttonGender;      Font fontDisplay = new Font("Arial", Font.PLAIN, 14);      public Registration() {          initializeFrame();      }      /\*\*       \* Initializes the JFrame and GUI components for the registration form.       \*/      public void initializeFrame() {          frameRegistration = new JFrame("Registration Form");          frameRegistration.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);          frameRegistration.setSize(400, 400);          frameRegistration.getContentPane().setBackground(Color.DARK\_GRAY);          frameRegistration.setLayout(null);          JLabel labelName = new JLabel("Name:");          labelName.setBounds(30, 30, 100, 30);          labelName.setForeground(Color.WHITE);          labelName.setFont(fontDisplay);          frameRegistration.add(labelName);          fieldDisplayName = new JTextField();          fieldDisplayName.setBounds(140, 30, 200, 30);          fieldDisplayName.setBackground(Color.GRAY);          fieldDisplayName.setForeground(Color.WHITE);          fieldDisplayName.setFont(fontDisplay);          fieldDisplayName.setBorder(BorderFactory.createEmptyBorder());          frameRegistration.add(fieldDisplayName);          JLabel labelCourse = new JLabel("Course:");          labelCourse.setBounds(30, 70, 100, 30);          labelCourse.setForeground(Color.WHITE);          labelCourse.setFont(fontDisplay);          frameRegistration.add(labelCourse);          String[] courses = { "BSCSSE", "BSCSDS", "BSCSAI", "BSIT" };          fieldCourse = new JComboBox<>(courses);          fieldCourse.setBounds(140, 70, 200, 30);          fieldCourse.setBackground(Color.GRAY);          fieldCourse.setForeground(Color.WHITE);          fieldCourse.setFont(fontDisplay);          fieldCourse.setBorder(BorderFactory.createEmptyBorder());          frameRegistration.add(fieldCourse);          JLabel labelYear = new JLabel("Year:");          labelYear.setBounds(30, 110, 100, 30);          labelYear.setForeground(Color.WHITE);          labelYear.setFont(fontDisplay);          frameRegistration.add(labelYear);          String[] years = { "1st", "2nd", "3rd", "4th" };          fieldYear = new JComboBox<>(years);          fieldYear.setBounds(140, 110, 200, 30);          fieldYear.setBackground(Color.GRAY);          fieldYear.setForeground(Color.WHITE);          fieldYear.setFont(fontDisplay);          fieldYear.setBorder(BorderFactory.createEmptyBorder());          frameRegistration.add(fieldYear);          JLabel labelGender = new JLabel("Gender:");          labelGender.setBounds(30, 150, 100, 30);          labelGender.setForeground(Color.WHITE);          labelGender.setFont(fontDisplay);          frameRegistration.add(labelGender);          radioMale = new JRadioButton("Male");          radioMale.setBounds(140, 150, 80, 30);          radioMale.setBackground(Color.DARK\_GRAY);          radioMale.setForeground(Color.WHITE);          radioMale.setFont(fontDisplay);          radioFemale = new JRadioButton("Female");          radioFemale.setBounds(230, 150, 80, 30);          radioFemale.setBackground(Color.DARK\_GRAY);          radioFemale.setForeground(Color.WHITE);          radioFemale.setFont(fontDisplay);          buttonGender = new ButtonGroup();          buttonGender.add(radioMale);          buttonGender.add(radioFemale);          frameRegistration.add(radioMale);          frameRegistration.add(radioFemale);          JLabel labelAddress = new JLabel("Address:");          labelAddress.setBounds(30, 190, 100, 30);          labelAddress.setForeground(Color.WHITE);          labelAddress.setFont(fontDisplay);          frameRegistration.add(labelAddress);          fieldDisplayAddress = new JTextField();          fieldDisplayAddress.setBounds(140, 190, 200, 30);          fieldDisplayAddress.setBackground(Color.GRAY);          fieldDisplayAddress.setForeground(Color.WHITE);          fieldDisplayAddress.setFont(fontDisplay);          fieldDisplayAddress.setBorder(BorderFactory.createEmptyBorder());          frameRegistration.add(fieldDisplayAddress);          JLabel labelEmail = new JLabel("Email:");          labelEmail.setBounds(30, 230, 100, 30);          labelEmail.setForeground(Color.WHITE);          labelEmail.setFont(fontDisplay);          frameRegistration.add(labelEmail);          fieldDisplayEmail = new JTextField();          fieldDisplayEmail.setBounds(140, 230, 200, 30);          fieldDisplayEmail.setBackground(Color.GRAY);          fieldDisplayEmail.setForeground(Color.WHITE);          fieldDisplayEmail.setFont(fontDisplay);          fieldDisplayEmail.setBorder(BorderFactory.createEmptyBorder());          frameRegistration.add(fieldDisplayEmail);          JLabel labelContact = new JLabel("Contact Number:");          labelContact.setBounds(30, 270, 120, 30);          labelContact.setForeground(Color.WHITE);          labelContact.setFont(fontDisplay);          frameRegistration.add(labelContact);          fieldDisplayContact = new JTextField();          fieldDisplayContact.setBounds(140, 270, 200, 30);          fieldDisplayContact.setBackground(Color.GRAY);          fieldDisplayContact.setForeground(Color.WHITE);          fieldDisplayContact.setFont(fontDisplay);          fieldDisplayContact.setBorder(BorderFactory.createEmptyBorder());          frameRegistration.add(fieldDisplayContact);          buttonRegister = new JButton("REGISTER");          buttonRegister.setBounds(50, 320, 110, 30);          buttonRegister.setBackground(Color.BLUE);          buttonRegister.setForeground(Color.WHITE);          buttonRegister.setFont(fontDisplay);          frameRegistration.add(buttonRegister);          buttonClear = new JButton("CLEAR");          buttonClear.setBounds(160, 320, 110, 30);          buttonClear.setBackground(Color.BLUE);          buttonClear.setForeground(Color.WHITE);          buttonClear.setFont(fontDisplay);          frameRegistration.add(buttonClear);          buttonExit = new JButton("EXIT");          buttonExit.setBounds(270, 320, 100, 30);          buttonExit.setBackground(Color.BLUE);          buttonExit.setForeground(Color.WHITE);          buttonExit.setFont(fontDisplay);          frameRegistration.add(buttonExit);          buttonExit.addActionListener(e -> System.exit(0));          buttonClear.addActionListener(e -> clearFields());          buttonRegister.addActionListener(e -> registerUser());          frameRegistration.setVisible(true);      }      /\*\*       \* Clears all input fields in the registration form.       \*/      public void clearFields() {          fieldDisplayName.setText("");          fieldCourse.setSelectedIndex(0);          fieldYear.setSelectedIndex(0);          buttonGender.clearSelection();          fieldDisplayAddress.setText("");          fieldDisplayEmail.setText("");          fieldDisplayContact.setText("");      }      /\*\*       \* Handles the registration process by validating inputs and displaying the       \* entered registration details in a new window.       \*/      public void registerUser() {          String name = fieldDisplayName.getText();          String course = (String) fieldCourse.getSelectedItem();          String year = (String) fieldYear.getSelectedItem();          String address = fieldDisplayAddress.getText();          String email = fieldDisplayEmail.getText();          String contact = fieldDisplayContact.getText();          String gender = "";          if (radioMale.isSelected()) {              gender = "Male";          } else if (radioFemale.isSelected()) {              gender = "Female";          }          // Check if there are no missing fields          if (name.isEmpty() || course.isEmpty() || year.isEmpty() || gender.isEmpty() || address.isEmpty()                  || email.isEmpty() || contact.isEmpty()) {              JOptionPane.showMessageDialog(frameRegistration, "Please, fill out all the details", "Error",                      JOptionPane.ERROR\_MESSAGE);              return;          }          // Check if the name contains any numbers or special characters          if (!name.matches("[a-zA-Z\\s]+")) {              JOptionPane.showMessageDialog(frameRegistration, "Name should only contain letters", "Error",                      JOptionPane.ERROR\_MESSAGE);              return;          }          // Check contact number if its a number          if (!contact.matches("\\d+")) {              JOptionPane.showMessageDialog(frameRegistration, "Contact number must be a number", "Error",                      JOptionPane.ERROR\_MESSAGE);              return;          }          JFrame frameDetails = new JFrame("Registration Details");          frameDetails.setSize(300, 400);          frameDetails.getContentPane().setBackground(Color.DARK\_GRAY);          frameDetails.setLayout(null);          JLabel labelDetails = new JLabel("Registration Details:");          labelDetails.setBounds(50, 10, 200, 30);          labelDetails.setForeground(Color.WHITE);          frameDetails.add(labelDetails);          JLabel labelNameDetails = new JLabel("Name: " + name);          labelNameDetails.setBounds(50, 50, 200, 30);          labelNameDetails.setForeground(Color.WHITE);          frameDetails.add(labelNameDetails);          JLabel labelCourseDetails = new JLabel("Course: " + course);          labelCourseDetails.setBounds(50, 90, 200, 30);          labelCourseDetails.setForeground(Color.WHITE);          frameDetails.add(labelCourseDetails);          JLabel labelYearDetails = new JLabel("Year: " + year);          labelYearDetails.setBounds(50, 130, 200, 30);          labelYearDetails.setForeground(Color.WHITE);          frameDetails.add(labelYearDetails);          JLabel labelGenderDetails = new JLabel("Gender: " + gender);          labelGenderDetails.setBounds(50, 170, 200, 30);          labelGenderDetails.setForeground(Color.WHITE);          frameDetails.add(labelGenderDetails);          JLabel labelAddressDetails = new JLabel("Address: " + address);          labelAddressDetails.setBounds(50, 210, 200, 30);          labelAddressDetails.setForeground(Color.WHITE);          frameDetails.add(labelAddressDetails);          JLabel labelEmailDetails = new JLabel("Email: " + email);          labelEmailDetails.setBounds(50, 250, 200, 30);          labelEmailDetails.setForeground(Color.WHITE);          frameDetails.add(labelEmailDetails);          JLabel labelContactDetails = new JLabel("Contact: " + contact);          labelContactDetails.setBounds(50, 290, 200, 30);          labelContactDetails.setForeground(Color.WHITE);          frameDetails.add(labelContactDetails);          frameDetails.setVisible(true);      }      public static void main(String[] args) {          new Registration();      }  } |

