

Project Code:

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

public class Data_Com extends javax.swing.JFrame {

    private String A;
    private String B;

    public Data_Com() {
        initComponents();
    }

    private void initComponents() {
        JPanel jPanel = new JPanel();
        JButton resetButton = new JButton();
        JButton calculateButton = new JButton();
        JButton crcButton = new JButton(); // Added CRC Button
        JTextField resultField = new JTextField();
        JLabel outputLabel = new JLabel();
        JTextField inputField2 = new JTextField();
        JLabel inputLabel2 = new JLabel();
        JTextField flagField = new JTextField();
        JLabel flagLabel = new JLabel();
        JComboBox<String> combobox = new JComboBox<>();
        JLabel titleLabel = new JLabel();
        JTextField inputField1 = new JTextField();
        JLabel inputLabel1 = new JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        resetButton.setFont(new java.awt.Font("Tahoma", 1, 18));
        resetButton.setText("RESET");
        resetButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                resetActionPerformed(evt, inputField1, inputField2, resultField, flagField);
            }
        });

        calculateButton.setFont(new java.awt.Font("Tahoma", 1, 18));
        calculateButton.setText("CALCULATE");
        calculateButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
```

```

        calculateActionPerformed(evt, combobox, inputField1, inputField2, flagField,
resultField);
    }
});

crcButton.setFont(new java.awt.Font("Tahoma", 1, 18)); // CRC Button
crcButton.setText("CALCULATE CRC");
crcButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        crcActionPerformed(evt, inputField1, flagField, resultField);
    }
});

outputLabel.setFont(new java.awt.Font("Tahoma", 1, 14));
outputLabel.setText("OUTPUT:");

inputLabel2.setFont(new java.awt.Font("Tahoma", 1, 14));
inputLabel2.setText("INPUT:");

flagLabel.setFont(new java.awt.Font("Tahoma", 1, 14));
flagLabel.setText("FLAG:");

combobox.setFont(new java.awt.Font("Tahoma", 1, 18));
combobox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[]{
    "Select an Option", "Hamming Distance", "Parity Check", "Bit Stuffing", "Bit
Destuffing", "Character Stuffing", "Character Destuffing"
}));

titleLabel.setFont(new java.awt.Font("Tahoma", 1, 18));
titleLabel.setText("BDCLCI");

inputLabel1.setFont(new java.awt.Font("Tahoma", 1, 14));
inputLabel1.setText("INPUT:");

GroupLayout jPanelLayout = new GroupLayout(jPanel);
jPanel.setLayout(jPanelLayout);
jPanelLayout.setHorizontalGroup(
    jPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
        .addGroup(jPanelLayout.createSequentialGroup()
            .addComponent(inputLabel1)
            .addComponent(inputLabel2)
            .addComponent(outputLabel)
            .addGap(18, 18, 18)

```

```

.addGroup(jPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING, false)
    .addComponent(inputField1, GroupLayout.DEFAULT_SIZE, 300,
Short.MAX_VALUE)
    .addComponent(inputField2)
    .addComponent(resultField))
.addGap(30, 30, 30)
.addComponent(flagLabel)
.addGap(18, 18, 18)
.addComponent(flagField, GroupLayout.PREFERRED_SIZE, 300,
GroupLayout.PREFERRED_SIZE)
.addContainerGap(30, Short.MAX_VALUE))
.addGroup(jPanelLayout.createSequentialGroup()
.addGap(150, 150, 150)
.addComponent(resetButton, GroupLayout.PREFERRED_SIZE, 150,
GroupLayout.PREFERRED_SIZE)
.addGap(50, 50, 50)
.addComponent(calculateButton, GroupLayout.PREFERRED_SIZE, 150,
GroupLayout.PREFERRED_SIZE)
.addGap(50, 50, 50)
.addComponent(crcButton, GroupLayout.PREFERRED_SIZE, 200,
GroupLayout.PREFERRED_SIZE) // CRC Button
.addContainerGap(GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGroup(GroupLayout.Alignment.TRAILING,
jPanelLayout.createSequentialGroup()
.addContainerGap(GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
    .addGroup(GroupLayout.Alignment.TRAILING,
jPanelLayout.createSequentialGroup()
        .addComponent(titleLabel, GroupLayout.PREFERRED_SIZE, 200,
GroupLayout.PREFERRED_SIZE)
        .addGap(345, 345, 345))
    .addGroup(GroupLayout.Alignment.TRAILING,
jPanelLayout.createSequentialGroup()
        .addComponent(combobox, GroupLayout.PREFERRED_SIZE, 400,
GroupLayout.PREFERRED_SIZE)
        .addGap(250, 250, 250))))
);
jPanelLayout.setVerticalGroup(
    jPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
        .addGroup(jPanelLayout.createSequentialGroup()
            .addGap(30, 30, 30)
            .addComponent(titleLabel)
            .addGap(30, 30, 30)

```

```

        .addComponent(combobox, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
        .addGap(30, 30, 30)

    .addGroup(jPanelLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)
        .addComponent(inputLabel1)
        .addComponent(inputField1, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
        .addComponent(flagLabel)
        .addComponent(flagField, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
        .addGap(30, 30, 30)

    .addGroup(jPanelLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)
        .addComponent(inputLabel2)
        .addComponent(inputField2, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
        .addGap(30, 30, 30)

