Lab Sheet 2

[1] WAP in java to create multiple threads with following technique:

a) Implementing Runnable

Source code:

```
package Lab2;
class MyRunnable implements Runnable {
 @Override
 public void run() {
     for (int i = 1; i <= 5; i++) {
         System.out.println(Thread.currentThread().getId() + " Value " + i);
     }
 }
}

public class RunnableExample {
 public static void main(String args[]) {
     Thread thread1 = new Thread(new MyRunnable());
     Thread thread2 = new Thread(new MyRunnable());

     // Start the threads
     thread1.start();
     thread2.start();
 }
}
```

Output:

```
15 Value 2
15 Value 3
15 Value 4
15 Value 5
14 Value 2
14 Value 3
14 Value 4
14 Value 5
```

b) Extend Thread class.

Source code:

```
package Lab2;
//  1 b

class MyThread extends Thread {
 @Override
 public void run() {
     for (int i = 1; i <= 5; i++) {
         System.out.println(Thread.currentThread().getId() + " Value " + i);
     }
 }
}
```

```
public class ThreadExample {
 public static void main(String args[]) {
     // Creating multiple threads by extending Thread class
     MyThread thread1 = new MyThread();
     MyThread thread2 = new MyThread();

     // Start the threads
     thread1.start();
     thread2.start();
 }
}
```

Output:

```
<terminated> ThreadExample [Java Application] C:
14 Value 3
14 Value 4
14 Value 5
15 Value 1
15 Value 2
15 Value 3
15 Value 4
15 Value 5
```

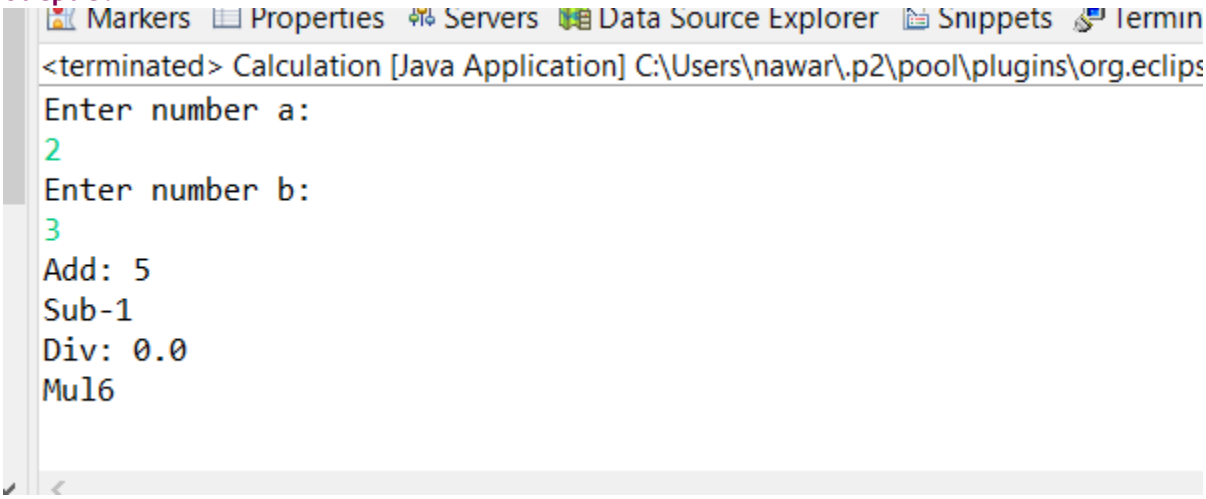
[2] WAP to take two numbers as input from user and perform Addition, subtraction, multiplication and division. [Catch the possible exception if any].

