

Report On

Title of the Course Project

Submitted in partial fulfillment of the requirements of the Course project in
Semester III of Second Year Artificial Intelligence and Data Science

By
Shreeya Hudekar (Roll No. 15)
Charmi Jani (Roll No. 17)
Tejashree Karekar (Roll No. 20)

Supervisor
Prof. Sneha.M.Yadav



University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science



(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the project entitled “Currency Converter” is a bonafide work of Shreeya Hudekar (Roll No. 15), Charmi Jani (Roll No. 17), Tejashree Karekar (Roll No. 20), submitted to the University of Mumbai in partial fulfillment of the requirement for the **Course project in semester III of Second Year** Artificial Intelligence and Data Science engineering.

Supervisor

Prof. Sneha.M.Yadav

Dr. Tatwadarshi P. N.
Head of Department

Table of Contents

Pg. No

Chapter No		Title	Page No.
1		OVERVIEW	1
2		PROGRAM AND OUTPUT	2
3		EXPLANATION	7

1 Overview:

Overview of the Java Currency Converter Program:

The Java Currency Converter program is a simple graphical application that allows users to convert currency values between Indian Rupees (INR) and US Dollars (USD) using a Java Swing-based Graphical User Interface (GUI). This program is designed to provide a straightforward means of performing currency conversions with real-time results. Here's an overview of the key aspects of the code and program:

1. Importing Required Libraries:

- The program begins by importing the necessary Java Swing libraries for creating a GUI.

2. `GFG` Class:

- The `GFG` class contains the entire code for the currency converter.

3. `converter` Method:

- This method is the heart of the program and is responsible for creating the GUI and implementing the currency conversion logic.

4. GUI Components:

- Within the `converter` method, various GUI components are created:
 - `JFrame`: A main application window.
 - Labels for "Rupees" and "Dollars."
 - Text fields for entering the amount in rupees and dollars.
 - Buttons for converting from rupees to dollars, from dollars to rupees, and for closing the program.

5. Positioning GUI Components:

- The `setBounds` method is used to define the position and size of each GUI component on the frame.

6. Action Listeners:

- Action listeners are added to the conversion buttons ("INR" and "Dollar"). These listeners respond to button clicks and perform currency conversion based on a specified exchange rate. There is also an action listener for the "close" button to exit the application.

7. Conversion Logic:

- The "INR" button converts an amount from rupees to dollars using a predefined conversion rate.
- The "Dollar" button converts an amount from dollars to rupees using the same rate. The converted values are displayed in real-time in the respective text fields.

8. Frame Layout and Visibility:

- The frame's layout is set to `null`, allowing manual positioning of components.
- The frame's size is set to 400x300 pixels.
- The frame is set to be visible, making the GUI accessible to users.

9. `main` Method:

- The `main` method serves as the entry point of the program. It calls the `converter` method to create and display the currency converter GUI.

In summary, this Java program exemplifies a simple currency conversion tool, offering users a convenient way to convert between Indian Rupees and US Dollars via a user-friendly graphical interface. It demonstrates the usage of Java Swing for creating GUI applications, event handling with action listeners, and basic currency conversion calculations. Users can input currency values and immediately see the converted results.

2 Program and Output:

```
// Java program to convert from  
// rupee to the dollar and vice-versa  
// using Java Swing
```

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
public class GFG {  
  
    // Function to convert from rupee  
    // to the dollar and vice-versa  
    // using Java Swing  
    public static void converter()  
    {  
  
        // Creating a new frame using JFrame  
        JFrame f = new JFrame("CONVERTER");  
  
        // Creating two labels  
        JLabel l1, l2;
```

```

// Creating two text fields.
// One for rupee and one for
// the dollar
JTextField t1, t2;

// Creating three buttons
JButton b1, b2, b3;

// Naming the labels and setting
// the bounds for the labels
l1 = new JLabel("Rupees:");
l1.setBounds(20, 40, 60, 30);
l2 = new JLabel("Dollars:");
l2.setBounds(170, 40, 60, 30);

// Initializing the text fields with
// 0 by default and setting the
// bounds for the text fields
t1 = new JTextField("0");
t1.setBounds(80, 40, 50, 30);
t2 = new JTextField("0");
t2.setBounds(240, 40, 50, 30);

