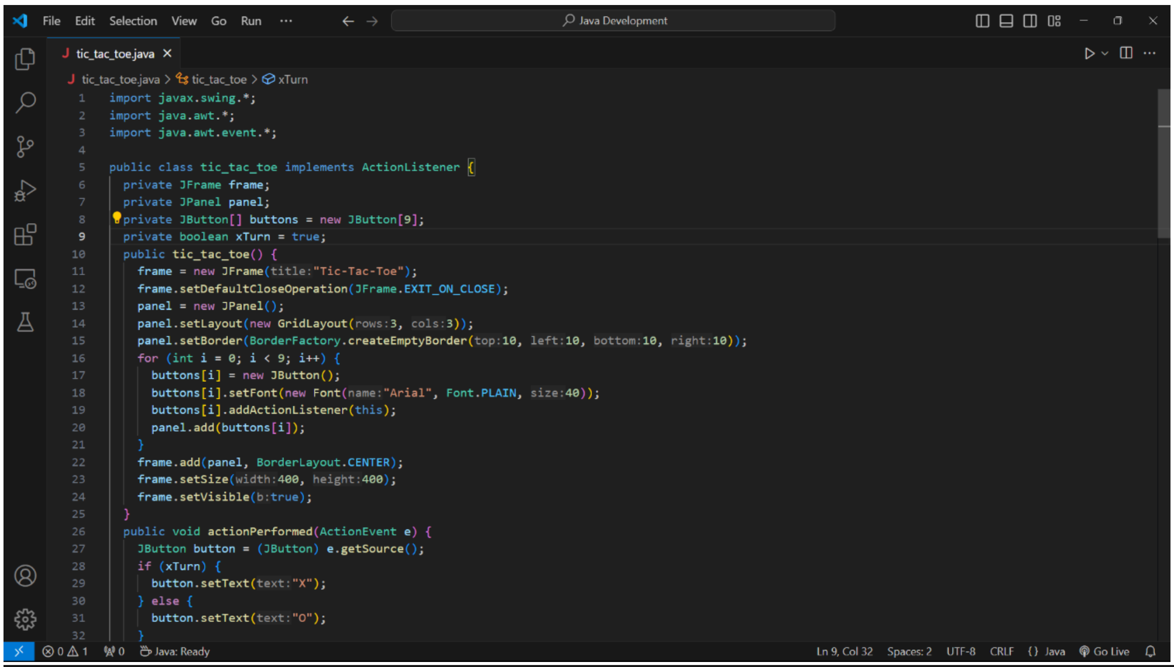
ASSINGMENT-4

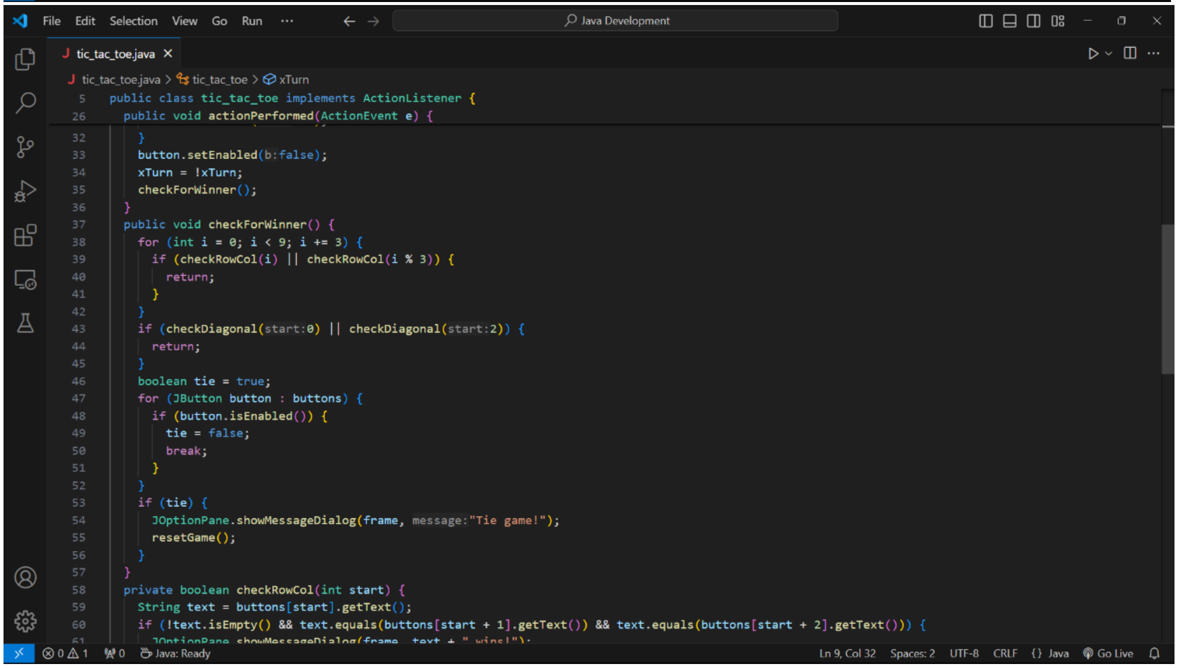
APPLICATIONS

- **SUBMITTED BY: - KANISHK**
- **CLASS: - CSE-37**
- **ROLL NO: - 2205130**

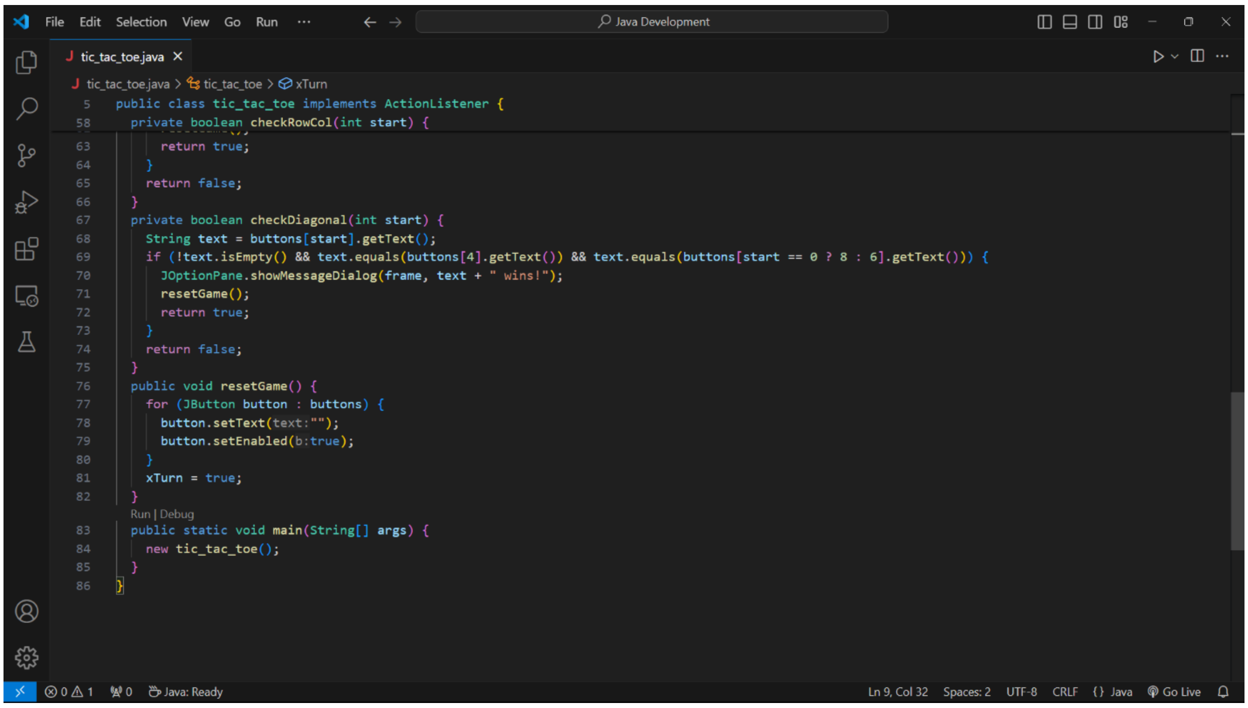
1. NOTEPAD INPUT: -


File Edit Selection View Go Run ···
Java Development
tic_tac_toe.java
tic_tac_toe.java > tic_tac_toe > xTurn
0 ⚠ 1 0 Java: Ready
Ln 9, Col 32 Spaces: 2 UTF-8 CRLF {} Java Go Live

