



Ahsanullah University of Science and Technology (AUST)
Department of Computer Science and Engineering

LABORATORY MANUAL

Course No.: CSE 2100
Course Title: Software Development-II

For the students of 2nd Year, 1st semester of
B.Sc. in Computer Science and Engineering program

TABLE OF CONTENTS

| | |
|---|-----------|
| COURSE OBJECTIVES..... | 1 |
| PREFERRED TOOLS..... | 1 |
| TEXT/REFERENCE BOOK..... | 1 |
| ADMINISTRATIVE POLICY OF THE LABORATORY | 1 |
| LIST OF SESSIONS | |
| SESSION 1: | 2 |
| Introduction to Java Collection. | |
| SESSION 2: | 3 |
| Introduction to Java Swing. | |
| SESSION 3: | 5 |
| Introduction to File Read and Write in java and graphics in java swing. | |
| SESSION 4: | 8 |
| A brief introduction of Java Networking. | |
| SESSION 5 & 6: | 9 |
| Introduction to JavaFx. | |
| SESSION 7: | 11 |
| Introduction to Java MultiThreading. | |
| FINAL PROJECT SUBMISSION | 12 |

COURSE OBJECTIVES

- The objective of this laboratory course is to give students experience designing and implementing large program using different tools and framework of Java language. The main outcome of this laboratory work is to design interactive Graphical User Interface (GUI) using Java Swing and JavaFX. Students will also get some advanced Java programming experiences; covering topics such as—game developing tools like 2d graphics, multithreading, networking, advanced data structure, and database programming. At the end of the course student will develop a software project in a group or individually using Java.

PREFERRED TOOL(S)

1. NetBeans.
2. Eclipse
3. Java development kit

TEXT/REFERENCE BOOK(S)

1. “The Complete Reference” written by “Herbert Schildt”. 7th Edition.
2. “Java How to program” written by “Harvey Deitel”. 3rd Edition

ADMINISTRATIVE POLICY OF THE LABORATORY

1. Students must perform class assignments given by the corresponding teacher individually without help of others.
2. Student are asked to do some home assignments and each student is interviewed individually to restrict Plagiarism.
3. Student have to form a group of maximum 3 members and the group presents a project proposal at the beginning of a semester. The group should develop the proposed project within the entire period of a semester; at the later part of a semester, the group has to present the project. Each student of the group is individually evaluated.
4. Plagiarism is strictly forbidden and will be dealt with punishment.

Session 1

OBJECTIVES:

The objective of this problem is to give knowledge about Java Collection Framework.

PROBLEM DESCRIPTION:

Create Student class with attributes name and ID, add the student object in the arraylist, traverse the list and sort the list in ascending order of ID.

STEPS:

1. Create a Student object with attributes name and ID.
2. Insert a student object in ArrayList using add().
3. Traverse the list and print each attributes of all student objects.
4. Sort the list according to the ascending order of ID.

Sample Input/Output:

```
Rahim 101
Karim 100
John 105

Output of task # 3:
Rahim 101
Karim 100
```

Session 2

PROBLEM 1:

OBJECTIVES: The objective of this problem is to create a simple Graphical User Interface using Java Swing Components.

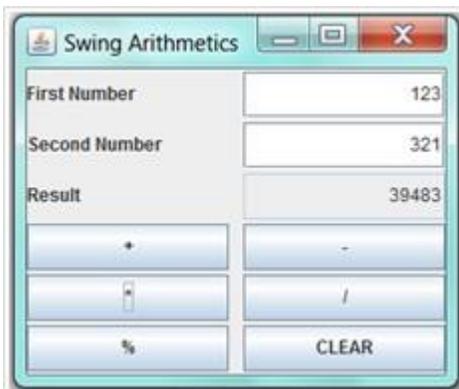
PROBLEM DESCRIPTION:

Write a Java program called SwingArithmetics that works as a simple calculator. Use default layout to arrange buttons for the digits and for the +, -, *, % and clear operations. Add a text field to display the result. Handle any possible exceptions such as divided by zero.

STEPS:

1. Create frame using JFrame()
2. Create buttons using JButton() and text field using JTextField()
3. Place the buttons and textfield in the frame by using add()
4. Handle possible exceptions and show message in dialog box using JOptionPane()

SAMPLE INPUT/OUTPUT:



PROBLEM 2:

OBJECTIVES: The objective of this problem is to create a student registration form using different swing components.

PROBLEM DESCRIPTION:

Create a Java program that will work as a simple student registration form. Students need to use various swing components like JMenu, JButton, JRadioButton, JComboBox, JTable, JPasswordField, JTextField, JLabel to design the frames. Information about student entered in the student registration form must be displayed in the JTable.

STEPS:

1. Design the following Interface (see below sample Input/Output) as student registration form.
2. Clicking on the calculate grade must able to compute the grade and show in Grade textField.
3. Clicking the show button information of student will display in the table.