Source code:

```
package Lab2;

///question 2
import java.util.Scanner;
//import java.io.Execption;
import java.io.IOException;

public class Calculation {

	public static void main(String[] args) {
		// TODO Auto-generated method stub


		System.out.println("Enter number a: ");
		int a = new Scanner(System.in).nextInt();

		System.out.println("Enter number b: ");
		int b = new Scanner(System.in).nextInt();

		try {

			int add=a+b;
			int sub=a-b;
```

```
                float div= a/b;
                int mul= a*b;

                System.out.println("Add: "+add);
                System.out.println("Sub"+sub);
                System.out.println("Div: "+div);
                System.out.println("Mul"+mul);

            }
            catch(Exception e){
                System.out.println(e.getMessage());
            }

        }

}
```

Output:



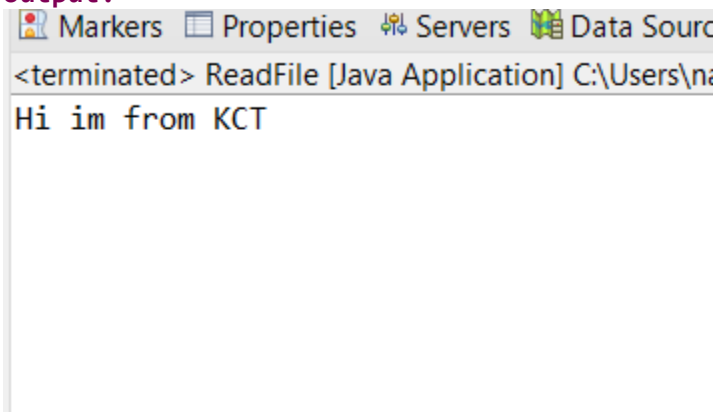
[3] WAP to read the content of a file and display it.

Source code:

```
package Lab2;
////question 3
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class ReadFile {

        public static void main(String[] args) throws FileNotFoundException {
            // TODO Auto-generated method stub
            String
path="C:\\Users\\nawar\\OneDrive\\Documents\\Eclipse_JAVA\\College_Lab_Report\\bin\\L
ab2\\filename.txt";
            File file= new File(path);
            Scanner myReader= new Scanner(file);
            while (myReader.hasNextLine()) {
```

```
            String data = myReader.nextLine();
            System.out.println(data);
          }

      }

}
```

Output:



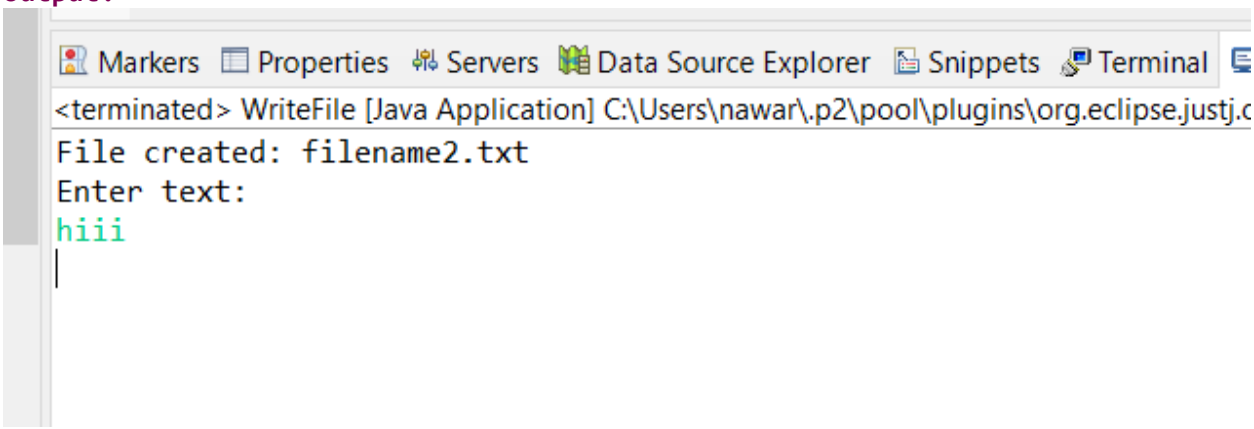
[4] WAP to write some content in a file.