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class tic_tac_toe implements ActionListener {
  private JFrame frame;
  private JPanel panel;
  private JButton[] buttons = new JButton[9];
  private boolean xTurn = true;
  public tic_tac_toe() {
    frame = new JFrame(title:"Tic-Tac-Toe");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    panel = new JPanel();
    panel.setLayout(new GridLayout(rows:3, cols:3));
    panel.setBorder(BorderFactory.createEmptyBorder(top:10, left:10, bottom:10, right:10));
    for (int i = 0; i < 9; i++) {
      buttons[i] = new JButton();
      buttons[i].setFont(new Font(name:"Arial", Font.PLAIN, size:40));
      buttons[i].addActionListener(this);
      panel.add(buttons[i]);
    }
    frame.add(panel, BorderLayout.CENTER);
    frame.setSize(width:400, height:400);
    frame.setVisible(b:true);
  }
  public void actionPerformed(ActionEvent e) {
    JButton button = (JButton) e.getSource();
    if (xTurn) {
      button.setText(text:"X");
    } else {
      button.setText(text:"O");
    }
```

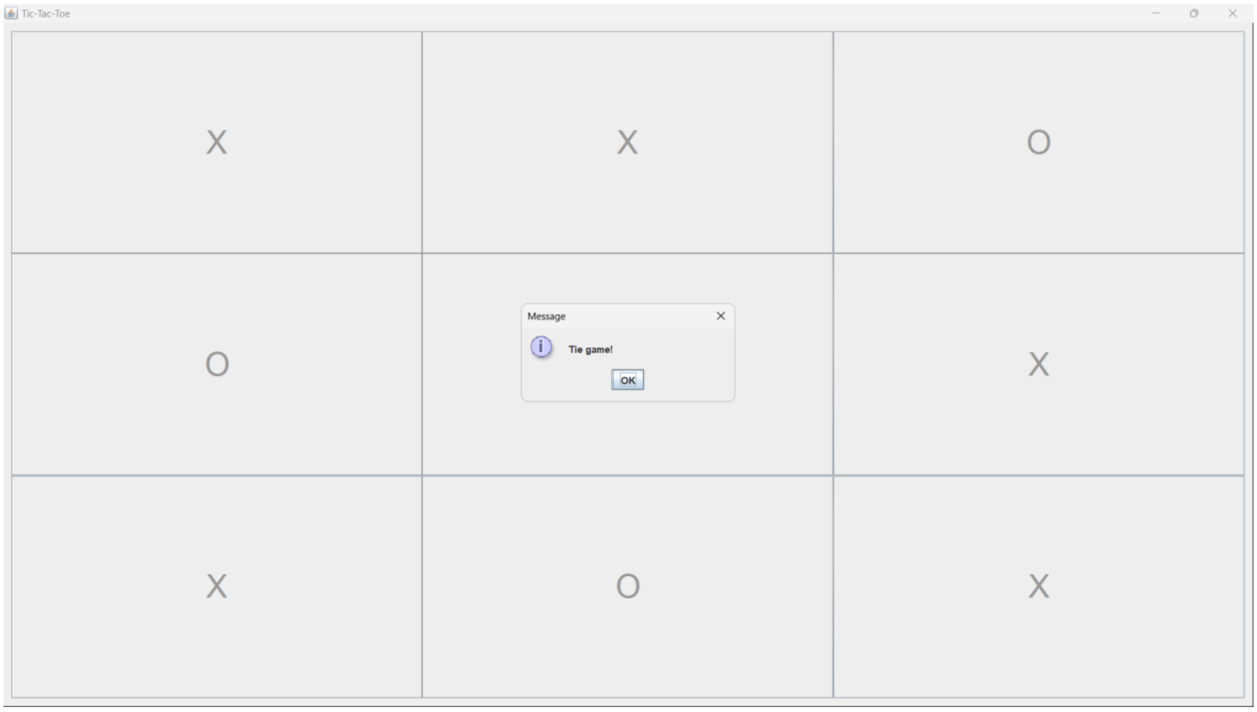

File Edit Selection View Go Run ···
Java Development
tic_tac_toe.java
tic_tac_toe.java > tic_tac_toe > xTurn
0 ⚠ 1 0 Java: Ready
Ln 9, Col 32 Spaces: 2 UTF-8 CRLF {} Java Go Live

```java
public class tic_tac_toe implements ActionListener {
  public void actionPerformed(ActionEvent e) {
    }
    button.setEnabled(b:false);
    xTurn = !xTurn;
    checkForWinner();
  }
  public void checkForWinner() {
    for (int i = 0; i < 9; i += 3) {
      if (checkRowCol(i) || checkRowCol(i % 3)) {
        return;
      }
    }
    if (checkDiagonal(start:0) || checkDiagonal(start:2)) {
      return;
    }
    boolean tie = true;
    for (JButton button : buttons) {
      if (button.isEnabled()) {
        tie = false;
        break;
      }
    }
    if (tie) {
      JOptionPane.showMessageDialog(frame, message:"Tie game!");
      resetGame();
    }
  }
  private boolean checkRowCol(int start) {
    String text = buttons[start].getText();
    if (!text.isEmpty() && text.equals(buttons[start + 1].getText()) && text.equals(buttons[start + 2].getText())) {
      JOptionPane.showMessageDialog(frame, text + " wins!");
```

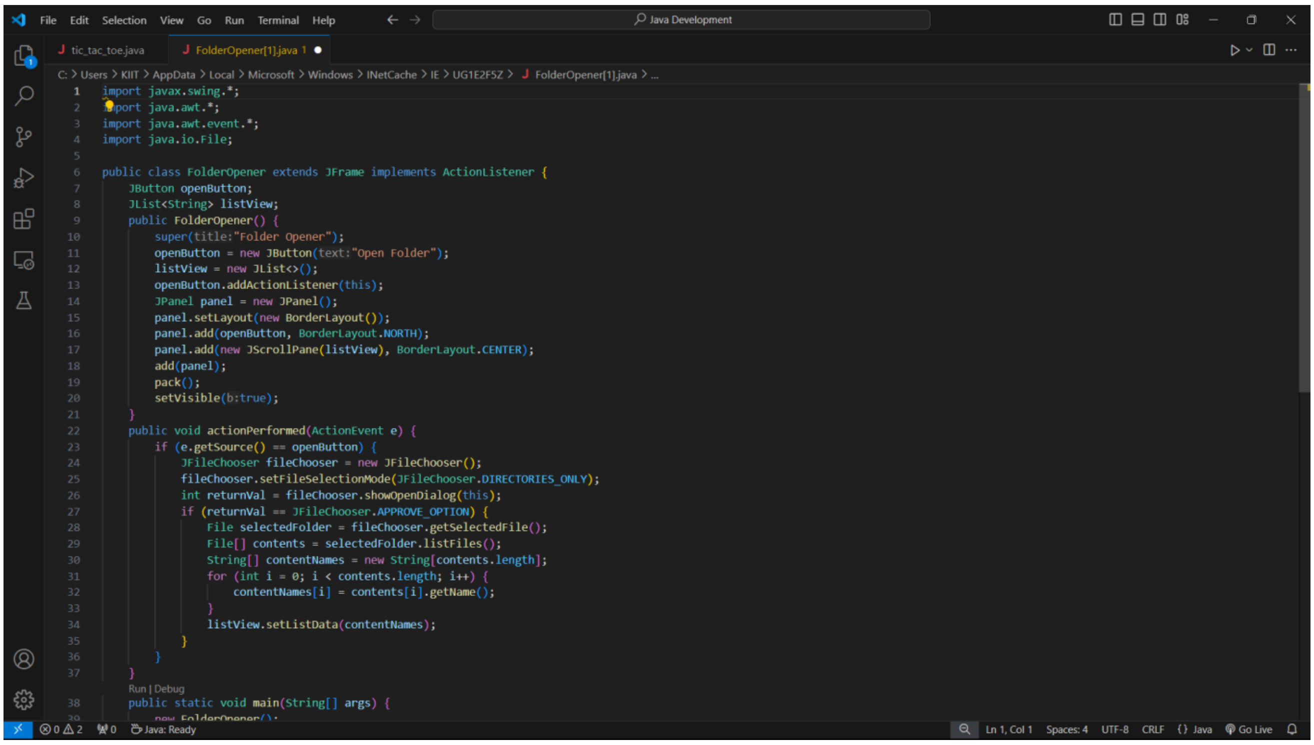

File Edit Selection View Go Run ···
Java Development
tic_tac_toe.java
tic_tac_toe.java > tic_tac_toe > xTurn
⊗ 0 ⚠ 1 0 Java: Ready
Ln 9, Col 32 Spaces: 2 UTF-8 CRLF {} Java Go Live