// Creating a button for INR,
// one button for the dollar
// and one button to close
// and setting the bounds
b1 = new JButton("INR");
b1.setBounds(50, 80, 60, 15);
b2 = new JButton("Dollar");
b2.setBounds(190, 80, 60, 15);
b3 = new JButton("close");
b3.setBounds(150, 150, 60, 30);

// Adding action listener
b1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        // Converting to double
        double d
            = Double.parseDouble(t1.getText());

        // Converting rupees to dollars
        double d1 = (d / 83.24);

        // Getting the string value of the
        // calculated value
        String str1 = String.valueOf(d1);

        3.
        // Placing it in the text box

```

```

        t2.setText(str1);
    }
});

// Adding action listener
b2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        // Converting to double
        double d2
            = Double.parseDouble(t2.getText());

        // converting Dollars to rupees
        double d3 = (d2 * 83.24);

        // Getting the string value of the
        // calculated value
        String str2 = String.valueOf(d3);

        // Placing it in the text box
        t1.setText(str2);
    }
});

// Action listener to close the form
b3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        f.dispose();
    }
});

// Default method for closing the frame
f.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
});

// Adding the created objects
// to the form
f.add(l1);
f.add(t1);
f.add(l2);
f.add(t2);
f.add(b1);
f.add(b2);
f.add(b3);

4.

f.setLayout(null);

```

```

        f.setSize(400, 300);
        f.setVisible(true);
    }

    // Driver code
    public static void main(String args[])
    {
        converter();
    }
}

// Java program to convert from
// rupee to the dollar and vice-versa
// using Java Swing

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class GFG {

    // Function to convert from rupee
    // to the dollar and vice-versa
    // using Java Swing
    public static void converter()
    {

        // Creating a new frame using JFrame
        JFrame f = new JFrame("CONVERTER");

        // Creating two labels
        JLabel l1, l2;

```

2.

```

        // Creating two text fields.
        // One for rupee and one for
        // the dollar
        JTextField t1, t2;

        // Creating three buttons
        JButton b1, b2, b3;

        // Naming the labels and setting
        // the bounds for the labels
        l1 = new JLabel("Rupees:");
        l1.setBounds(20, 40, 60, 30);
        l2 = new JLabel("Dollars:");
        l2.setBounds(170, 40, 60, 30);

        // Initializing the text fields with
        // 0 by default and setting the
        // bounds for the text fields

```

```

t1 = new JTextField("0");
t1.setBounds(80, 40, 50, 30);
t2 = new JTextField("0");
t2.setBounds(240, 40, 50, 30);

// Creating a button for INR,
// one button for the dollar
// and one button to close
// and setting the bounds
b1 = new JButton("INR");
b1.setBounds(50, 80, 60, 15);
b2 = new JButton("Dollar");
b2.setBounds(190, 80, 60, 15);
b3 = new JButton("close");
b3.setBounds(150, 150, 60, 30);

// Adding action listener
b1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        // Converting to double
        double d
            = Double.parseDouble(t1.getText());

        // Converting rupees to dollars
        double d1 = (d / 83.24);

        // Getting the string value of the
        // calculated value
        String str1 = String.valueOf(d1);

        3.
        // Placing it in the text box
        t2.setText(str1);
    }
});

// Adding action listener
b2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        // Converting to double
        double d2
            = Double.parseDouble(t2.getText());

        // converting Dollars to rupees
        double d3 = (d2 * 83.24);

        // Getting the string value of the
        // calculated value
        String str2 = String.valueOf(d3);
    }
});

```



```

        // Placing it in the text box
        t1.setText(str2);
    }
});

// Action listener to close the form
b3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        f.dispose();
    }
});