Sample Input/Output:

The screenshot shows a Java Swing window titled "Student Registration Form". The window is divided into two main sections: "Student Information" and "Student grade".

Student Information:

- Title: A dropdown menu with "Mr." selected.
- First Name: A text field.
- Middle Name: A text field.
- Last Name: A text field.
- Gender: Two radio buttons, "Male" and "Female".
- Department: A group of checkboxes for "CSE", "CE", "Textile", "Arch", "BBA", "ME", and "NPE".
- Year: A dropdown menu with "1st" selected.
- Semester: A dropdown menu with "1st" selected.

Student grade:

- Course No: A list box containing "CSE2100", "CSE 205", and "CSE204".
- Total Marks: A text field.
- Grade: A text field.
- Calculate Grade: A button.

Below the "Student grade" section, there is a "Show" button and a JTable with the following columns: Name, Gender, Department, Semester, Course No, and Grade. The table is currently empty.

Session 3

PROBLEM 1:

OBJECTIVES: The objective of this session is to provide in depth knowledge about Java Swing components specially JMenu, Radio button, ComboBox, JTable and event handling in Swing components. The brief introduction of file reading and writing utilities are also given in the session.

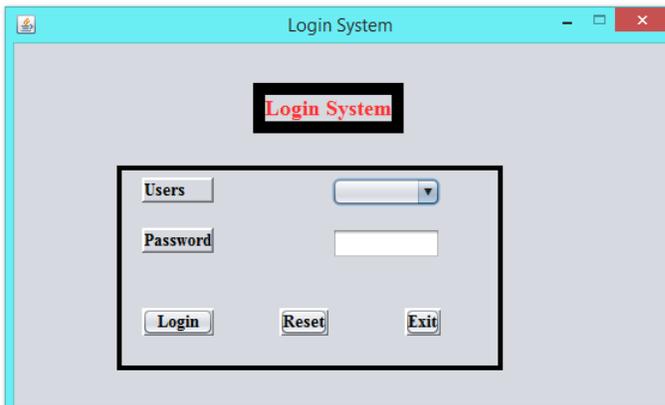
PROBLEM DESCRIPTION:

Create a Java program that will work as a simple employee management system where admin can login into the system and manage the employee information. The system has two frames one is login for Admin and other is to add/edit/delete employees' information. Admin can access the employee information frame if he/she is authenticated. Students need to use various swing components like JMenu, JButton, JRadioButton, JComboBox, JTable, JPasswordField, JTextField, JLabel to design the frames. Information about employee must be displayed in the JTable. Contents of the JTable need to be saved in the .txt file and can load the contents to the JTable from the .txt file.

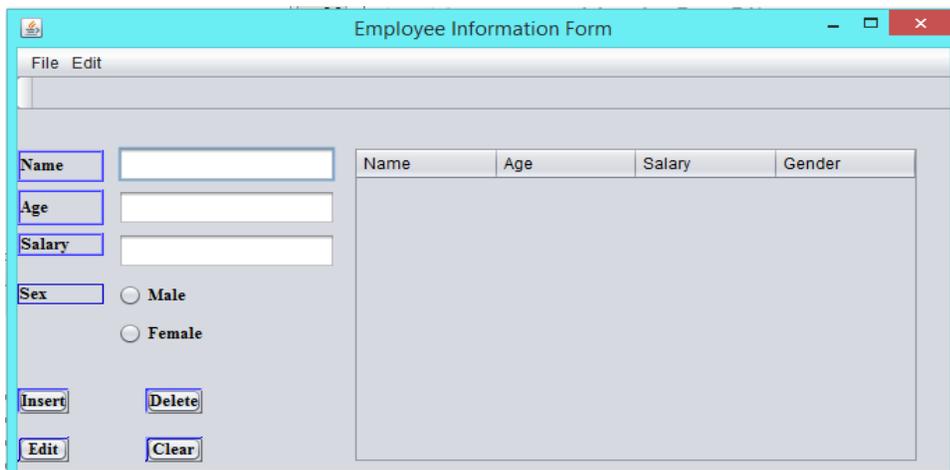
STEPS:

1. Design the following Interface (see input/output) as login system.
2. Use JComboBox to select the users.
3. Use JPasswordField to input the password.
4. Clicking the login button the following EmployeeInfo frame will open if the admin password is matched. Create object of EmployeeInfo class.
5. Create object of DefaultTableModel class to access the contents of the table.
6. Click save menu item to save the contents of the table to .txt file. Use FileWriter and BufferedWriter class to write the contents of the table to the file.
7. Click open menu item to load the contents of the .txt file to the table.