```java
public class tic_tac_toe implements ActionListener {
  private boolean checkRowCol(int start) {
      return true;
    }
    return false;
  }
  private boolean checkDiagonal(int start) {
    String text = buttons[start].getText();
    if (!text.isEmpty() && text.equals(buttons[4].getText()) && text.equals(buttons[start == 0 ? 8 : 6].getText())) {
      JOptionPane.showMessageDialog(frame, text + " wins!");
      resetGame();
      return true;
    }
    return false;
  }
  public void resetGame() {
    for (JButton button : buttons) {
      button.setText(text:"");
      button.setEnabled(b:true);
    }
    xTurn = true;
  }
  Run | Debug
  public static void main(String[] args) {
    new tic_tac_toe();
  }
}
```

OUTPUT: -

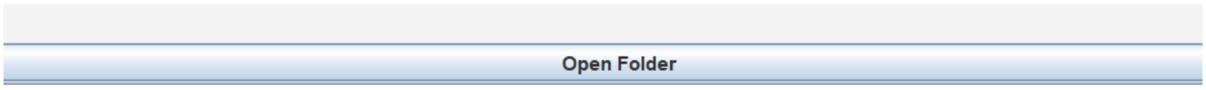


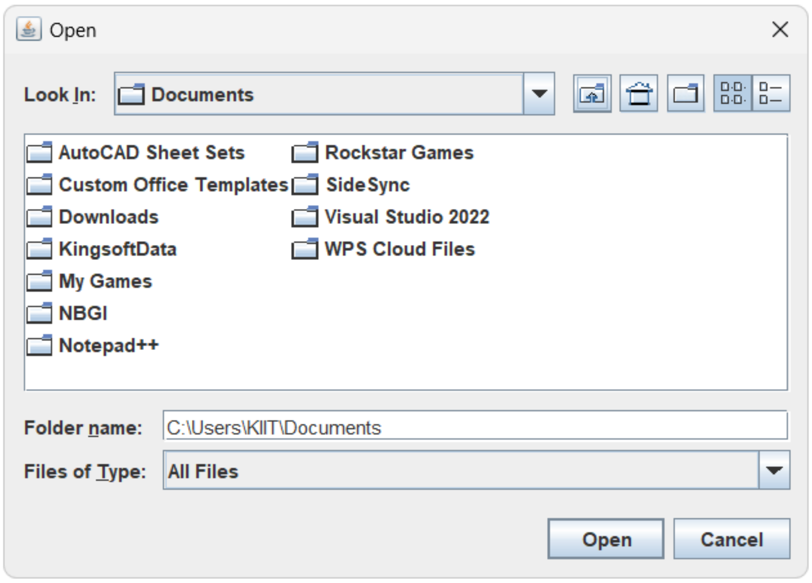
2.FOLDER_OPENER INPUT: -



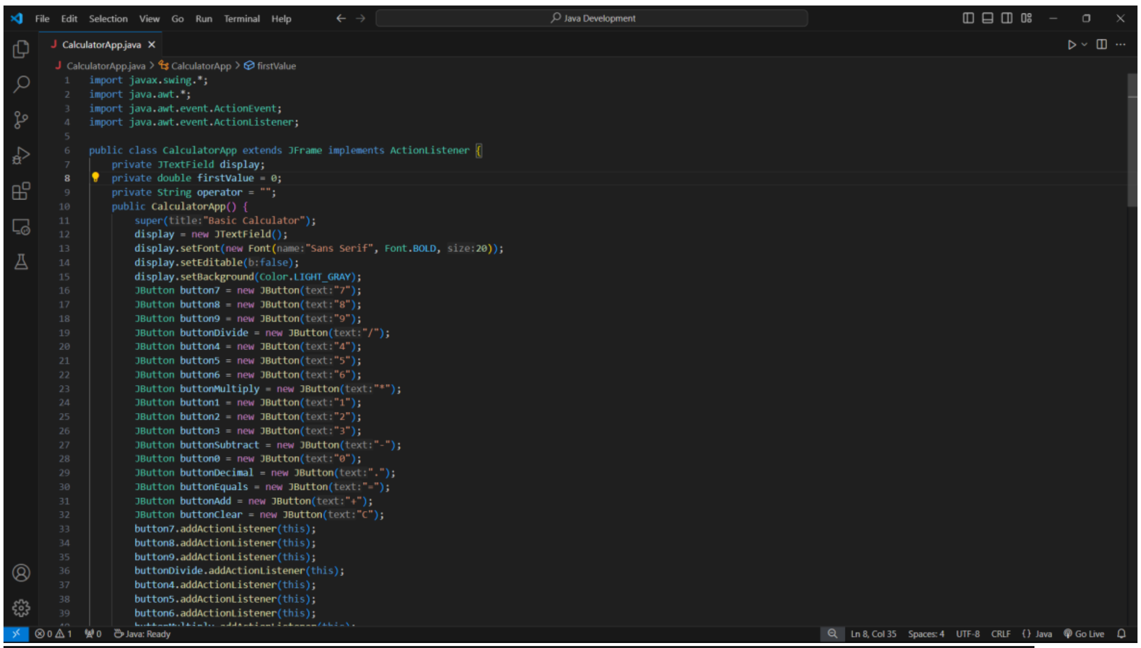
```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.File;