1. **QUESTION AND ANSWER**
2. What are the coding conventions you applied in your program?

**I followed the coding convention of using capitalized nouns for class names, verbs in camelCase for method names, and short, meaningful camelCase variable names. I also included comments for self-documentation to improve code understanding and readability.**

1. How do you able to use the Javadoc?

**I used Javadoc by adding comments above some methods and class, it's not all because some are self-explanatory. These comments started with /\*\* and ended with \*/. Within the Javadoc comments, I provided the description of the class and method.**

1. What part of your program used comments?

**In my program, I used it to describe the Registration class and some methods in my code with the use of Javadoc comments meanwhile I used the inline comment for the block of code that check the validity of the user's input.**

1. **ASSESSMENT**

|  |  |
| --- | --- |
| Department | Computer Science |
| Subject Code | CSSSPEC2 |
| Description | Programming Tools and Techniques |
| Term/Academic Year | 1st Term SY 2016-2017 |

|  |  |
| --- | --- |
| Topic | Coding Convention and Standards |
| Lab Activity No | 2 |
| Lab Activity | **GUI-Based Registration Module** |
| CLO | **1, 2** |

**Note: The following rubrics/metrics will be used to grade students’ output in the lab exercise 2.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Exceptional | Acceptable | Amateur | Unsatisfactory |
| Specifications  (40%) | The program works and meets all of the specifications. (40) | The program works and produces the correct results and displays them correctly. It also meets most of the other specifications. (35-39) | The program produces correct results but does not display them correctly. (30-34) | The program is producing incorrect results. (20-29) |
| Design  (15 %) | The design is exceptionally attractive. Program is "user-friendly" with informative and consistent prompts and messages. (15) | The design is fairly attractive. Program is "user-friendly" with informative and consistent prompts and messages. (13-14) | The design is fairly attractive. Program is not "user-friendly" but still provide informative and consistent prompts and messages. (10-12) | The design is unattractive and not user-friendly (8-9) |
| Efficiency (20%) | The code is extremely efficient without sacrificing readability and understanding. (20) | The code is fairly efficient without sacrificing readability and understanding. (17-19) | The code is brute force and unnecessarily long. (14-16) | The code is huge and appears to be patched together. (10-13) |
| Readability  (10 %) | The code is exceptionally well organized and very easy to follow. (10) | The code is fairly easy to read. (8-9) | The code is readable only by someone who knows what it is supposed to be doing. (6-7) | The code is poorly organized and very difficult to read. (4-5) |
| Delivery  (15%) | The program was delivered on time. (15) | The program was delivered within a day of the due date. (13-14) | The code was within 2 days of the due date. (10-12) | The code was within a week of the due date. (8-9) |
| Total: 100% |  |  |  |  |