SAMPLE INPUT/OUTPUT:

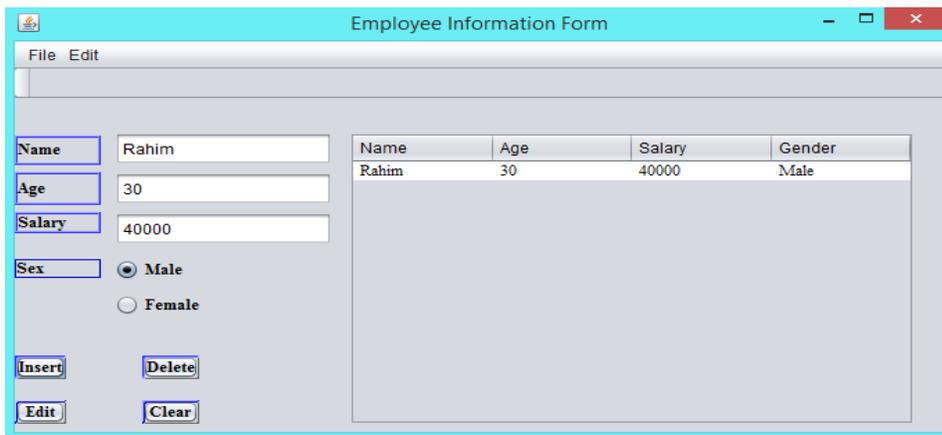


A screenshot of a window titled "Login System". The window has a light blue title bar with standard Windows window controls. The main content area is light gray and features a central black-bordered box. Inside this box, there is a "Users" label above a dropdown menu, a "Password" label above a text input field, and three buttons labeled "Login", "Reset", and "Exit" arranged horizontally at the bottom.



A screenshot of a window titled "Employee Information Form". The window has a light blue title bar and a menu bar with "File" and "Edit". The main area is divided into two sections. On the left, there are input fields for "Name", "Age", and "Salary", and radio buttons for "Sex" (Male and Female). Below these are buttons for "Insert", "Delete", "Edit", and "Clear". On the right, there is a table with the following structure:

| Name | Age | Salary | Gender |
|------|-----|--------|--------|
|------|-----|--------|--------|



A screenshot of the same "Employee Information Form" window, but now with data entered. The input fields on the left contain "Rahim", "30", and "40000". The "Male" radio button is selected. The table on the right now contains one row of data:

| Name | Age | Salary | Gender |
|-------|-----|--------|--------|
| Rahim | 30 | 40000 | Male |

PROBLEM 2:

OBJECTIVES: A brief introduction to geometric shapes for example line, rectangle etc using Java 2D Graphics.

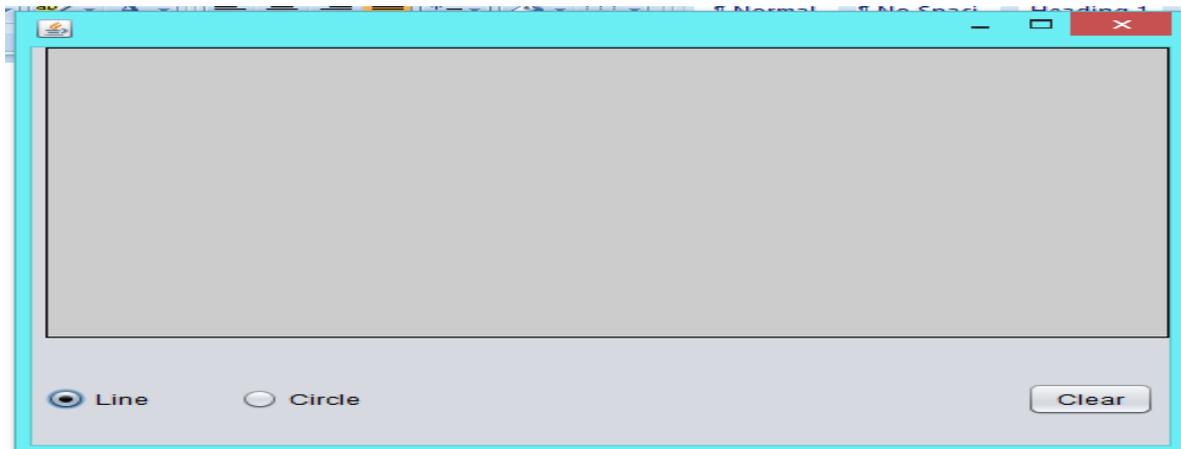
PROBLEM DESCRIPTION:

Create a java program that will work as simple paint application using java Graphics2D class. The program use mouse dragging event to draw line and circle on the JPanel.

STEPS:

1. Design the following interface for simple paint application:
2. When line radio button is selected then line will be drawn by dragging the mouse.
3. Create object of Graphics2D class and use getGraphics() method to get graphics context for JPanel.
4. Use mousePressed(), mouseReleased() and mouseDragging() events for free hand drawing.
5. Repeat the steps 3 and 4 when circle radio button is selected for drawing circle on the JPanel.