public class FolderOpener extends JFrame implements ActionListener {
    JButton openButton;
    JList<String> listView;
    public FolderOpener() {
        super(title:"Folder Opener");
        openButton = new JButton(text:"Open Folder");
        listView = new JList<>();
        openButton.addActionListener(this);
        JPanel panel = new JPanel();
        panel.setLayout(new BorderLayout());
        panel.add(openButton, BorderLayout.NORTH);
        panel.add(new JScrollPane(listView), BorderLayout.CENTER);
        add(panel);
        pack();
        setVisible(b:true);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == openButton) {
            JFileChooser fileChooser = new JFileChooser();
            fileChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
            int returnVal = fileChooser.showOpenDialog(this);
            if (returnVal == JFileChooser.APPROVE_OPTION) {
                File selectedFolder = fileChooser.getSelectedFile();
                File[] contents = selectedFolder.listFiles();
                String[] contentNames = new String[contents.length];
                for (int i = 0; i < contents.length; i++) {
                    contentNames[i] = contents[i].getName();
                }
                listView.setListData(contentNames);
            }
        }
    }
    public static void main(String[] args) {
        new FolderOpener();
```

OUTPUT: -

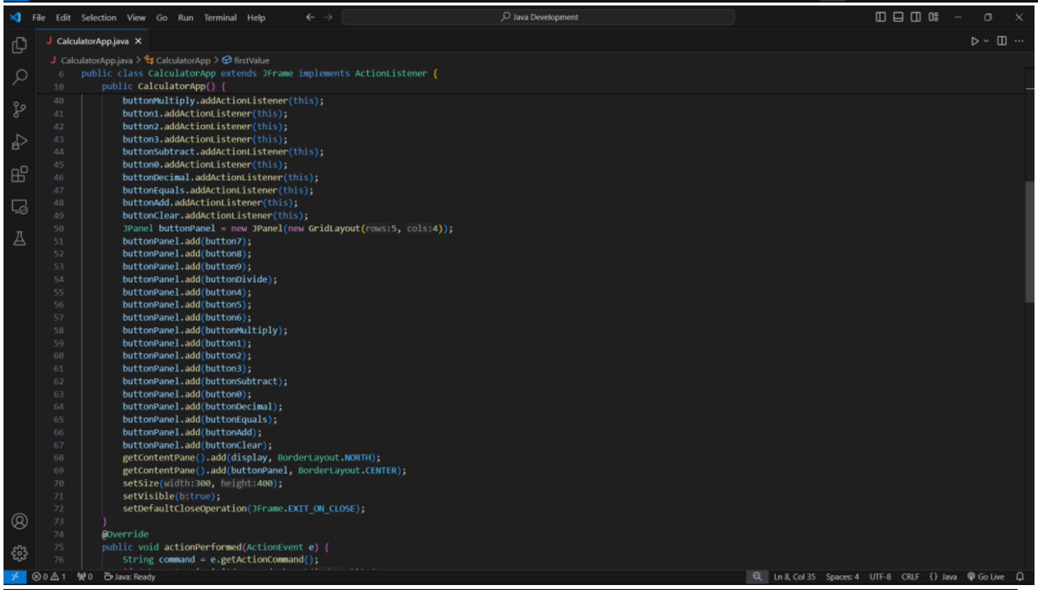




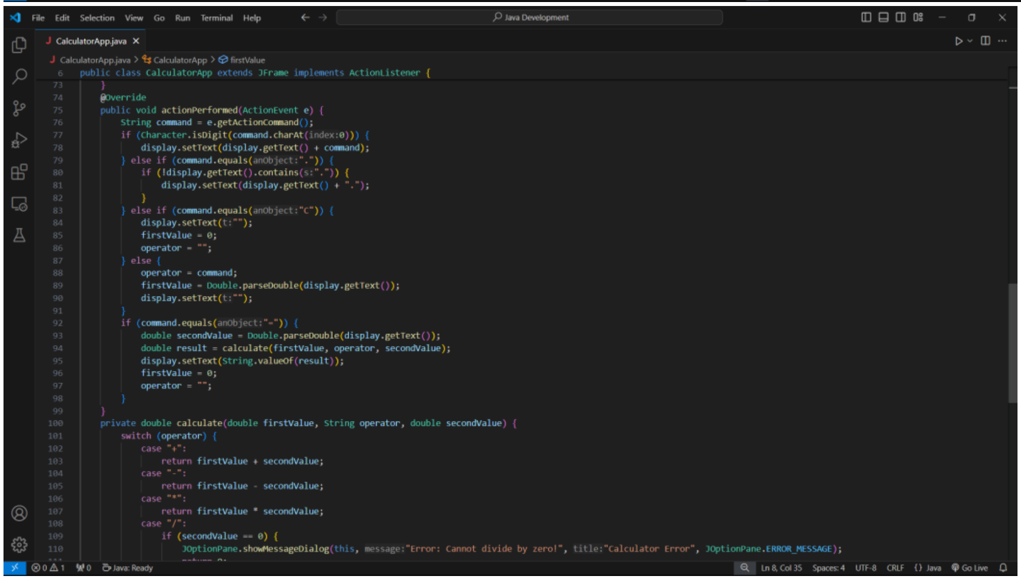
3.CALCULATOR_APP INPUT: -



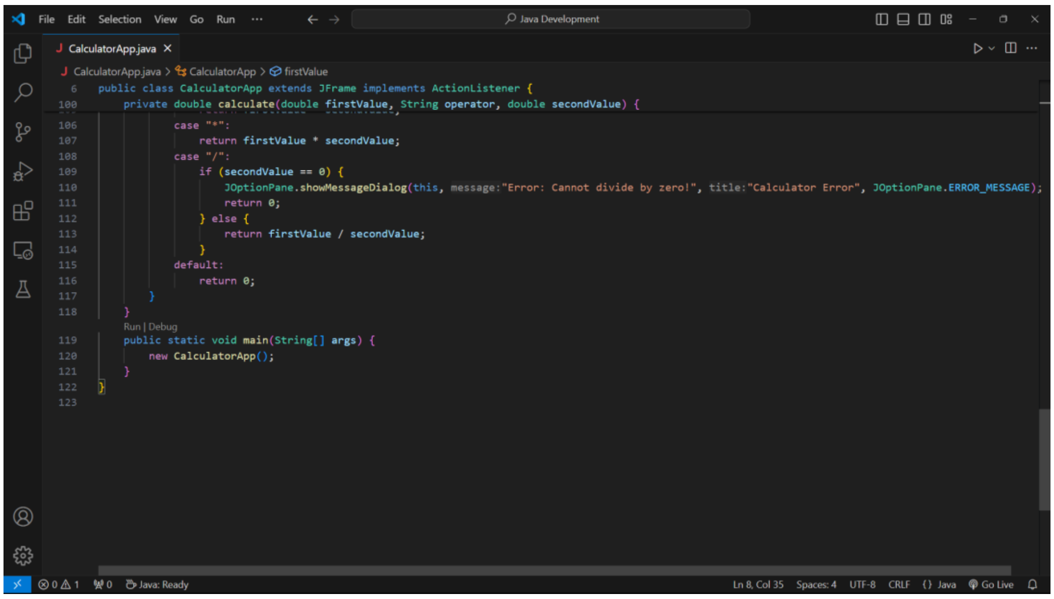
```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class CalculatorApp extends JFrame implements ActionListener {
    private JTextField display;
    private double firstValue = 0;
    private String operator = "";
    public CalculatorApp() {
        super(title:"Basic Calculator");
        display = new JTextField();
        display.setFont(new Font(name:"Sans Serif", Font.BOLD, size:20));
        display.setEditable(b:false);
        display.setBackground(Color.LIGHT_GRAY);
        JButton button7 = new JButton(text:"7");
        JButton button8 = new JButton(text:"8");
        JButton button9 = new JButton(text:"9");
        JButton buttonDivide = new JButton(text:"/");
        JButton button4 = new JButton(text:"4");
        JButton button5 = new JButton(text:"5");
        JButton button6 = new JButton(text:"6");
        JButton buttonMultiply = new JButton(text:"*");
        JButton button1 = new JButton(text:"1");
        JButton button2 = new JButton(text:"2");
        JButton button3 = new JButton(text:"3");
        JButton buttonSubtract = new JButton(text:"-");
        JButton button0 = new JButton(text:"0");
        JButton buttonDecimal = new JButton(text:".");
        JButton buttonEquals = new JButton(text:"=");
        JButton buttonAdd = new JButton(text:"+");
        JButton buttonClear = new JButton(text:"C");
        button7.addActionListener(this);
        button8.addActionListener(this);
        button9.addActionListener(this);
        buttonDivide.addActionListener(this);
        button4.addActionListener(this);
        button5.addActionListener(this);
        button6.addActionListener(this);
```