Source code:

```
package Lab2;

import java.io.File;

import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

////question 4
public class WriteFile {

      public static void main(String[] args) throws IOException {
            // TODO Auto-generated method stub

            String
path="C:\\Users\\nawar\\OneDrive\\Documents\\Eclipse_JAVA\\College_Lab_Report\\bin\\Lab2\\filename2.txt";
            File myObj  = new File(path);

            if(myObj.createNewFile()) {
                  System.out.println("File created: " + myObj .getName());
            }else {
              System.out.println("File already exists.");
            }

            System.out.println("Enter text: ");
```

```
            String data= new Scanner(System.in).nextLine();


            FileWriter myWriter = new FileWriter(path);
           myWriter.write(data);
           myWriter.close();

      }

}
```

Output:



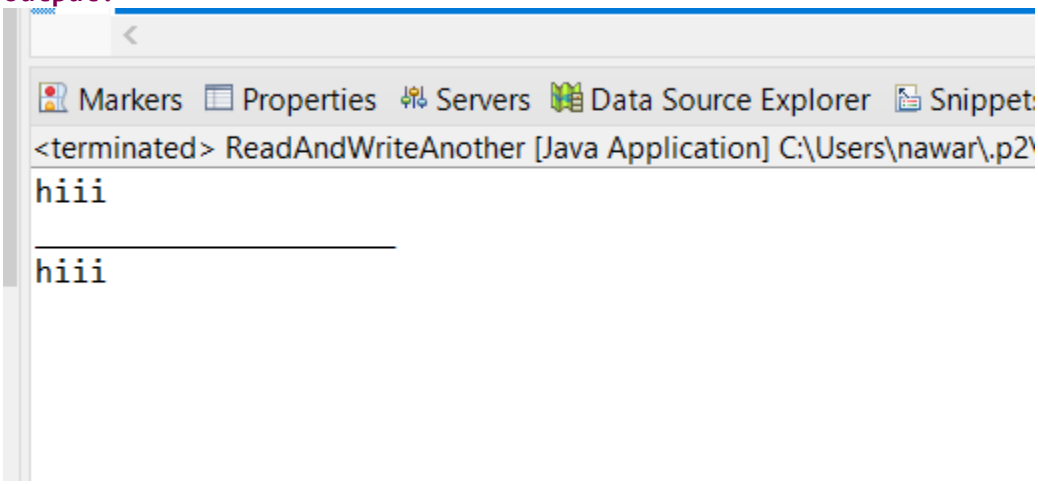
[5] WAP to read the content of a file and write it to another file.

Source code:

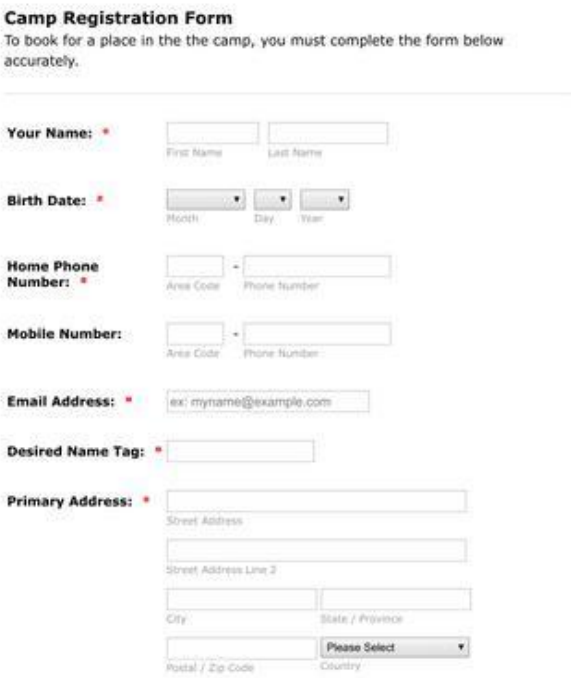
```
package Lab2;

///question 5
import java.io.*;

import java.util.Scanner;
public class ReadAndWriteAnother {

      public static void main(String[] args) throws IOException {
            // TODO Auto-generated method stub
            String
pathRead="C:\\Users\\nawar\\OneDrive\\Documents\\Eclipse_JAVA\\College_Lab_Report\\bi
n\\Lab2\\filename2.txt";
            String
pathWrite="C:\\Users\\nawar\\OneDrive\\Documents\\Eclipse_JAVA\\College_Lab_Report\\b
in\\Lab2\\filename3.txt";
            FileWriter fw;
            File file= new File(pathRead);
            Scanner sc = new Scanner(file);

            while(sc.hasNextLine()) {
```

```
                String data = sc.nextLine();
                System.out.println(data);
                fw= new FileWriter(pathWrite);
                fw.write(data);
                fw.close();
            }
            System.out.println("_____________________");

            File file2= new File(pathWrite);
            Scanner sc2 = new Scanner(file2);

            while(sc2.hasNextLine()) {
                String data= sc2.nextLine();
                                    System.out.println(data);

            }
            sc.close();

        }

}
```

Output:



[6] Design a simple registration form using swing components. (Form should include textfield, password field, radio button, check box, combo box, image, and text area. Following form is just for your reference!)