    .addGroup(jPanelLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)
        .addComponent(outputLabel)
        .addComponent(resultField, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
        .addGap(30, 30, 30)

    .addGroup(jPanelLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)
        .addComponent(resetButton)
        .addComponent(calculateButton)
        .addComponent(crcButton)) // CRC Button
        .addContainerGap(30, Short.MAX_VALUE))
    );

    GroupLayout layout = new GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(GroupLayout.Alignment.LEADING)
            .addComponent(jPanel, GroupLayout.DEFAULT_SIZE,
        GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(GroupLayout.Alignment.LEADING)
            .addComponent(jPanel, GroupLayout.DEFAULT_SIZE,
        GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();

```

```

    }

    private void resetActionPerformed(ActionEvent evt, JTextField inputField1, JTextField
inputField2, JTextField resultField, JTextField flagField) {
        inputField1.setText("");
        inputField2.setText("");
        resultField.setText("");
        flagField.setText("");
    }

    private void calculateActionPerformed(ActionEvent evt, JComboBox<String> combobox,
JTextField inputField1, JTextField inputField2, JTextField flagField, JTextField resultField) {
        String selectedOption = combobox.getSelectedItem().toString();
        switch (selectedOption) {
            case "Hamming Distance":
                handleHammingDistance(inputField1, inputField2, resultField);
                break;
            case "Parity Check":
                handleParityCheck(inputField1, inputField2, resultField);
                break;
            case "Bit Stuffing":
                handleBitStuffing(inputField1, flagField, resultField);
                break;
            case "Bit Destuffing":
                handleBitDestuffing(inputField1, flagField, resultField);
                break;
            case "Character Stuffing":
                handleCharacterStuffing(inputField1, flagField, resultField);
                break;
            case "Character Destuffing":
                handleCharacterDestuffing(inputField1, flagField, resultField);
                break;
            default:
                JOptionPane.showMessageDialog(this, "Please select a valid option.");
                break;
        }
    }

    private void handleHammingDistance(JTextField inputField1, JTextField inputField2,
JTextField resultField) {
        A = inputField1.getText();
        B = inputField2.getText();
        if (A.length() != B.length()) {
            resultField.setText("Not Possible");
        } else {
            int dist = 0;

```

```

        for (int i = 0; i < A.length(); i++) {
            if (A.charAt(i) != B.charAt(i)) {
                dist++;
            }
        }
        resultField.setText(Integer.toString(dist));
    }
}

private void handleParityCheck(JTextField inputField1, JTextField inputField2, JTextField
resultField) {
    // Add parity check logic here
}

private void handleBitStuffing(JTextField inputField1, JTextField flagField, JTextField
resultField) {
    // Example bit stuffing logic
    String input = inputField1.getText();
    String flag = flagField.getText();
    String stuffed = input.replaceAll("11111", "111110"); // Bit stuffing example
    resultField.setText(flag + stuffed + flag);
}

private void handleBitDestuffing(JTextField inputField1, JTextField flagField, JTextField
resultField) {
    // Example bit destuffing logic
    String input = inputField1.getText();
    String flag = flagField.getText();
    String destuffed = input.replaceAll("111110", "11111"); // Bit destuffing example
    resultField.setText(destuffed);
}

private void handleCharacterStuffing(JTextField inputField1, JTextField flagField,
JTextField resultField) {
    // Example character stuffing logic
    String input = inputField1.getText();
    String flag = flagField.getText();
    String stuffed = input.replace(flag, flag + "ESC"); // Character stuffing example
    resultField.setText(flag + stuffed + flag);
}

private void handleCharacterDestuffing(JTextField inputField1, JTextField flagField,
JTextField resultField) {
    // Example character destuffing logic
    String input = inputField1.getText();
    String flag = flagField.getText();

```

```

        String destuffed = input.replace(flag + "ESC", flag); // Character destuffing example
        resultField.setText(destuffed);
    }

    private void crcActionPerformed(ActionEvent evt, JTextField inputField1, JTextField
flagField, JTextField resultField) {
        String input = inputField1.getText();
        String flag = flagField.getText();

        // Example CRC calculation using CRC-32 polynomial (0xEDB88320)
        int crc = calculateCRC(input.getBytes());
        resultField.setText(Integer.toHexString(crc).toUpperCase());
    }

    private int calculateCRC(byte[] bytes) {
        int crc = 0xFFFFFFFF;
        for (byte b : bytes) {
            crc ^= (b & 0xFF);
            for (int i = 0; i < 8; i++) {
                if ((crc & 0x01) != 0) {
                    crc = (crc >>> 1) ^ 0xEDB88320; // CRC-32 Polynomial
                } else {
                    crc >>>= 1;
                }
            }
        }
        return ~crc;
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Data_Com().setVisible(true);
            }
        });
    }

    // Variables declaration
    private JComboBox<String> combobox;
    private JButton calculateButton;
    private JButton resetButton;
    private JButton crcButton; // CRC Button
    private JLabel outputLabel;
    private JLabel inputLabel1;
    private JLabel inputLabel2;
    private JLabel flagLabel;

```

```
private JLabel titleLabel;  
private JTextField resultField;  
private JTextField inputField1;  
private JTextField inputField2;  
private JTextField flagField;
```

```
}
```