```java
public class CalculatorApp extends JFrame implements ActionListener {
    public CalculatorApp() {
        buttonMultiply.addActionListener(this);
        button1.addActionListener(this);
        button2.addActionListener(this);
        button3.addActionListener(this);
        buttonSubtract.addActionListener(this);
        button0.addActionListener(this);
        buttonDecimal.addActionListener(this);
        buttonEquals.addActionListener(this);
        buttonAdd.addActionListener(this);
        buttonClear.addActionListener(this);
        JPanel buttonPanel = new JPanel(new GridLayout(rows:5, cols:4));
        buttonPanel.add(button7);
        buttonPanel.add(button8);
        buttonPanel.add(button9);
        buttonPanel.add(buttonDivide);
        buttonPanel.add(button4);
        buttonPanel.add(button5);
        buttonPanel.add(button6);
        buttonPanel.add(buttonMultiply);
        buttonPanel.add(button1);
        buttonPanel.add(button2);
        buttonPanel.add(button3);
        buttonPanel.add(buttonSubtract);
        buttonPanel.add(button0);
        buttonPanel.add(buttonDecimal);
        buttonPanel.add(buttonEquals);
        buttonPanel.add(buttonAdd);
        buttonPanel.add(buttonClear);
        getContentPane().add(display, BorderLayout.NORTH);
        getContentPane().add(buttonPanel, BorderLayout.CENTER);
        setSize(width:300, height:400);
        setVisible(b:true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();
```

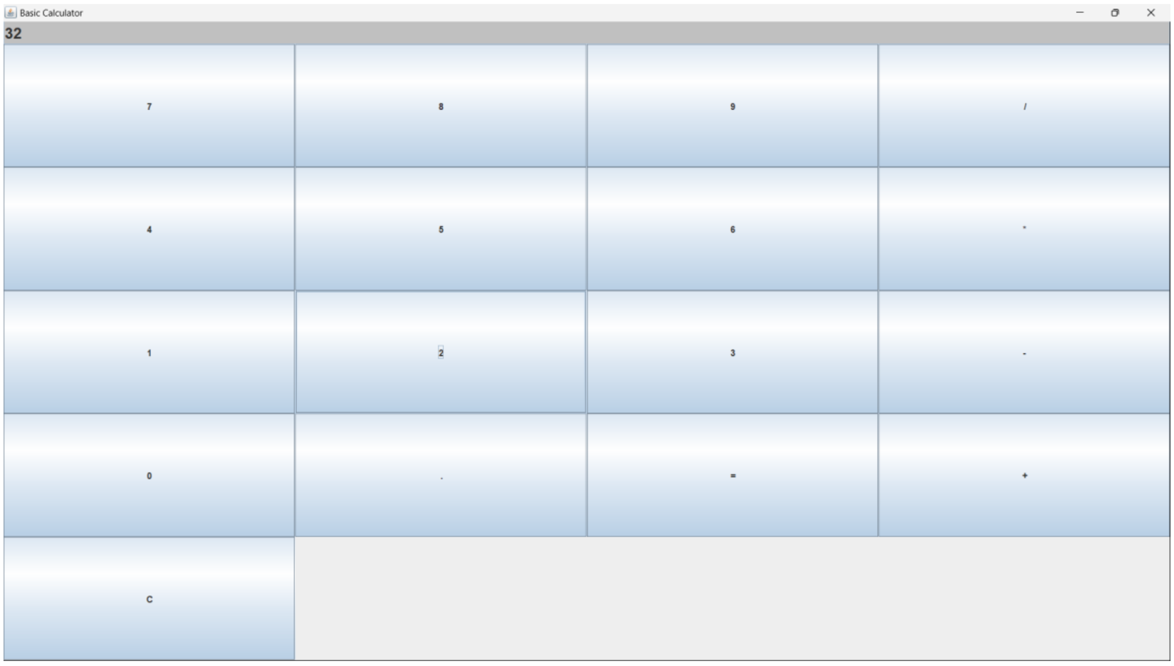


```java
public class CalculatorApp extends JFrame implements ActionListener {
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();
        if (Character.isDigit(command.charAt(index:0))) {
            display.setText(display.getText() + command);
        } else if (command.equals(anObject:".")) {
            if (!display.getText().contains(s:".")) {
                display.setText(display.getText() + ".");
            }
        } else if (command.equals(anObject:"C")) {
            display.setText(t:"");
            firstValue = 0;
            operator = "";
        } else {
            operator = command;
            firstValue = Double.parseDouble(display.getText());
            display.setText(t:"");
        }
        if (command.equals(anObject:"=")) {
            double secondValue = Double.parseDouble(display.getText());
            double result = calculate(firstValue, operator, secondValue);
            display.setText(String.valueOf(result));
            firstValue = 0;
            operator = "";
        }
    }
    private double calculate(double firstValue, String operator, double secondValue) {
        switch (operator) {
            case "+":
                return firstValue + secondValue;
            case "-":
                return firstValue - secondValue;
            case "*":
                return firstValue * secondValue;
            case "/":
                if (secondValue == 0) {
                    JOptionPane.showMessageDialog(this, message:"Error: Cannot divide by zero!", title:"Calculator Error", JOptionPane.ERROR_MESSAGE);
```

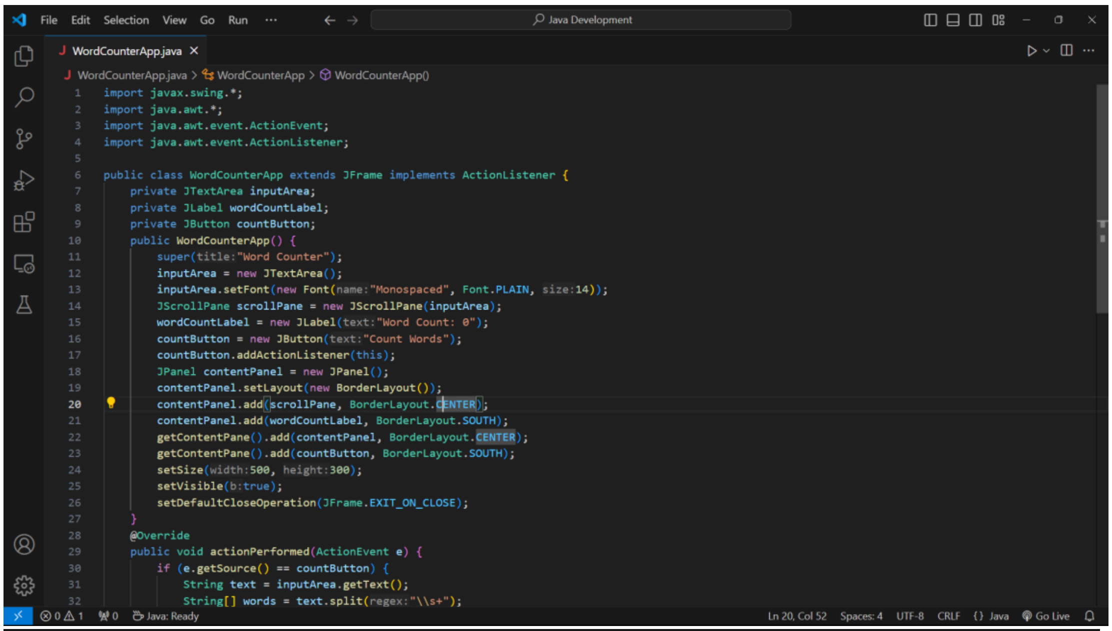