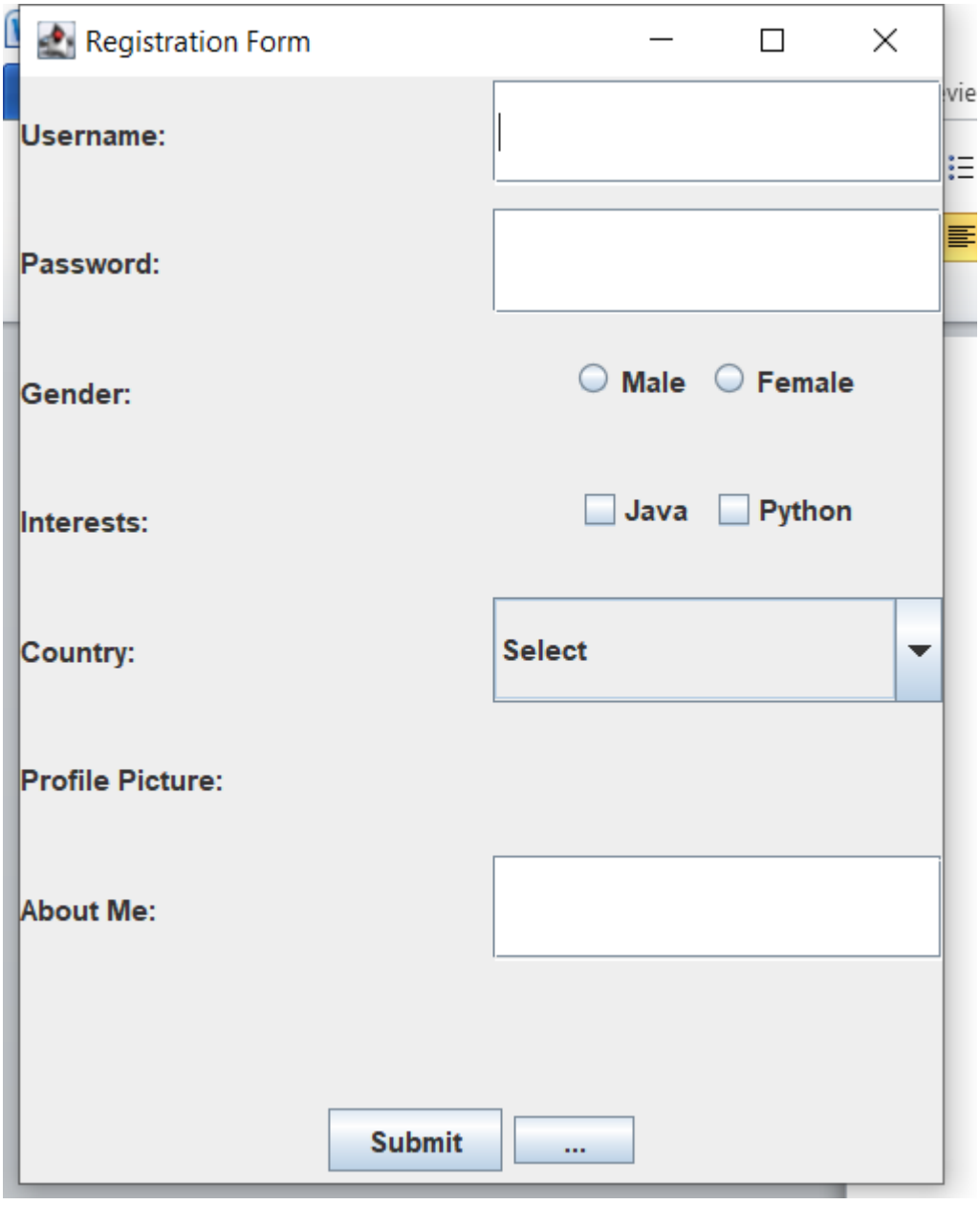
Source code:

```
package Lab2;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class RegistrationForm {

    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("Registration Form");
        frame.setSize(400, 500);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create a panel for the form
        JPanel formPanel = new JPanel();
        formPanel.setLayout(new GridLayout(8, 2, 10, 10));

        // Add components to the form
        formPanel.add(new JLabel("Username:"));
        JTextField usernameField = new JTextField();
```

```
        formPanel.add(usernameField);

        formPanel.add(new JLabel("Password:"));
        JPasswordField passwordField = new JPasswordField();
        formPanel.add(passwordField);

        formPanel.add(new JLabel("Gender:"));
        JRadioButton maleRadioButton = new JRadioButton("Male");
        JRadioButton femaleRadioButton = new JRadioButton("Female");
        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(maleRadioButton);
        genderGroup.add(femaleRadioButton);
        JPanel genderPanel = new JPanel();
        genderPanel.add(maleRadioButton);
        genderPanel.add(femaleRadioButton);
        formPanel.add(genderPanel);

        formPanel.add(new JLabel("Interests:"));
        JCheckBox javaCheckBox = new JCheckBox("Java");
        JCheckBox pythonCheckBox = new JCheckBox("Python");
        JPanel interestsPanel = new JPanel();
        interestsPanel.add(javaCheckBox);
        interestsPanel.add(pythonCheckBox);