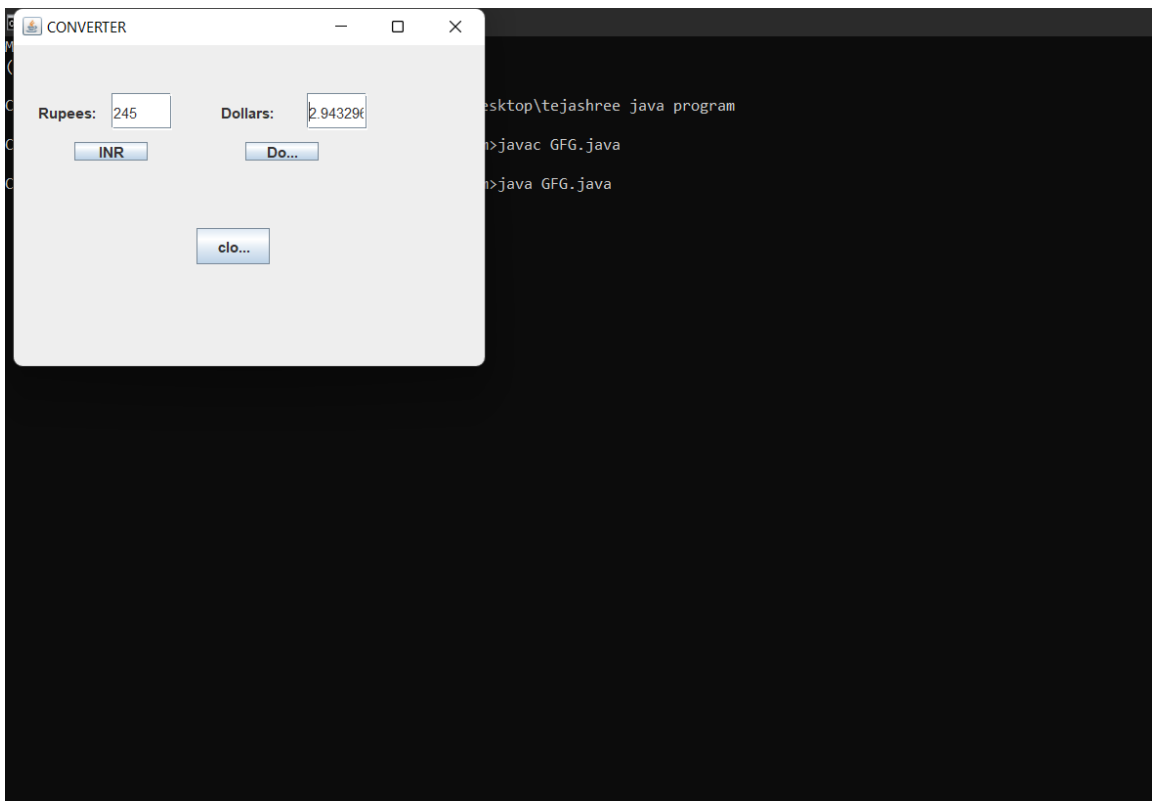
// Default method for closing the frame
f.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
});

// Adding the created objects
// to the form
f.add(l1);
f.add(t1);
f.add(l2);
f.add(t2);
f.add(b1);
f.add(b2);
f.add(b3);

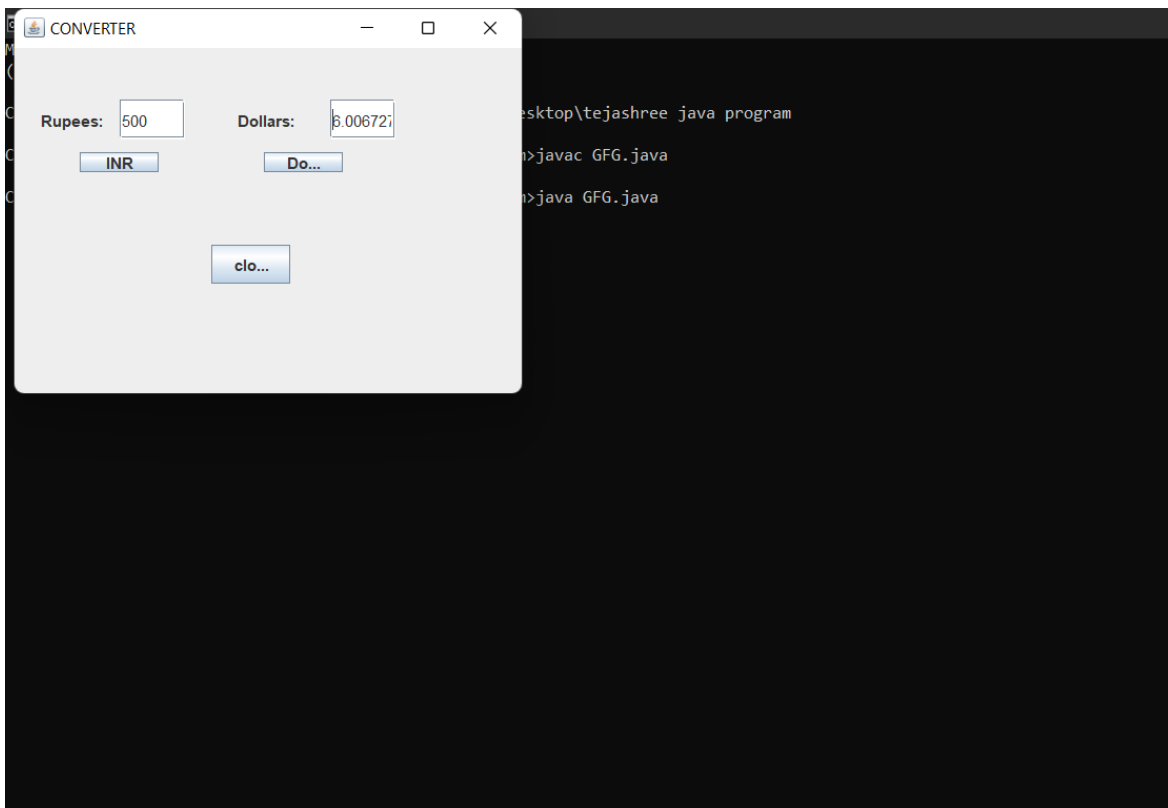
4.
f.setLayout(null);
f.setSize(400, 300);
f.setVisible(true);
}