```java
public class CalculatorApp extends JFrame implements ActionListener {
    private double calculate(double firstValue, String operator, double secondValue) {
            case "*":
                return firstValue * secondValue;
            case "/":
                if (secondValue == 0) {
                    JOptionPane.showMessageDialog(this, message:"Error: Cannot divide by zero!", title:"Calculator Error", JOptionPane.ERROR_MESSAGE);
                    return 0;
                } else {
                    return firstValue / secondValue;
                }
            default:
                return 0;
        }
    }
    Run | Debug
    public static void main(String[] args) {
        new CalculatorApp();
    }
}
```

OUTPUT: -

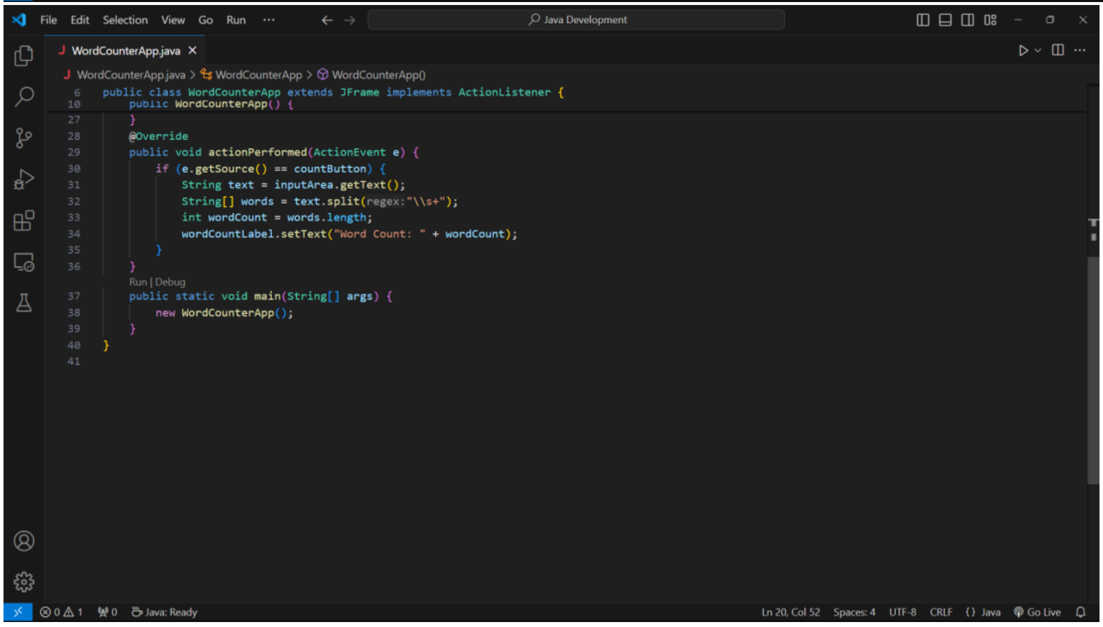


4. WORD_COUNTER_APP. INPUT: -



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

public class WordCounterApp extends JFrame implements ActionListener {
    private JTextArea inputArea;
    private JLabel wordCountLabel;
    private JButton countButton;
    public WordCounterApp() {
        super(title:"Word Counter");
        inputArea = new JTextArea();
        inputArea.setFont(new Font(name:"Monospaced", Font.PLAIN, size:14));
        JScrollPane scrollPane = new JScrollPane(inputArea);
        wordCountLabel = new JLabel(text:"Word Count: 0");
        countButton = new JButton(text:"Count Words");
        countButton.addActionListener(this);
        JPanel contentPanel = new JPanel();
        contentPanel.setLayout(new BorderLayout());
        contentPanel.add(scrollPane, BorderLayout.CENTER);
        contentPanel.add(wordCountLabel, BorderLayout.SOUTH);
        getContentPane().add(contentPanel, BorderLayout.CENTER);
        getContentPane().add(countButton, BorderLayout.SOUTH);
        setSize(width:500, height:300);
        setVisible(b:true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == countButton) {
            String text = inputArea.getText();
            String[] words = text.split(regex:"\\s+");
```

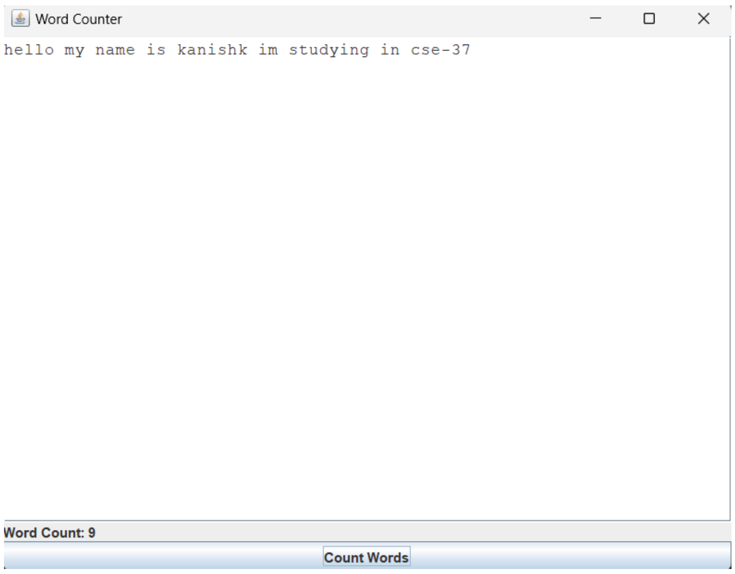
File Edit Selection View Go Run
Java Development
WordCounterApp.java
WordCounterApp.java > WordCounterApp > WordCounterApp()
Ln 20, Col 52 Spaces: 4 UTF-8 CRLF {} Java Go Live
Java: Ready



```java
public class WordCounterApp extends JFrame implements ActionListener {
    public WordCounterApp() {
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == countButton) {
            String text = inputArea.getText();
            String[] words = text.split(regex:"\\s+");
            int wordCount = words.length;
            wordCountLabel.setText("Word Count: " + wordCount);
        }
    }
    Run | Debug
    public static void main(String[] args) {
        new WordCounterApp();
    }
}
```

File Edit Selection View Go Run
Java Development
WordCounterApp.java
WordCounterApp.java > WordCounterApp > WordCounterApp()
Ln 20, Col 52 Spaces: 4 UTF-8 CRLF {} Java Go Live
Java: Ready