        formPanel.add(interestsPanel);

        formPanel.add(new JLabel("Country:"));
        String[] countries = {"Select", "USA", "Canada", "UK", "Australia"};
        JComboBox<String> countryComboBox = new JComboBox<>(countries);
        formPanel.add(countryComboBox);

        formPanel.add(new JLabel("Profile Picture:"));
        ImageIcon defaultImage = new ImageIcon("default_profile.png"); // Provide a
path to your image
        JLabel imageLabel = new JLabel(defaultImage);
        formPanel.add(imageLabel);

        formPanel.add(new JLabel("About Me:"));
        JTextArea aboutMeTextArea = new JTextArea();
        JScrollPane scrollPane = new JScrollPane(aboutMeTextArea);
        formPanel.add(scrollPane);

        // Create a button to submit the form
        JButton submitButton = new JButton("Submit");
        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // You can handle form submission logic here
                JOptionPane.showMessageDialog(frame, "Form submitted successfully!");
            }
        });

        // Add the form panel and submit button to the main frame
        frame.setLayout(new BorderLayout());
        frame.add(formPanel, BorderLayout.CENTER);
        frame.add(submitButton, BorderLayout.SOUTH);
```

```
        // Set frame visibility
        frame.setVisible(true);
    }
}
```

Output:



[7] Design a simple calculator using GUI components (pick and drop). Calculator must perform at least all the arithmetic operations and reset option).