// Driver code
public static void main(String args[])
{
    converter();
}
}

```



6.



3 Explanation:

This Java program is a simple currency converter that allows the user to convert between Indian Rupees (INR) and US Dollars (USD) using a graphical user interface (GUI) created with Java Swing. Here's an explanation of the code:

1. Importing Necessary Libraries:

- The program starts by importing the required Java Swing libraries for creating a GUI.

2. `GFG` Class:

- This class contains the entire code for the currency converter.

3. `converter` Method:

- This method is responsible for creating the GUI and implementing the currency conversion logic.

4. GUI Components:

- Several GUI components are created within this method, including:
 - `JFrame f`: A new frame that acts as the main window.
 - `JLabel l1` and `JLabel l2`: Labels for "Rupees" and "Dollars."
 - `JTextField t1` and `JTextField t2`: Text fields for entering the amount in rupees and dollars.
 - `JButton b1`, `JButton b2`, and `JButton b3`: Buttons for converting from rupees to dollars, from dollars to rupees, and for closing the program, respectively.

7.

5. Positioning GUI Components:

- The `setBounds` method is used to set the position and size of each GUI component on the frame.

6. Action Listeners:

- Action listeners are added to the "INR" button (`b1`) and "Dollar" button (`b2`). These listeners respond to button clicks and perform the actual currency conversion.
 - `b1` ActionListener: Converts rupees to dollars.
 - `b2` ActionListener: Converts dollars to rupees.
- Additionally, there is an action listener for the "close" button (`b3`) that closes the application when clicked.

7. Conversion Logic:

- When the "INR" button is clicked, the program converts the value in the rupees text field (`t1`) to dollars using the conversion rate of 83.24 (you may need to adjust this rate according to the current exchange rate). The result is displayed in the dollars text field (`t2`).
- When the "Dollar" button is clicked, the program converts the value in the dollars text field (`t2`) to rupees using the same conversion rate. The result is displayed in the rupees text field (`t1`).

8. Window Listener:

- A window listener is added to the frame to handle the event when the user closes the application window. It ensures that the program exits properly.

9. Adding Components to the Frame:

- All the GUI components (labels, text fields, and buttons) are added to the frame using the ``add`` method.

10. Frame Layout and Visibility:

- The frame layout is set to ``null``, meaning that components are manually positioned.
- The frame size is set to 400x300 pixels.
- Finally, the frame is set to be visible, making the GUI accessible to the user.

11. ``main`` Method:

- The ``main`` method is the entry point of the program and simply calls the ``converter`` method to create and display the currency converter GUI.

When you run this program, a window with text fields for rupees and dollars, along with buttons for conversion, will appear. Users can input an amount in one currency, click the corresponding button to convert it to the other currency, and see the result in real-time.