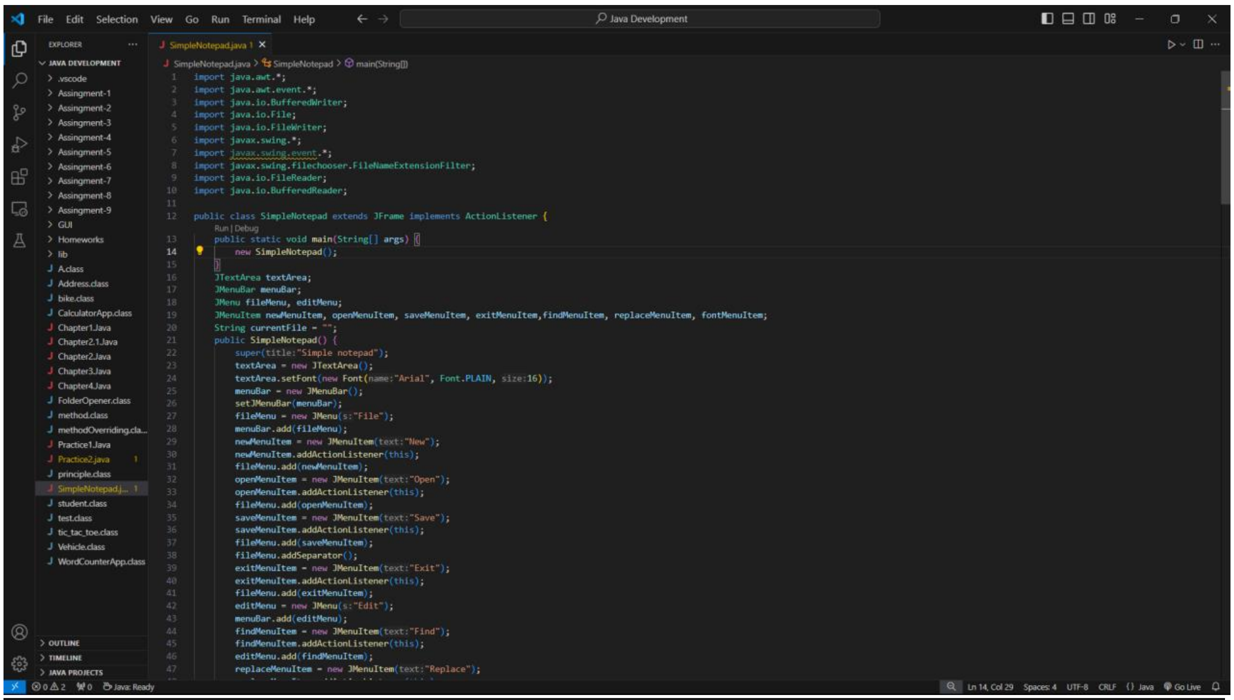
OUTPUT: -



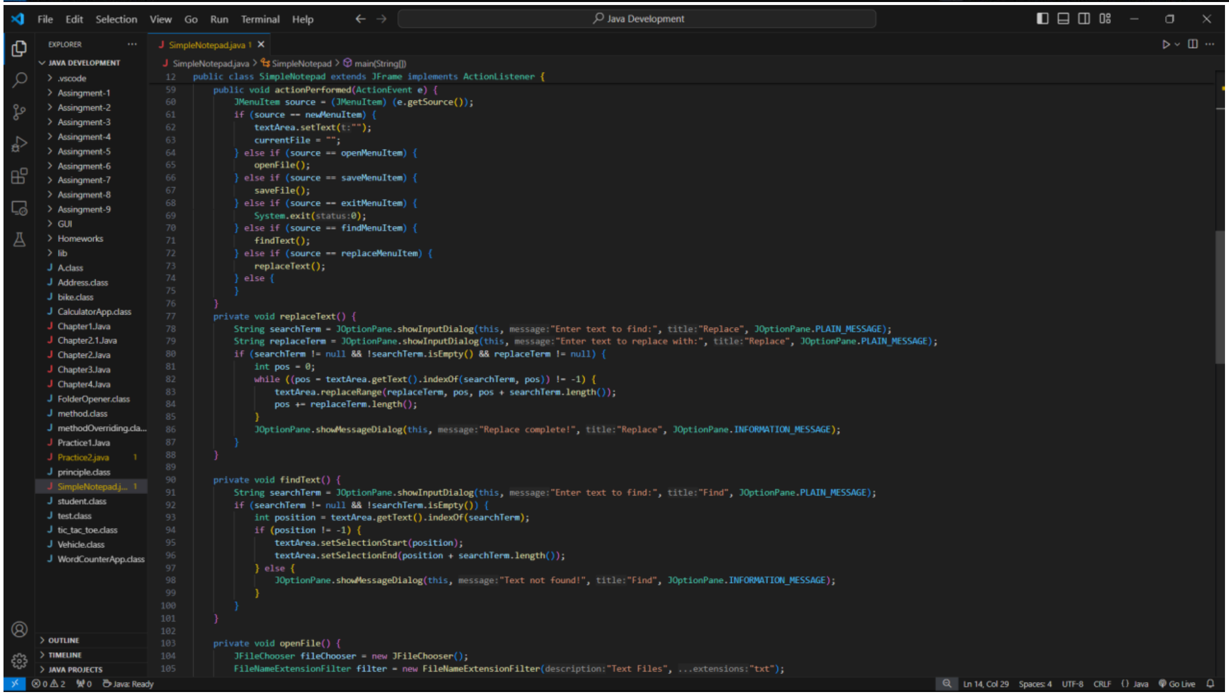
```
hello my name is kanishk im studying in cse-37
```