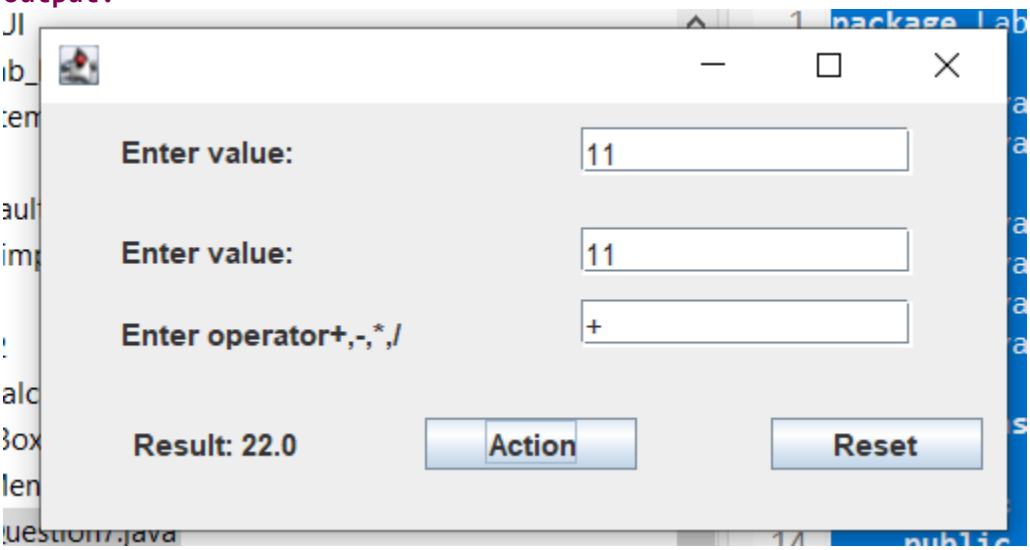
Source code:

```
package Lab2;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class Question7 {

    public JFrame frame;
    public JTextField textField;
    public JTextField textField_1;
    public JLabel resultLabel;
    public JLabel lblNewLabel_2;
    public JLabel lblNewLabel_3;

    public double num1, num2, result;
    public char op;
    private JTextField textField_2;

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Question7 n = new Question7();
    }

    public Question7() {
        initialize();
        frame.setVisible(true);
    }

    private void initialize() {
        frame = new JFrame();
        frame.setBounds(100, 100, 401, 208);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);

        textField = new JTextField();
        textField.setBounds(216, 10, 133, 19);
        frame.getContentPane().add(textField);
        textField.setColumns(10);

        textField_1 = new JTextField();
        textField_1.setBounds(216, 50, 133, 19);
        frame.getContentPane().add(textField_1);
        textField_1.setColumns(10);

        JLabel lblNewLabel = new JLabel("Enter value: ");
```

```
        lblNewLabel.setBounds(32, 13, 100, 13);
        frame.getContentPane().add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("Enter value: ");
        lblNewLabel_1.setBounds(32, 53, 100, 13);
        frame.getContentPane().add(lblNewLabel_1);

        lblNewLabel_2 = new JLabel("Result: ");
        lblNewLabel_2.setBounds(37, 130, 95, 13);
        frame.getContentPane().add(lblNewLabel_2);

        resultLabel = new JLabel("");
        resultLabel.setBounds(154, 130, 133, 13);
        frame.getContentPane().add(resultLabel);

        JButton btnNewButton = new JButton("Action");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                performAction();
            }
        });
        btnNewButton.setBounds(154, 126, 85, 21);
        frame.getContentPane().add(btnNewButton);

        JButton btnNewButton_1 = new JButton("Reset");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                reset();
            }
        });
        btnNewButton_1.setBounds(292, 126, 85, 21);
        frame.getContentPane().add(btnNewButton_1);

        lblNewLabel_3 = new JLabel("Enter operator\n" + "+,-,*,/");
        lblNewLabel_3.setBounds(32, 86, 150, 13);
        frame.getContentPane().add(lblNewLabel_3);

        textField_2 = new JTextField();
        textField_2.setBounds(216, 79, 133, 19);
        frame.getContentPane().add(textField_2);
        textField_2.setColumns(10);
    }

    private void performAction() {
        String a = textField_2.getText();
        op = a.charAt(0);

        try {
            num1 = Double.parseDouble(textField.getText());
            num2 = Double.parseDouble(textField_1.getText());

            switch (op) {
                case '+':
                    result = num1 + num2;
                    break;
```

```
                case '-':
                    result = num1 - num2;
                    break;
                case '*':
                    result = num1 * num2;
                    break;
                case '/':
                    if (num2 != 0) {
                        result = num1 / num2;
                    } else {
                        lblNewLabel_2.setText("Error");
                        return;
                    }
                    break;
            }

            lblNewLabel_2.setText("Result: " + result);

        } catch (NumberFormatException ex) {
            resultLabel.setText("Invalid input");
        }
    }

    private void reset() {
        textField.setText("");
        textField_1.setText("");
        lblNewLabel_2.setText("Result: ");
        textField_2.setText("");
        num1 = 0;
        num2 = 0;
        result = 0;
        op = '\0';
    }
}
```