SAMPLE INPUT/OUTPUT:



Session 4

OBJECTIVES: A brief introduction of Java Networking: connection-oriented socket programming.

PROBLEM DESCRIPTION:

Write a simple program where there will be two java programs work as a client and a server. The client will send a simple text and the server have to receive the text and print the text.

STEPS:

Server Part

1. Create a class that represents a server.
2. Use ServerSocket class to create an instance of ServerSocket
3. Create an object of DataInputStream class
4. The Object will wait until the message from a client is arrived.

Client part

1. Create a class the represents a client.
2. Create an object of Socket class that establishes connection with the server.
3. Create an instance of DataOutputStream class to send the data.
4. Use the object of DataOutputStream to send a string

SAMPLE INPUT/OUTPUT:

```
Server Side  
Message from a client: Hello Server
```

Session 5 & Session 6

PROBLEM 1:

OBJECTIVES: Create a simple JavaFx program that will display a circle in GUI.

PROBLEM DESCRIPTION:

Create a simple JavaFx program that will display a circle in GUI.

STEPS:

1. **Creating a Class:** Create a Java class and inherit the Application class of the package javafx.application and implement the start() method
2. **Creating a Circle:** You can create a circle in JavaFX by instantiating the class named Circle which belongs to a package javafx.scene.shape
3. **Setting Properties to the Circle:**
4. **Creating a pane to hold the circle:**
5. **Creating a Scene Object:** Create a Scene by instantiating the class named Scene which belongs to the package javafx.scene. To this class, pass the Pane object (root) that was created in the previous step. In addition to the root object, you can also pass two double parameters representing height and width of the screen along with the object of the Group class as follows.
6. **Setting the Title of the Stage:** You can set the title to the stage using the setTitle() method of the Stage class.
7. **Adding Scene to the Stage:** You can add a Scene object to the stage using the method setScene() of the class named Stage.
8. **Displaying the Contents of the Stage:** Display the contents of the scene using the method named show() of the Stage class
9. **Launching the Application:** Launch the JavaFX application by calling the static method launch() of the Application class from the main method

PROBLEM 2:

OBJECTIVES: Event handling and scene changing program using JavaFx.

PROBLEM DESCRIPTION:

Create a JavaFx application that will change the scene by clicking button. Your program will have two scenes: scene 1 and scene 2. Suppose scene 1 is default scene. By clicking button at scene 1, scene 2 will appear. And by clicking button at scene 2, the program will go back to scene 1.

STEPS:

Create first scene with two objects label and button

1. Create label: `Label label = new Label("Welcome to the Scene 1");`
2. Create button1:
`Button btn1 = new Button("Go to Scene 2");`
Use lambda expression to set the event on button1:
`btn1.setOnAction(e->window.setScene(scene2));`
3. Use VBox layout to add the label and button on scene 1
`VBox layout1 = new VBox(20); //In VBox layout pane nodes are arranged in a single column`
4. Use `addAll()` method to add the label and button to the layout1.
`layout1.getChildren().addAll(label, btn1);`
5. Add layout1 to the scene 1
`scene1 = new Scene(layout1, 200,200);`

Create scene 2 with one object

6. `Button btn2 = new Button("Go back to scene 1");`
`btn2.setOnAction(e->window.setScene(scene1));`
7. Use StackPane layout to add button 2
`StackPane layout2 = new StackPane();`
`layout2.getChildren().add(btn2);`
8. Add layout 2 to the scene 2

Default scene

9. Set scene 1 as default scene - `window.setScene(scene1);`
10. Display the window - `window.show();`

Session 7

PROBLEM 1:

OBJECTIVES: The objective of this lab is to introduce to multithreading.

PROBLEM DESCRIPTION:

Write a simple multithreaded program so that the program can create three threads. Each thread prints 1 to 5 after its name; and each thread waits for other threads to complete. After completion of all three thread, the main thread will terminate.

STEPS:

1. Create a class that implements Runnable interface
2. Implement the run() method appropriately.
3. Create a class that contain main method.
4. Inside the main method, three thread will be created.
5. Use an appropriate method so the threads will waits for other thread to finish.

SAMPLE INPUT/OUTPUT:

```
New thread: Thread[One,5,main]
New thread: Thread[Two,5,main]
New thread: Thread[Three,5,main]
Thread One is alive: true

Thread Two is alive: true
Thread Three is alive: true
Waiting for threads to finish.
One: 5

Three: 5

Two: 5

One: 4

Three: 4

Two: 4

Three: 3

Two: 3
```

FINAL Project Submission

Students develop their projects in a group consists of three/two members. There will be a viva and Q/A session after project submission.