Word Count: 9

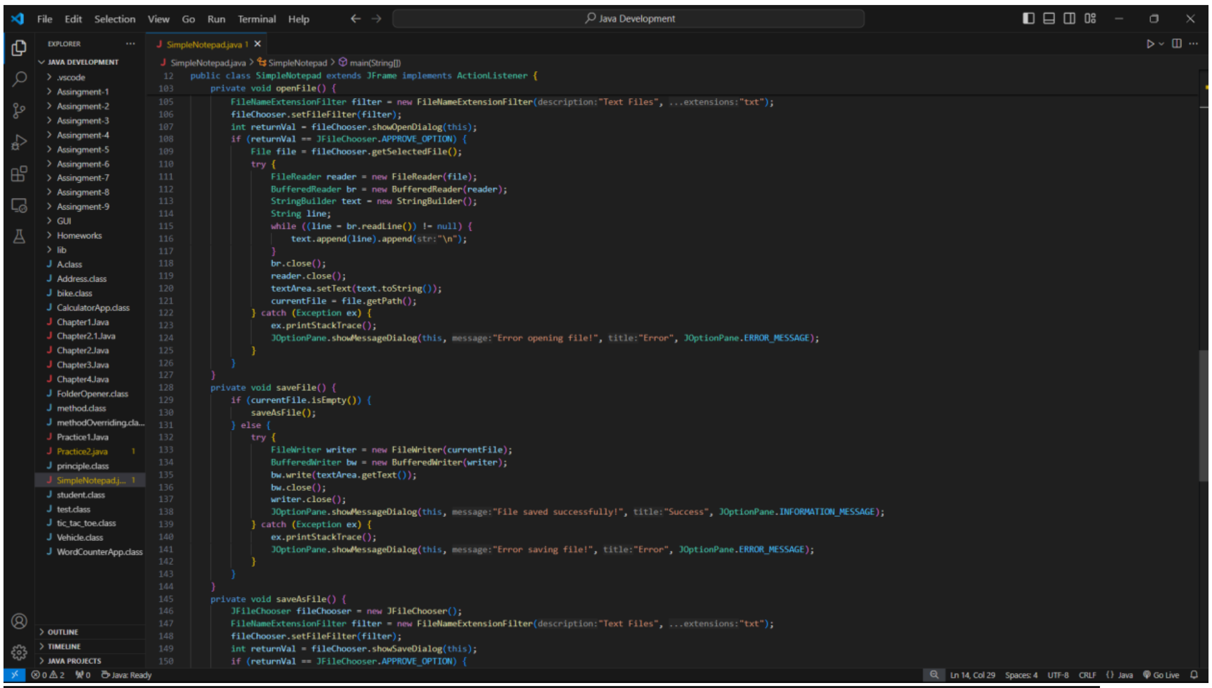
5.SIMPLE NOTEPAD INPUT: -



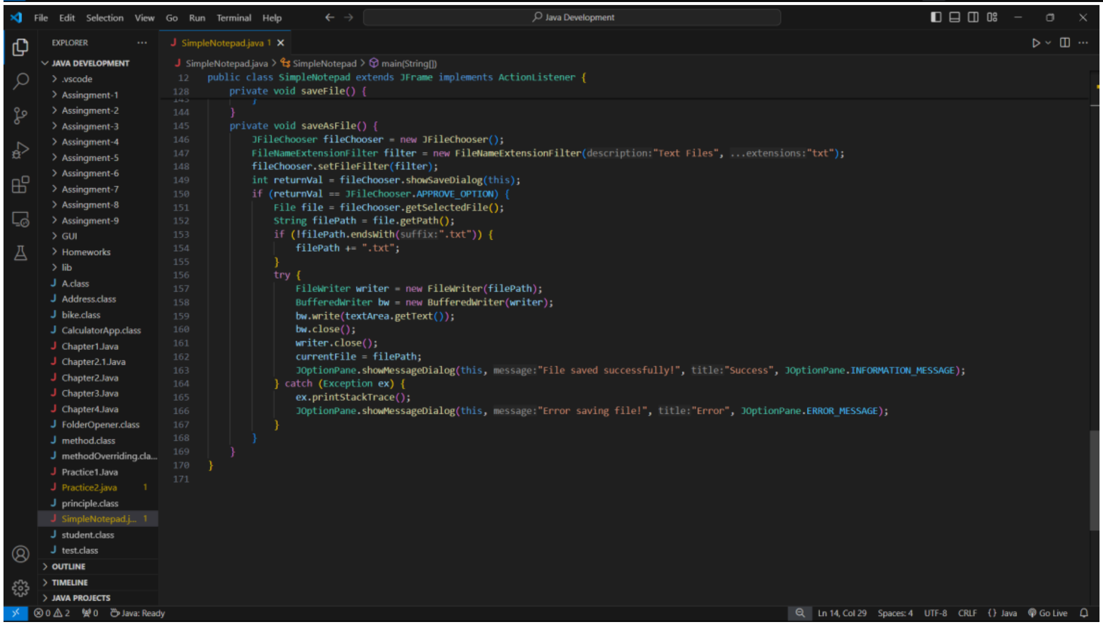
```java
import java.awt.*;
import java.awt.event.*;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.filechooser.FileNameExtensionFilter;
import java.io.FileReader;
import java.io.BufferedReader;

public class SimpleNotepad extends JFrame implements ActionListener {
    public static void main(String[] args) {
        new SimpleNotepad();
    }
    JTextArea textArea;
    JMenuBar menuBar;
    JMenu fileMenu, editMenu;
    JMenuItem newMenuItem, openMenuItem, saveMenuItem, exitMenuItem,findMenuItem, replaceMenuItem, fontMenuItem;
    String currentFile = "";
    public SimpleNotepad() {
        super(title:"Simple notepad");
        textArea = new JTextArea();
        textArea.setFont(new Font(name:"Arial", Font.PLAIN, size:16));
        menuBar = new JMenuBar();
        setJMenuBar(menuBar);
        fileMenu = new JMenu(s:"File");
        menuBar.add(fileMenu);
        newMenuItem = new JMenuItem(text:"New");
        newMenuItem.addActionListener(this);
        fileMenu.add(newMenuItem);
        openMenuItem = new JMenuItem(text:"Open");
        openMenuItem.addActionListener(this);
        fileMenu.add(openMenuItem);
        saveMenuItem = new JMenuItem(text:"Save");
        saveMenuItem.addActionListener(this);
        fileMenu.add(saveMenuItem);
        fileMenu.addSeparator();
        exitMenuItem = new JMenuItem(text:"Exit");
        exitMenuItem.addActionListener(this);
        fileMenu.add(exitMenuItem);
        editMenu = new JMenu(s:"Edit");
        menuBar.add(editMenu);
        findMenuItem = new JMenuItem(text:"Find");
        findMenuItem.addActionListener(this);
        editMenu.add(findMenuItem);
        replaceMenuItem = new JMenuItem(text:"Replace");
```



```java
public class SimpleNotepad extends JFrame implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        JMenuItem source = (JMenuItem) (e.getSource());
        if (source == newMenuItem) {
            textArea.setText(t:"");
            currentFile = "";
        } else if (source == openMenuItem) {
            openFile();
        } else if (source == saveMenuItem) {
            saveFile();
        } else if (source == exitMenuItem) {
            System.exit(status:0);
        } else if (source == findMenuItem) {
            findText();
        } else if (source == replaceMenuItem) {
            replaceText();
        } else {
        }
    }
    private void replaceText() {
        String searchTerm = JOptionPane.showInputDialog(this, message:"Enter text to find:", title:"Replace", JOptionPane.PLAIN_MESSAGE);
        String replaceTerm = JOptionPane.showInputDialog(this, message:"Enter text to replace with:", title:"Replace", JOptionPane.PLAIN_MESSAGE);
        if (searchTerm != null && !searchTerm.isEmpty() && replaceTerm != null) {
            int pos = 0;
            while ((pos = textArea.getText().indexOf(searchTerm, pos)) != -1) {
                textArea.replaceRange(replaceTerm, pos, pos + searchTerm.length());
                pos += replaceTerm.length();
            }
            JOptionPane.showMessageDialog(this, message:"Replace complete!", title:"Replace", JOptionPane.INFORMATION_MESSAGE);
        }
    }

    private void findText() {
        String searchTerm = JOptionPane.showInputDialog(this, message:"Enter text to find:", title:"Find", JOptionPane.PLAIN_MESSAGE);
        if (searchTerm != null && !searchTerm.isEmpty()) {
            int position = textArea.getText().indexOf(searchTerm);
            if (position != -1) {
                textArea.setSelectionStart(position);
                textArea.setSelectionEnd(position + searchTerm.length());
            } else {
                JOptionPane.showMessageDialog(this, message:"Text not found!", title:"Find", JOptionPane.INFORMATION_MESSAGE);
            }
        }
    }

    private void openFile() {
        JFileChooser fileChooser = new JFileChooser();
        FileNameExtensionFilter filter = new FileNameExtensionFilter(description:"Text Files", ...extensions:"txt");
```

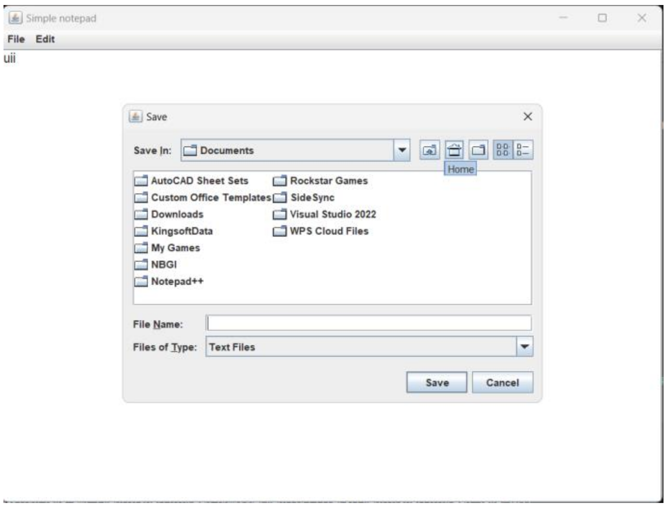


```
public class SimpleNotepad extends JFrame implements ActionListener {
    private void openFile() {
        FileNameExtensionFilter filter = new FileNameExtensionFilter(description:"Text Files", ...extensions:"txt");
        fileChooser.setFileFilter(filter);
        int returnVal = fileChooser.showOpenDialog(this);
        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            try {
                FileReader reader = new FileReader(file);
                BufferedReader br = new BufferedReader(reader);
                StringBuilder text = new StringBuilder();
                String line;
                while ((line = br.readLine()) != null) {
                    text.append(line).append(str:"\n");
                }
                br.close();
                reader.close();
                textArea.setText(text.toString());
                currentFile = file.getPath();
            } catch (Exception ex) {
                ex.printStackTrace();
                JOptionPane.showMessageDialog(this, message:"Error opening file!", title:"Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
    private void saveFile() {
        if (currentFile.isEmpty()) {
            saveAsFile();
        } else {
            try {
                FileWriter writer = new FileWriter(currentFile);
                BufferedWriter bw = new BufferedWriter(writer);
                bw.write(textArea.getText());
                bw.close();
                writer.close();
                JOptionPane.showMessageDialog(this, message:"File saved successfully!", title:"Success", JOptionPane.INFORMATION_MESSAGE);
            } catch (Exception ex) {
                ex.printStackTrace();
                JOptionPane.showMessageDialog(this, message:"Error saving file!", title:"Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
    private void saveAsFile() {
        JFileChooser fileChooser = new JFileChooser();
        FileNameExtensionFilter filter = new FileNameExtensionFilter(description:"Text Files", ...extensions:"txt");
        fileChooser.setFileFilter(filter);
        int returnVal = fileChooser.showSaveDialog(this);
        if (returnVal == JFileChooser.APPROVE_OPTION) {
```



```
public class SimpleNotepad extends JFrame implements ActionListener {
    private void saveFile() {
    }
    private void saveAsFile() {
        JFileChooser fileChooser = new JFileChooser();
        FileNameExtensionFilter filter = new FileNameExtensionFilter(description:"Text Files", ...extensions:"txt");
        fileChooser.setFileFilter(filter);
        int returnVal = fileChooser.showSaveDialog(this);
        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            String filePath = file.getPath();
            if (!filePath.endsWith(suffix:".txt")) {
                filePath += ".txt";
            }
            try {
                FileWriter writer = new FileWriter(filePath);
                BufferedWriter bw = new BufferedWriter(writer);
                bw.write(textArea.getText());
                bw.close();
                writer.close();
                currentFile = filePath;
                JOptionPane.showMessageDialog(this, message:"File saved successfully!", title:"Success", JOptionPane.INFORMATION_MESSAGE);
            } catch (Exception ex) {
                ex.printStackTrace();
                JOptionPane.showMessageDialog(this, message:"Error saving file!", title:"Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}
```

OUTPUT: -



```
Simple notepad
File  Edit
uii

Save
Save In: Documents
AutoCAD Sheet Sets        Rockstar Games
Custom Office Templates   SideSync
Downloads                 Visual Studio 2022
KingsoftData              WPS Cloud Files
My Games
NBGI
Notepad++

File Name:
Files of Type: Text Files
                                  Save   Cancel
```