Output:

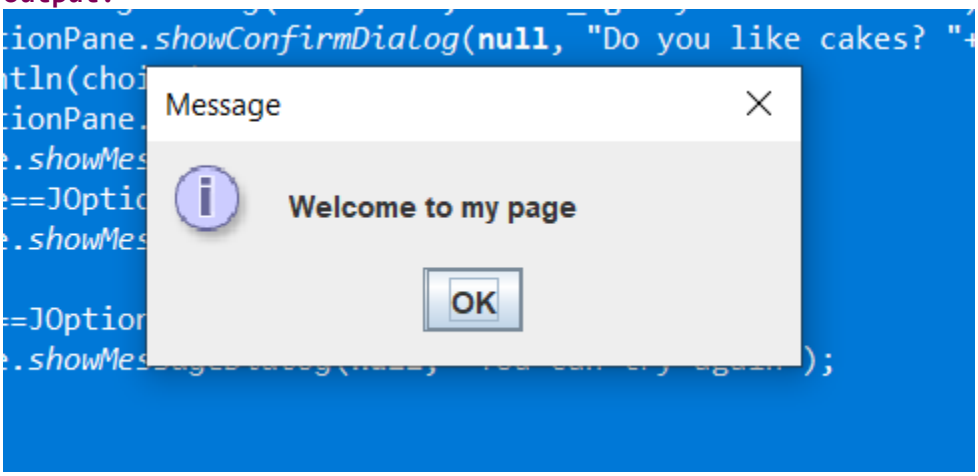


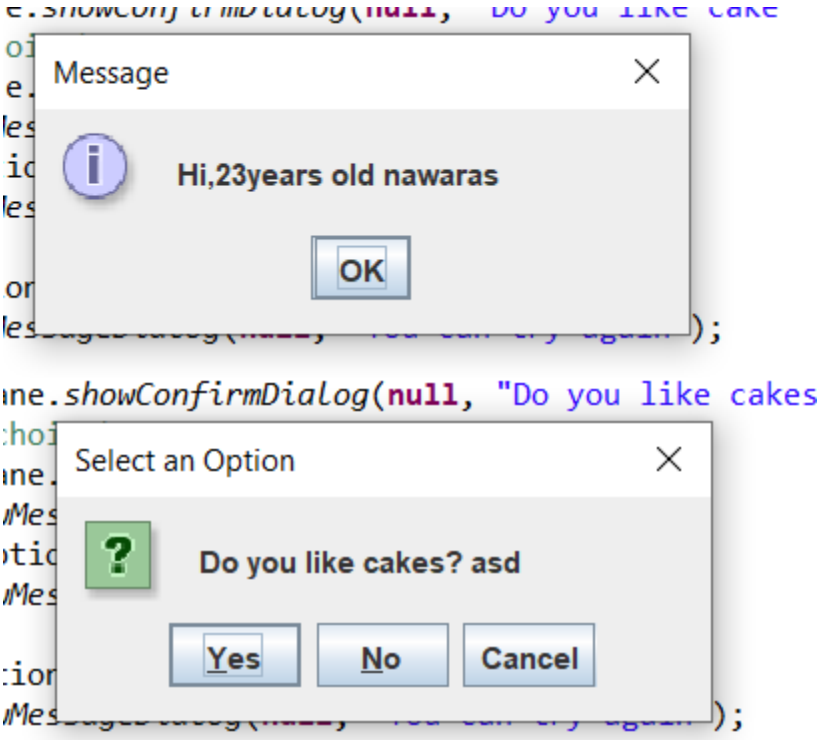
[8] WAP to implement different dialog boxes.

Source code:

```
package Lab2;

//question  8
import javax.swing.JOptionPane;
public class JBox {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		JOptionPane.showMessageDialog(null, "Welcome to my page");
		String name=JOptionPane.showInputDialog(null, "What's your name!!");

		String age=JOptionPane.showInputDialog(null, "What's your age!!");
		int real_age= Integer.parseInt(age);

		JOptionPane.showMessageDialog(null, "Hi,"+real_age+"years old "+name);

		int choice=JOptionPane.showConfirmDialog(null, "Do you like cakes?
"+name);
//		System.out.println(choice);
		if(choice==JOptionPane.YES_OPTION) {
			JOptionPane.showMessageDialog(null, "Thank you !!");
		}else if(choice==JOptionPane.NO_OPTION) {
			JOptionPane.showMessageDialog(null, "Im sad!!");
		}
		else if(choice==JOptionPane.CANCEL_OPTION) {
			JOptionPane.showMessageDialog(null, "You can try again");
		}

	}
}
```

Output:





[9] WAP to implement menu and popup menu option in Java.

Source code:

```
package Lab2;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MenuExample {
    public MenuExample() {
        JFrame frame = new JFrame("PopupMenu Example");

        // Create a popup menu
        JPopupMenu popupMenu = new JPopupMenu("Edit");

        // Create menu items
        JMenuItem cut = new JMenuItem("Cut");
        cut.setActionCommand("Cut");
        JMenuItem copy = new JMenuItem("Copy");
        copy.setActionCommand("Copy");
        JMenuItem paste = new JMenuItem("Paste");
        paste.setActionCommand("Paste");

        // Add menu items to the popup menu
```

```
        popupMenu.add(cut);
        popupMenu.add(copy);
        popupMenu.add(paste);

        // Add action listeners to menu items
        cut.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(frame, "Cut action performed");
            }
        });

        copy.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(frame, "Copy action performed");
            }
        });

        paste.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(frame, "Paste action performed");
            }
        });

        // Add a dummy component to the frame to trigger the popup menu
        JButton dummyButton = new JButton("Right-click here");
        dummyButton.setComponentPopupMenu(popupMenu);

        // Use a layout manager (FlowLayout in this case)
        frame.setLayout(new FlowLayout());
        frame.add(dummyButton);

        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public static void main(String args[]) {
        SwingUtilities.invokeLater(() -> new MenuExample());
    }
}
```

Output: