

# Assignment

Date: 27/11/22

---

<b>DATA STRUCTURES</b>	<b>2</b>
HOME PAGE	2
ARRAY DATA STRUCTURE	6
STACK DATA STRUCTURE	12
QUEUE DATA STRUCTURE	18
CIRCULAR QUEUE STRUCTURE DATA STRUCTURE	26
SINGLY LINKED LIST DATA STRUCTURE	33
DOUBLY LINKED LIST DATA STRUCTURE	41

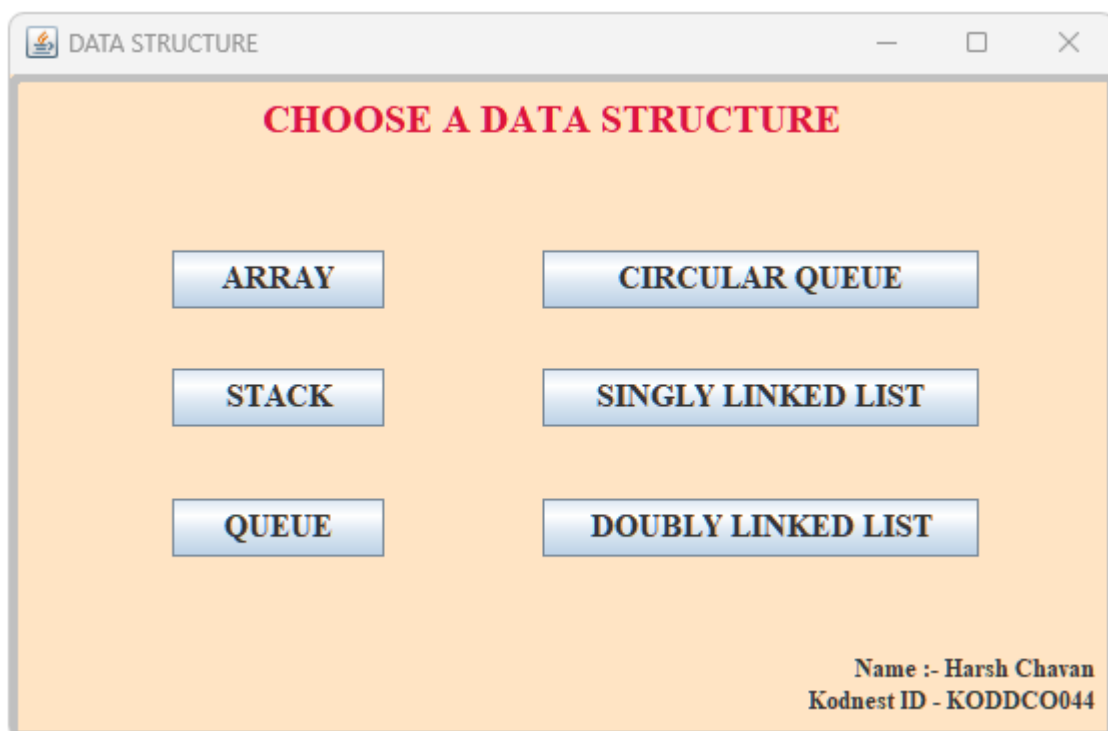
DOWNLOAD Jar File - [Link](#)

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

# DATA STRUCTURES

---

## HOME PAGE



```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import java.awt.Color;
import javax.swing.JLabel;
import javax.swing.JButton;
import java.awt.Font;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Canvas;
import javax.swing.border.LineBorder;
public class Home extends JFrame {
```

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

```

private JPanel contentPane;
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Home frame = new Home();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public Home() {
    setTitle("DATA STRUCTURE");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //          568, 371
    setBounds(100, 100, 568, 371);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(255, 228, 196));
    contentPane.setBorder(new LineBorder(new Color(192, 192, 192), 4, true));

    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("CHOOSE A DATA STRUCTURE");
    lblNewLabel.setForeground(new Color(220, 20, 60));
    lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 20));
    lblNewLabel.setBounds(126, 11, 289, 24);
    contentPane.add(lblNewLabel);

    JButton Array = new JButton("ARRAY");
    Array.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            // Array Window Opening
            // Array a = new Array();
            // a.setVisible(true);

            new Array().setVisible(true);
        }
    });
    Array.setFont(new Font("Times New Roman", Font.BOLD, 16));
    Array.setBounds(81, 88, 106, 29);
    contentPane.add(Array);

    JButton Stack = new JButton("STACK");
    Stack.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            // STACK Window Opening

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

        new Stack().setVisible(true);
    }
});
Stack.setFont(new Font("Times New Roman", Font.BOLD, 16));
Stack.setBounds(81, 147, 106, 29);
contentPane.add(Stack);

JButton btnQueue = new JButton("QUEUE");
btnQueue.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // QUEUE Window Opening
        new Queue().setVisible(true);
    }
});
btnQueue.setFont(new Font("Times New Roman", Font.BOLD, 16));
btnQueue.setBounds(81, 212, 106, 29);
contentPane.add(btnQueue);

JButton btnCircularQueue = new JButton("CIRCULAR QUEUE");
btnCircularQueue.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // CIRCULAR QUEUE Window Opening
        new CircularQueue().setVisible(true);
    }
});
btnCircularQueue.setFont(new Font("Times New Roman", Font.BOLD, 16));
btnCircularQueue.setBounds(266, 88, 218, 29);
contentPane.add(btnCircularQueue);

JButton btnSinglyLinkedList = new JButton("SINGLY LINKED LIST");
btnSinglyLinkedList.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // SINGLY LINKED LIST Window Opening
        new SinglyLinkedList().setVisible(true);
    }
});
btnSinglyLinkedList.setFont(new Font("Times New Roman", Font.BOLD, 16));
btnSinglyLinkedList.setBounds(266, 147, 218, 29);
contentPane.add(btnSinglyLinkedList);

JButton btnDoublyLinkedList = new JButton("DOUBLY LINKED LIST");
btnDoublyLinkedList.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // DOUBLY LINKED LIST Window Opening
        new DoublyLinkedList().setVisible(true);
    }
});
btnDoublyLinkedList.setFont(new Font("Times New Roman", Font.BOLD, 16));
btnDoublyLinkedList.setBounds(266, 212, 218, 29);

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```
contentPane.add(btnDoublyLinkedList);

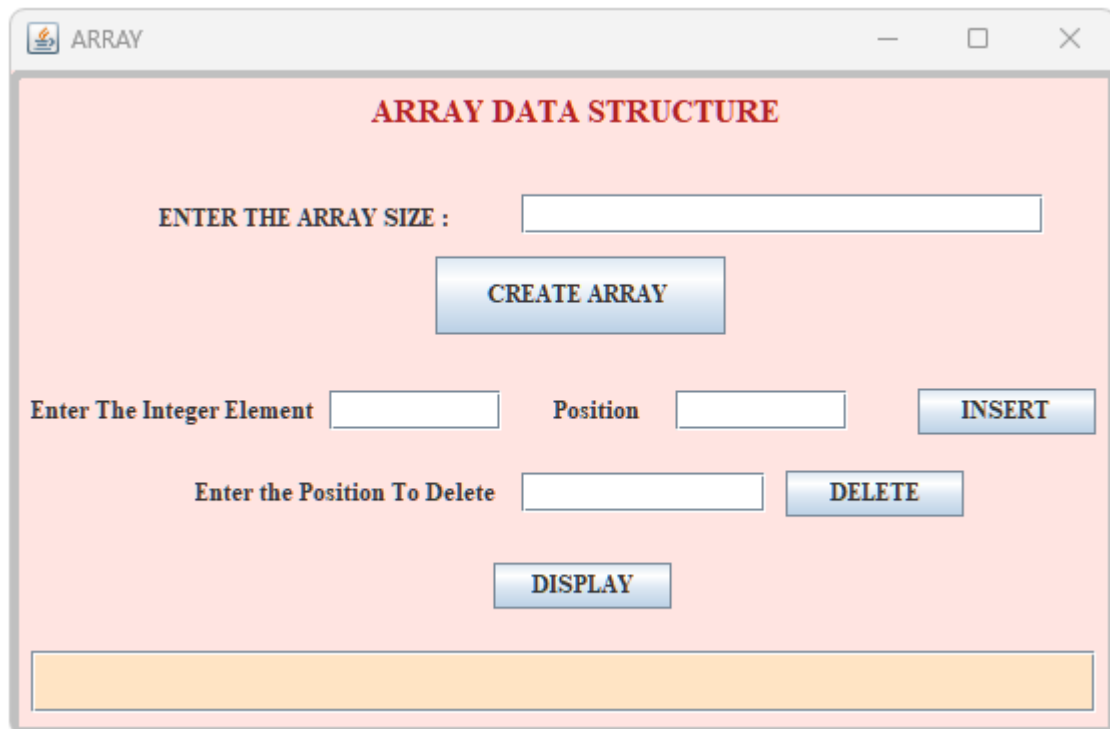
JLabel lblNewLabel_1 = new JLabel("Kodnest ID - KODDCO044");
lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_1.setBounds(399, 306, 143, 15);
contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("Name :- Harsh Chavan");
lblNewLabel_2.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_2.setBounds(422, 290, 120, 15);
contentPane.add(lblNewLabel_2);
}
}
```

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

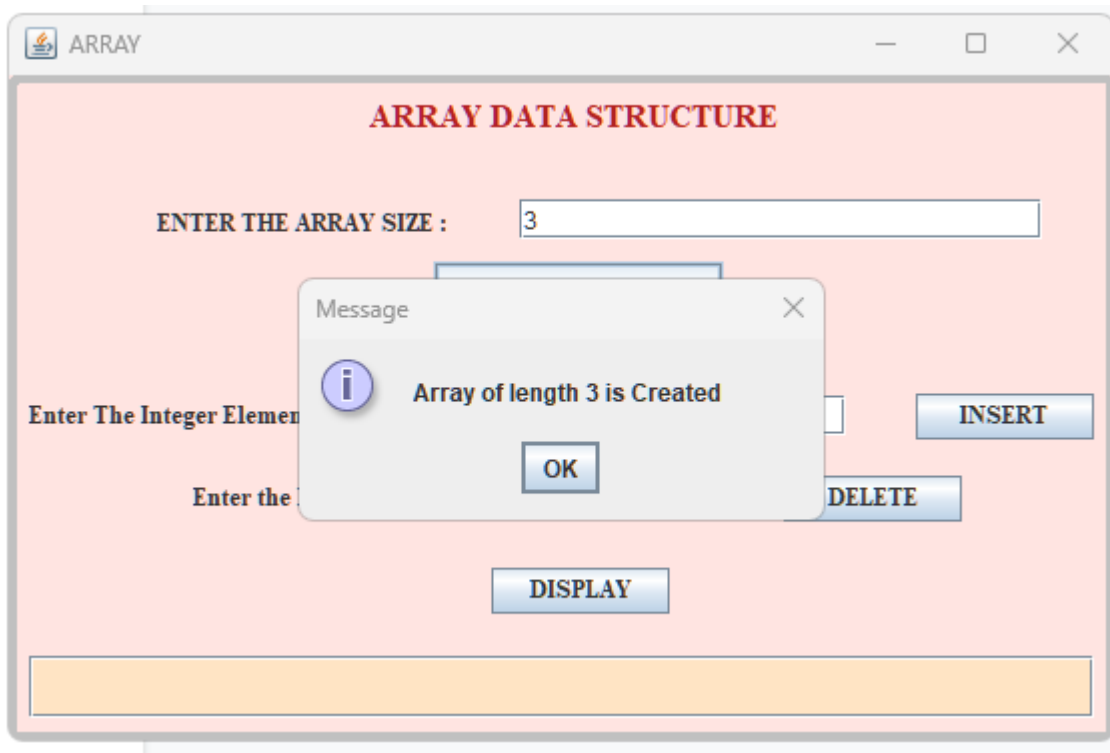
## ARRAY DATA STRUCTURE

---



The screenshot shows a window titled "ARRAY" with the following elements:

- ARRAY DATA STRUCTURE** (Title)
- ENTER THE ARRAY SIZE :** (Label)  (Input field)
- CREATE ARRAY** (Button)
- Enter The Integer Element** (Label)  (Input field)
- Position** (Label)  (Input field)
- INSERT** (Button)
- Enter the Position To Delete** (Label)  (Input field)
- DELETE** (Button)
- DISPLAY** (Button)
- A large empty text area at the bottom.



The screenshot shows the same window as above, but with the "ENTER THE ARRAY SIZE" field containing the value "3". A "Message" dialog box is displayed in the center, containing the text "Array of length 3 is Created" and an "OK" button. The dialog box is titled "Message" and has a close button (X) in the top right corner.

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

ARRAY

### ARRAY DATA STRUCTURE

ENTER THE ARRAY SIZE :

Enter The Integer Element

Enter the

**Message**

**Element 100 is Inserted at 0**

OK

INSERT

DELETE

DISPLAY

ARRAY

### ARRAY DATA STRUCTURE

ENTER THE ARRAY SIZE :

CREATE ARRAY

Enter The Integer Element  Position

INSERT

Enter the Position To Delete

DELETE

DISPLAY

**100 200 300**

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

**ARRAY DATA STRUCTURE**

ENTER THE ARRAY SIZE :

**CREATE ARRAY**

Enter The Integer Element  Position  **INSERT**

Enter the Position To Delete  **DELETE**

**DISPLAY**

**100 0 300**

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextArea;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.border.LineBorder;
public class Array extends JFrame {
    private JPanel contentPane;
    private JTextField tf2;
    private JTextField tf3;
    private JTextField tf4;
    private JTextField tf1;
    private int arr[];
    private JTextField displaybox;
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Array frame = new Array();
                    frame.setVisible(true);
```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan



```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

});

}

public Array() {
    setTitle("ARRAY ");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 568, 371);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(255, 228, 225));
    contentPane.setBorder(new LineBorder(new Color(192, 192, 192), 4, true));

    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("ARRAY DATA STRUCTURE");
    lblNewLabel.setForeground(new Color(178, 34, 34));
    lblNewLabel.setBackground(new Color(230, 230, 250));
    lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 16));
    lblNewLabel.setBounds(180, 11, 204, 20);
    contentPane.add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("ENTER THE ARRAY SIZE :");
    lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 12));
    lblNewLabel_1.setBounds(74, 67, 145, 15);
    contentPane.add(lblNewLabel_1);

    JLabel lblNewLabel_2 = new JLabel("Enter the Position To Delete ");
    lblNewLabel_2.setFont(new Font("Times New Roman", Font.BOLD, 12));
    lblNewLabel_2.setBounds(92, 204, 153, 15);
    contentPane.add(lblNewLabel_2);

    JLabel lblNewLabel_3 = new JLabel("Enter The Integer Element ");
    lblNewLabel_3.setFont(new Font("Times New Roman", Font.BOLD, 12));
    lblNewLabel_3.setBounds(10, 163, 144, 15);
    contentPane.add(lblNewLabel_3);

    JLabel lblNewLabel_4 = new JLabel("Position");
    lblNewLabel_4.setFont(new Font("Times New Roman", Font.BOLD, 12));
    lblNewLabel_4.setBounds(271, 163, 43, 15);
    contentPane.add(lblNewLabel_4);

    tf2 = new JTextField();
    tf2.setBounds(159, 160, 86, 20);
    contentPane.add(tf2);
    tf2.setColumns(10);

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

tf3 = new JTextField();
tf3.setBounds(332, 160, 86, 20);
contentPane.add(tf3);
tf3.setColumns(10);

tf4 = new JTextField();
tf4.setBounds(255, 201, 122, 20);
contentPane.add(tf4);
tf4.setColumns(10);

JButton btnNewButton = new JButton("CREATE ARRAY ");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //Create an Array
        //
        String length = tf1.getText();
        int len = Integer.valueOf(length);
        //OR
        int len = Integer.valueOf(tf1.getText());
        arr = new int[len];
        String message = "Array of length "+len+" is Created ";
        JOptionPane.showMessageDialog(contentPane, message);

    }
});
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnNewButton.setBounds(212, 93, 145, 39);
contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("INSERT");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //inserting the array
        int elem = Integer.valueOf(tf2.getText());
        int pos = Integer.valueOf(tf3.getText());
        arr[pos] = elem;
        String message = "Element "+elem+" is Inserted at "+pos+" ";
        JOptionPane.showMessageDialog(contentPane, message);
        tf2.setText("");
        tf3.setText("");
    }
});
btnNewButton_1.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnNewButton_1.setBounds(453, 159, 89, 23);
contentPane.add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("DELETE");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

        //deleting the array
        int del_pos = Integer.valueOf(tf4.getText());
        arr[del_pos] = 0;
        String message = "Element is Deleted at position "+del_pos+" ";
        JOptionPane.showMessageDialog(contentPane, message);
        tf4.setText("");
    }

});
btnNewButton_2.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnNewButton_2.setBounds(387, 200, 89, 23);
contentPane.add(btnNewButton_2);

JButton btnNewButton_3 = new JButton("DISPLAY");
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //displaying the array
        String display = "";
        for (int i = 0; i <= arr.length-1; i++) {
            display = display+" "+arr[i];
        }
        displaybox.setText(display);
    }
});
btnNewButton_3.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnNewButton_3.setBounds(241, 246, 89, 23);
contentPane.add(btnNewButton_3);

tf1 = new JTextField();
tf1.setColumns(10);
tf1.setBounds(255, 62, 261, 20);
contentPane.add(tf1);

displaybox = new JTextField();
displaybox.setFont(new Font("Times New Roman", Font.BOLD, 20));
displaybox.setBackground(new Color(255, 228, 196));
displaybox.setBounds(10, 290, 532, 31);
contentPane.add(displaybox);
displaybox.setColumns(10);
}
}

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

## STACK DATA STRUCTURE

---

The screenshot shows a window titled "STACK" with a light pink background. At the top, the text "STACK DATA STRUCTURE" is displayed in red. Below this, there is a label "Enter The Stack Size :" followed by an empty text input field. Underneath the input field is a blue button labeled "CREATE STACK". Further down, there is a label "Enter an Element :" followed by another empty text input field. To the right of this input field is a blue button labeled "PUSH". Below the "Enter an Element" input field are two more blue buttons, "POP" and "DISPLAY", stacked vertically. At the bottom of the window is a wide, empty yellow rectangular area.

This screenshot shows the same "STACK" window, but with the "Enter The Stack Size" input field now containing the number "3". A modal message box is centered on the screen. The message box has a title bar that says "Message" and a close button (X). It contains an information icon (i) and the text "Stack is created size of 3". Below the text is an "OK" button. In the background, the "Enter an Element" input field and the "PUSH", "POP", and "DISPLAY" buttons are still visible, though partially obscured by the message box. The yellow area at the bottom remains empty.

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

STACK

### STACK DATA STRUCTURE

Enter The Stack Size :

Enter an Element :

**Message** ×

**i** Push is Successful

**OK**

**PUSH**

**DISPLAY**

STACK

### STACK DATA STRUCTURE

Enter The Stack Size :

**CREATE STACK**

Enter an Element :

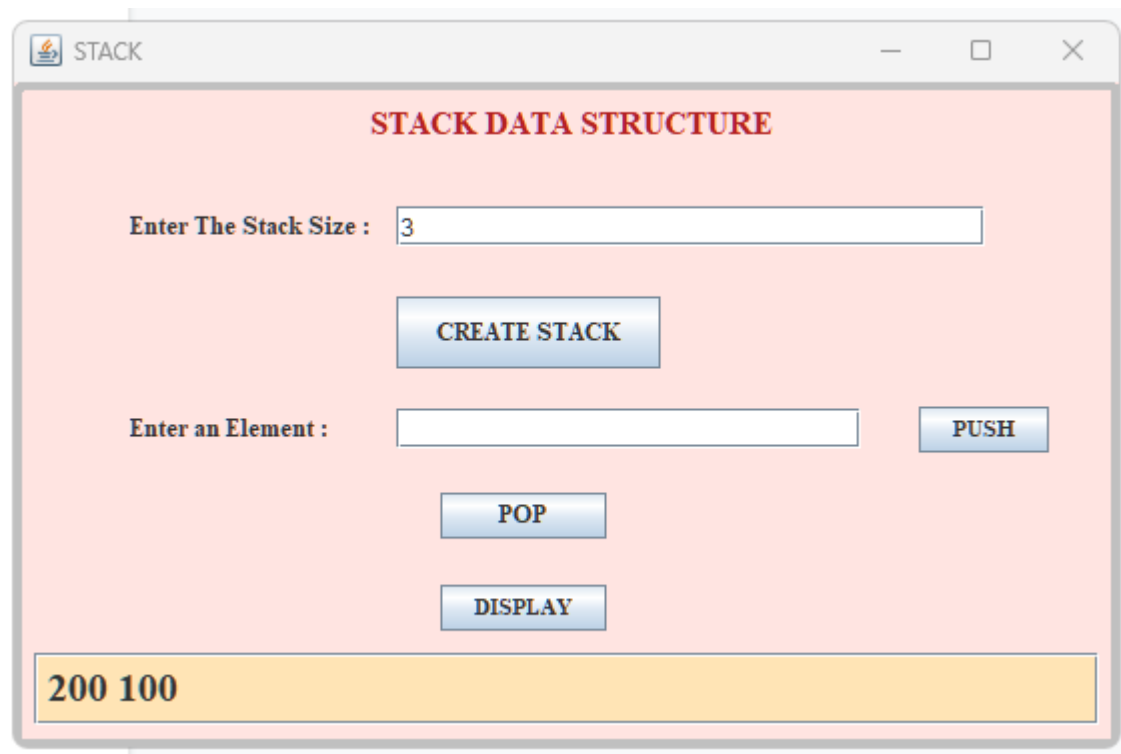
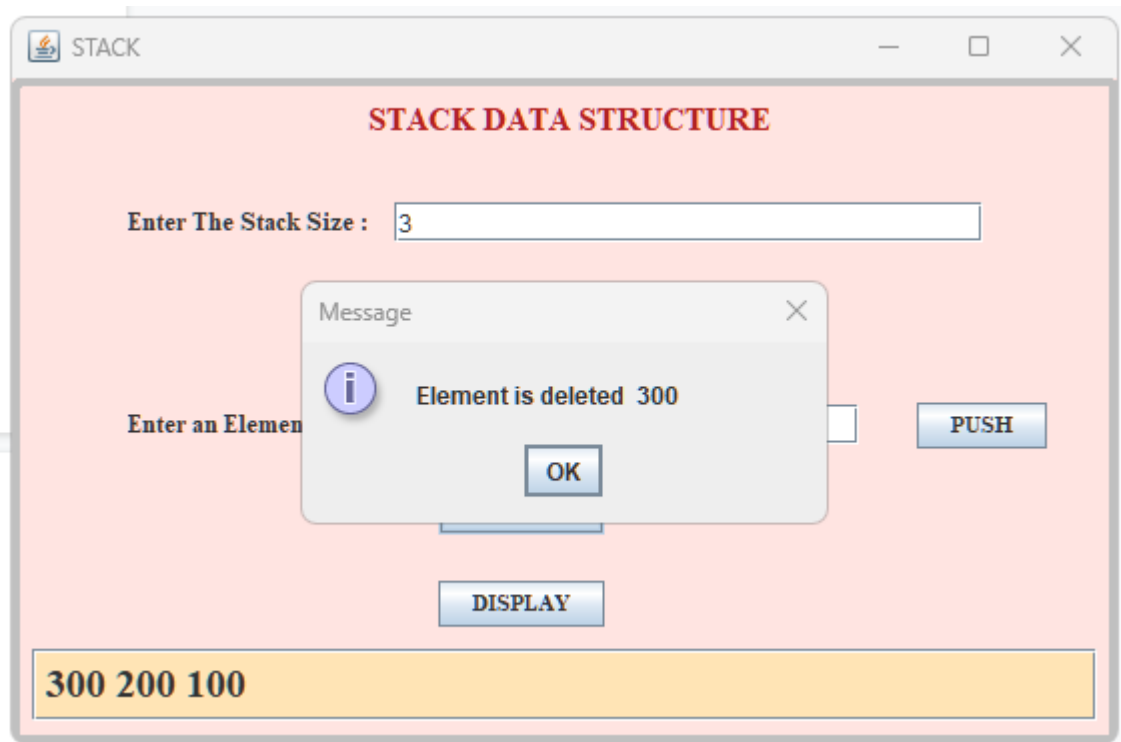
**PUSH**

**POP**

**DISPLAY**

**300 200 100**

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan



```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
```

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

```

import javax.swing.JButton;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextArea;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.border.LineBorder;
public class Stack extends JFrame {
    private JPanel contentPane;
    private JTextField tf1;
    private JTextField tf2;
    private JTextField displaybox;
    private int arr[];
    private int size;
    private int top = -1;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Stack frame = new Stack();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public Stack() {
        setTitle("STACK");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // 568, 371
        setBounds(100, 100, 568, 371);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(255, 228, 225));
        contentPane.setBorder(new LineBorder(new Color(192, 192, 192), 4, true));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("STACK DATA STRUCTURE");
        lblNewLabel.setForeground(new Color(178, 34, 34));
        lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 16));
        lblNewLabel.setBounds(178, 11, 201, 20);
        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("Enter The Stack Size :");
        lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 12));

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

lblNewLabel_1.setBounds(58, 65, 120, 15);
contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("Enter an Element :");
lblNewLabel_2.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_2.setBounds(58, 166, 99, 15);
contentPane.add(lblNewLabel_2);

tf1 = new JTextField();
tf1.setBounds(191, 62, 294, 20);
contentPane.add(tf1);
tf1.setColumns(10);

tf2 = new JTextField();
tf2.setColumns(10);
tf2.setBounds(191, 163, 232, 20);
contentPane.add(tf2);

JButton btnNewButton = new JButton("POP");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //POP
        if (top== -1) {
            JOptionPane.showMessageDialog(contentPane, "Stack is Empty");
        } else {
            String message = "Element is deleted " + arr[top];
            JOptionPane.showMessageDialog(contentPane, message);
            --top;
        }
    }
});
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnNewButton.setBounds(213, 205, 83, 23);
contentPane.add(btnNewButton);

JButton btnDisplay = new JButton("DISPLAY");
btnDisplay.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //DISPLAY
        String display = "";
        if (top== -1) {
            JOptionPane.showMessageDialog(contentPane, "Display is Not
Possible");
        } else {
            for (int i = top; i >= 0; i--) {
                display = display + " " + arr[i];
            }
        }
        displaybox.setText(display);
    }
});

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan



```

    }
});
btnDisplay.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnDisplay.setBounds(213, 251, 83, 23);
contentPane.add(btnDisplay);

JButton btnCreateStack = new JButton("CREATE STACK");
btnCreateStack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //Create Stack
        size = Integer.valueOf(tf1.getText());
        arr = new int[size];
        String message = "Stack is created size of "+size;
        JOptionPane.showMessageDialog(contentPane, message);

    }
});
btnCreateStack.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnCreateStack.setBounds(191, 107, 132, 36);
contentPane.add(btnCreateStack);

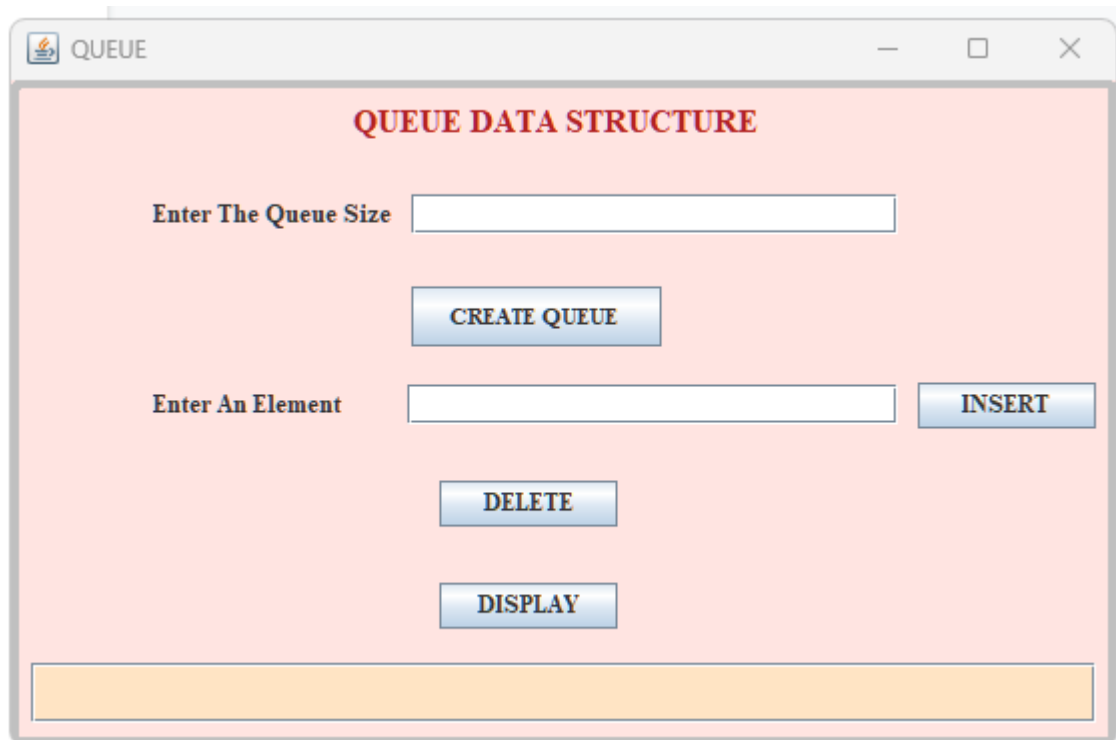
JButton btnPush = new JButton("PUSH");
btnPush.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //PUSH
        int elem;
        if(top == size - 1) {
            JOptionPane.showMessageDialog(contentPane, "Push is not
Possible");
        }
        else {
            elem = Integer.valueOf(tf2.getText());
            ++top;
            arr[top] = elem;
            JOptionPane.showMessageDialog(contentPane, "Push is Successful");
            tf2.setText("");
        }
    }
});
btnPush.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnPush.setBounds(452, 162, 65, 23);
contentPane.add(btnPush);
displaybox = new JTextField();
displaybox.setFont(new Font("Times New Roman", Font.BOLD, 20));
displaybox.setBackground(new Color(255, 228, 181));
displaybox.setBounds(10, 285, 532, 36);
contentPane.add(displaybox);
displaybox.setColumns(10);
}
}

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

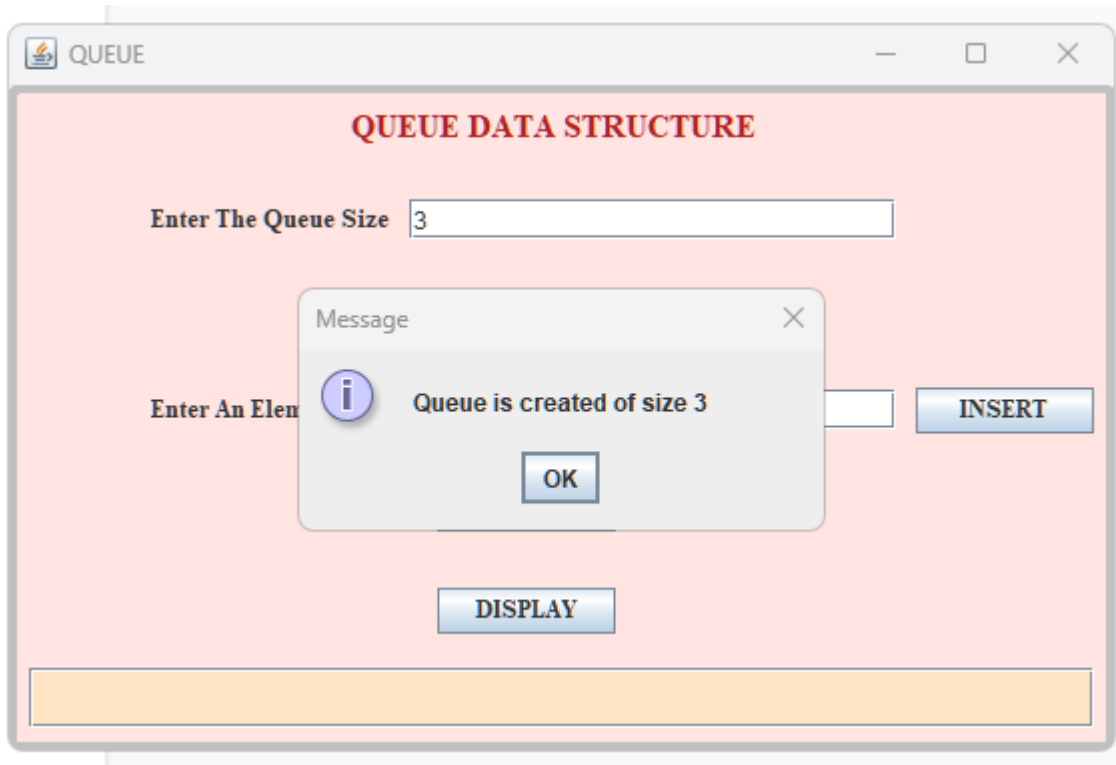
## QUEUE DATA STRUCTURE

---



The screenshot shows a window titled "QUEUE" with a red header "QUEUE DATA STRUCTURE". It contains the following elements:

- A label "Enter The Queue Size" followed by an empty text input field.
- A blue button labeled "CREATE QUEUE" below the first input field.
- A label "Enter An Element" followed by an empty text input field.
- A blue button labeled "INSERT" to the right of the second input field.
- A blue button labeled "DELETE" below the second input field.
- A blue button labeled "DISPLAY" below the "DELETE" button.
- A large empty orange rectangular area at the bottom of the window.



The screenshot shows the same window as above, but with the following changes:

- The "Enter The Queue Size" input field now contains the value "3".
- A modal dialog box titled "Message" is displayed in the center. It contains an information icon (i), the text "Queue is created of size 3", and an "OK" button.
- The "Enter An Element" input field is now empty.
- The "INSERT" and "DISPLAY" buttons remain visible.
- The large orange area at the bottom remains empty.

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

QUEUE

### QUEUE DATA STRUCTURE

Enter The Queue Size

Enter An Element

**Message** ✕

**i** Inserted Element is 100

**OK**

**INSERT**

**DISPLAY**

QUEUE

### QUEUE DATA STRUCTURE

Enter The Queue Size

**CREATE QUEUE**

Enter An Element

**INSERT**

**DELETE**

**DISPLAY**

**100 200 300**

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

QUEUE

### QUEUE DATA STRUCTURE

Enter The Queue Size

Enter An Element

**Message** ×

**i** Element is deleted is 200

**OK**

**INSERT**

**DISPLAY**

**200 300**

QUEUE

### QUEUE DATA STRUCTURE

Enter The Queue Size

**CREATE QUEUE**

Enter An Element

**INSERT**

**DELETE**

**DISPLAY**

**300**

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

```

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JTextArea;
import java.awt.Font;
import java.awt.Color;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.border.LineBorder;

public class Queue extends JFrame {

    private JPanel contentPane;
    private JTextField tf1;
    private JTextField tf2;
    private JTextField displaybox;
    private int q[];
    private int f = 0;
    private int r = -1;
    private int size;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Queue frame = new Queue();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public Queue() {
        setTitle("QUEUE");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // 568, 371
        setBounds(100, 100, 568, 371);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(255, 228, 225));
        contentPane.setBorder(new LineBorder(new Color(192, 192, 192), 4, true));
    }

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("QUEUE DATA STRUCTURE");
lblNewLabel.setForeground(new Color(178, 34, 34));
lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 16));
lblNewLabel.setBounds(171, 11, 202, 20);
contentPane.add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("Enter The Queue Size");
lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_1.setBounds(71, 60, 119, 15);
contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("Enter An Element ");
lblNewLabel_2.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_2.setBounds(71, 155, 97, 15);
contentPane.add(lblNewLabel_2);

tf1 = new JTextField();
tf1.setBounds(200, 57, 243, 20);
contentPane.add(tf1);
tf1.setColumns(10);

tf2 = new JTextField();
tf2.setColumns(10);
tf2.setBounds(198, 152, 245, 20);
contentPane.add(tf2);

JButton btnNewButton = new JButton("CREATE QUEUE ");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //create queue
        size = Integer.valueOf(tf1.getText());
        q = new int[size];
        String create_msg = "Queue is created of size "+size;
        JOptionPane.showMessageDialog(contentPane,create_msg);
    }
});
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton.setBounds(200, 103, 125, 30);
contentPane.add(btnNewButton);

JButton btnInsert = new JButton("INSERT");
btnInsert.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //insert
        int elem;
        if(r == size-1) {

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

Possible");

JOptionPane.showMessageDialog(contentPane,"Insertion is not
Possible");

        }else {
            elem = Integer.valueOf(tf2.getText());
            ++r;
            q[r] = elem;
            String insert_msg = "Inserted Element is "+elem;
            JOptionPane.showMessageDialog(contentPane,insert_msg);
            tf2.setText("");
        }
    }

});
btnInsert.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnInsert.setBounds(453, 151, 89, 23);
contentPane.add(btnInsert);

JButton btnDelete = new JButton("DELETE");
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //delete
        if(f>r || r == -1) {
            JOptionPane.showMessageDialog(contentPane,"Deletion is not
Possible");
        }else {
            String del_pos = "Element is deleted is "+q[f];
            JOptionPane.showMessageDialog(contentPane,del_pos);
            ++f;
        }
    }
});
btnDelete.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnDelete.setBounds(214, 200, 89, 23);
contentPane.add(btnDelete);

JButton btnDisplay = new JButton("DISPLAY");
btnDisplay.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //display
        if( f > r || r == -1) {
            JOptionPane.showMessageDialog(contentPane,"Display is not
Possible");

            displaybox.setText("");
        }else {
            String msg = "";
            for (int i = f; i <=r; i++) {
                msg = msg + " " +q[i];
            }
            displaybox.setText(msg);
        }
    }
});

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

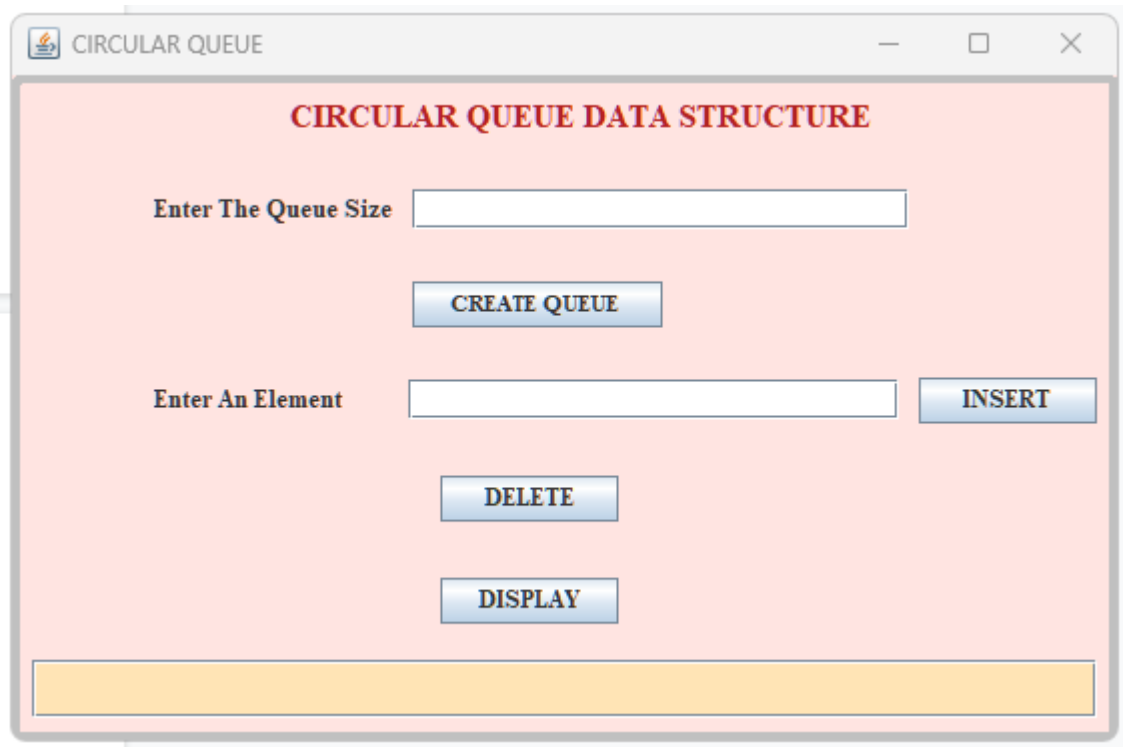
```
        }  
    });  
    btnDisplay.setFont(new Font("Times New Roman", Font.BOLD, 12));  
    btnDisplay.setBounds(214, 251, 89, 23);  
    contentPane.add(btnDisplay);  
    displaybox = new JTextField();  
    displaybox.setFont(new Font("Times New Roman", Font.BOLD, 20));  
    displaybox.setBackground(new Color(255, 228, 196));  
    displaybox.setBounds(10, 291, 532, 30);  
    contentPane.add(displaybox);  
    displaybox.setColumns(10);  
}  
}
```

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

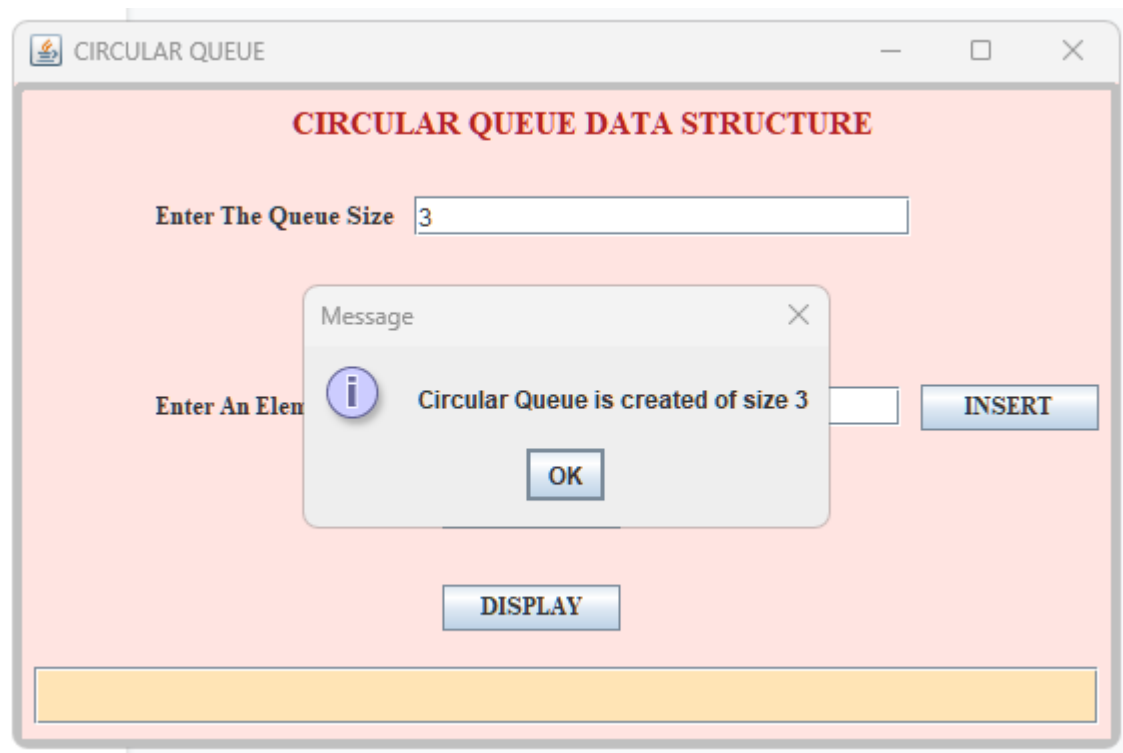


## CIRCULAR QUEUE STRUCTURE DATA STRUCTURE

---



The screenshot shows a window titled "CIRCULAR QUEUE" with a pink background. At the top, it says "CIRCULAR QUEUE DATA STRUCTURE" in red. Below this, there are two input fields: "Enter The Queue Size" and "Enter An Element". To the right of the "Enter The Queue Size" field is a "CREATE QUEUE" button. To the right of the "Enter An Element" field is an "INSERT" button. Below the "INSERT" button are two more buttons: "DELETE" and "DISPLAY". At the bottom of the window is a long yellow rectangular area, likely for displaying the queue's state.



This screenshot shows the same "CIRCULAR QUEUE" window, but now the "Enter The Queue Size" field contains the number "3". A message dialog box is open in the center of the window. The dialog box has a title bar that says "Message" and a close button (X). It contains an information icon (i) and the text "Circular Queue is created of size 3". Below the text is an "OK" button. The "Enter An Element" field and the "INSERT", "DELETE", and "DISPLAY" buttons are still visible. The yellow area at the bottom is empty.

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

CIRCULAR QUEUE

### CIRCULAR QUEUE DATA STRUCTURE

Enter The Queue Size

Enter An Element

**Message**

Inserted Element is 100

OK

INSERT

DISPLAY

CIRCULAR QUEUE

### CIRCULAR QUEUE DATA STRUCTURE

Enter The Queue Size

CREATE QUEUE

Enter An Element

INSERT

DELETE

DISPLAY

100 200 300

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

CIRCULAR QUEUE DATA STRUCTURE

Enter The Queue Size

Enter An Element

Message: Element is deleted is 200

OK

INSERT

DISPLAY

200 300

CIRCULAR QUEUE DATA STRUCTURE

Enter The Queue Size

CREATE QUEUE

Enter An Element

INSERT

DELETE

DISPLAY

300

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

```

import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.SwingConstants;
import javax.swing.border.LineBorder;

public class CircularQueue extends JFrame {

```

```

    private JPanel contentPane;
    private JTextField tf1;
    private JTextField textField_1;
    private JTextField tf2;
    private JTextField displaybox;
    private int cq[];
    private int f = 0;
    private int r = -1;
    private int size;
    private int count = 0;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    CircularQueue frame = new CircularQueue();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public CircularQueue() {

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

setTitle("CIRCULAR QUEUE");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// 568, 371
setBounds(100, 100, 568, 371);
contentPane = new JPanel();
contentPane.setBackground(new Color(255, 228, 225));
contentPane.setBorder(new LineBorder(new Color(192, 192, 192), 4, true));

setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("CIRCULAR QUEUE DATA STRUCTURE");
lblNewLabel.setForeground(new Color(178, 34, 34));
lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 16));
lblNewLabel.setBounds(139, 11, 290, 20);
contentPane.add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("Enter The Queue Size");
lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_1.setBounds(71, 60, 119, 15);
contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("Enter An Element ");
lblNewLabel_2.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_2.setBounds(71, 155, 97, 15);
contentPane.add(lblNewLabel_2);

tf2 = new JTextField();
tf2.setColumns(10);
tf2.setBounds(198, 152, 245, 20);
contentPane.add(tf2);

JButton btnNewButton = new JButton("CREATE QUEUE ");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // create
        size = Integer.valueOf(tf1.getText());
        cq = new int[size];
        String create_msg = "Circular Queue is created of size " + size;
        JOptionPane.showMessageDialog(contentPane, create_msg);
    }
});
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton.setBounds(200, 103, 125, 23);
contentPane.add(btnNewButton);

JButton btnInsert = new JButton("INSERT");
btnInsert.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

        // insert
        int elem;
        if (count == size) {
            JOptionPane.showMessageDialog(contentPane, "Insertion is not
Possible");
        } else {
            elem = Integer.valueOf(tf2.getText());
            r = (r + 1) % size;
            cq[r] = elem;
            count++;
            String insert_msg = "Inserted Element is " + elem;
            JOptionPane.showMessageDialog(contentPane, insert_msg);
            tf2.setText("");
        }
    }

});
btnInsert.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnInsert.setBounds(453, 151, 89, 23);
contentPane.add(btnInsert);

JButton btnDelete = new JButton("DELETE");
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // delete
        if( count == 0 ) {
            JOptionPane.showMessageDialog(contentPane, "Deletion is not
Possible");
        } else {
            String del_pos = "Element is deleted is "+cq[f];
            JOptionPane.showMessageDialog(contentPane, del_pos);
            f = ( f + 1 ) % size;
            count--;
        }
    }
});
btnDelete.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnDelete.setBounds(214, 200, 89, 23);
contentPane.add(btnDelete);

JButton btnDisplay = new JButton("DISPLAY");
btnDisplay.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // display
        int f1 = f;
        if( count == 0 ) {
            JOptionPane.showMessageDialog(contentPane, "Display is not
Possible");
        }

        displaybox.setText("");
    } else {

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

        String msg = "";
        for (int i = 1; i <=count; i++) {
            msg = msg + " " +cq[f1];
            f1 = (f1 + 1) % size;
        }
        displaybox.setText(msg);
    }

});
btnDisplay.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnDisplay.setBounds(214, 251, 89, 23);
contentPane.add(btnDisplay);

tf1 = new JTextField();
tf1.setBounds(200, 57, 248, 20);
contentPane.add(tf1);
tf1.setColumns(10);

textField_1 = new JTextField();
textField_1.setColumns(10);
textField_1.setBounds(200, 152, 243, 20);
contentPane.add(textField_1);

displaybox = new JTextField();
displaybox.setFont(new Font("Times New Roman", Font.BOLD, 20));
displaybox.setBackground(new Color(255, 228, 181));
displaybox.setBounds(10, 292, 532, 29);
contentPane.add(displaybox);
displaybox.setColumns(10);
}

}

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

# SINGLY LINKED LIST DATA STRUCTURE

---

**SINGLY LINKED LIST DATA STRUCTURE**

Enter The Element  **INSERT REAR**

Enter The Element  **INSERT FRONT**

**DELETE REAR**

**DELETE FRONT**

**DISPLAY**

**SINGLY LINKED LIST DATA STRUCTURE**

Enter The Element  **INSERT REAR**

Enter The Element  **INSERT FRONT**

**DELETE REAR**

**DELETE FRONT**

**DISPLAY**

**Message**

**Inserted Element is 100 at Rear End**

**OK**

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan



**SINGLY LINKED LIST**

### SINGLY LINKED LIST DATA STRUCTURE

Enter The Element  **INSERT REAR**

Enter The Element  **INSERT FRONT**

**Message**

**i** Inserted Element is 200 at Front End

**OK**

**DISPLAY**

**SINGLY LINKED LIST**

### SINGLY LINKED LIST DATA STRUCTURE

Enter The Element  **INSERT REAR**

Enter The Element  **INSERT FRONT**

**DELETE REAR**

**DELETE FRONT**

**DISPLAY**

**200 100 300 400**

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

**SINGLY LINKED LIST DATA STRUCTURE**

Enter The Element  **INSERT REAR**

Enter The Element  **INSERT FRONT**

**Message** ×

**i** Element deleted is 400

**OK**

**DISPLAY**

**200 100 300 400**

**SINGLY LINKED LIST DATA STRUCTURE**

Enter The Element  **INSERT REAR**

Enter The Element  **INSERT FRONT**

**Message** ×

**i** Element deleted is 100

**OK**

**DISPLAY**

**100 300**

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

**SINGLY LINKED LIST DATA STRUCTURE**

Enter The Element  **INSERT REAR**

Enter The Element  **INSERT FRONT**

**DELETE REAR**

**DELETE FRONT**

**DISPLAY**

**300**

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import java.awt.Color;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Font;
import javax.swing.JTextArea;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.SwingConstants;
import javax.swing.border.LineBorder;

public class SinglyLinkedList extends JFrame {

    private JPanel contentPane;
    private JTextField tf1;
    private JTextField tf2;
```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

private JTextField displaybox;

// Creation of NODE
class Node {
    int data;
    Node link;
}
private Node first;

public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                SinglyLinkedList frame = new SinglyLinkedList();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public SinglyLinkedList() {
    setTitle("SINGLY LINKED LIST");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // 568, 371
    setBounds(100, 100, 568, 371);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(255, 228, 225));
    contentPane.setBorder(new LineBorder(new Color(192, 192, 192), 4, true));

    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("SINGLY LINKED LIST DATA STRUCTURE");
    lblNewLabel.setForeground(new Color(178, 34, 34));
    lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 18));
    lblNewLabel.setBounds(101, 11, 356, 21);
    contentPane.add(lblNewLabel);

    tf1 = new JTextField();
    tf1.setBounds(162, 52, 219, 20);
    contentPane.add(tf1);
    tf1.setColumns(10);

    tf2 = new JTextField();
    tf2.setColumns(10);
    tf2.setBounds(162, 104, 219, 20);
    contentPane.add(tf2);

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

JLabel lblNewLabel_1 = new JLabel("Enter The Element ");
lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_1.setBounds(38, 55, 103, 15);
contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_1_1 = new JLabel("Enter The Element ");
lblNewLabel_1_1.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_1_1.setBounds(38, 107, 103, 15);
contentPane.add(lblNewLabel_1_1);

JButton btnNewButton = new JButton("INSERT REAR");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Insert rear
        Node temp;
        int elem = Integer.valueOf(tf1.getText());
        String insert_elem = "Inserted Element is " +elem+" at Rear End";
        JOptionPane.showMessageDialog(contentPane, insert_elem);
        Node newnode = new Node();
        newnode.data = elem;
        newnode.link = null;
        if (first == null) {
            first = newnode;
        } else {
            temp = first;
            while (temp.link != null) {
                temp = temp.link;
            }
            temp.link = newnode;
        }
        tf1.setText("");
    }
});
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnNewButton.setBounds(391, 51, 121, 23);
contentPane.add(btnNewButton);

JButton btnInsertFront = new JButton("INSERT FRONT");
btnInsertFront.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Insert front
        int elem = Integer.valueOf(tf2.getText());
        String insert_elem = "Inserted Element is " +elem+ " at Front End";
        JOptionPane.showMessageDialog(contentPane, insert_elem);
        Node newnode = new Node();
        newnode.data = elem;
        newnode.link = null;
        if (first == null) {
            first = newnode;

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

        } else {
            newnode.link = first;
            first = newnode;
        }
        tf2.setText("");
    }

});
btnInsertFront.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnInsertFront.setBounds(391, 103, 121, 23);
contentPane.add(btnInsertFront);

JButton btnDeleteRear = new JButton("DELETE REAR");
btnDeleteRear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // delete rear
        Node temp;
        if (first == null) {
            JOptionPane.showMessageDialog(contentPane, "Deletion not
possible");
        }
        if (first.link == null) {
            String dr = "Element is deleted is " + first.data;
            JOptionPane.showMessageDialog(contentPane, dr);
            first = null;
        } else {
            temp = first;
            while (temp.link.link != null) {
                temp = temp.link;
            }
            String dr = "Element deleted is " + temp.link.data;
            JOptionPane.showMessageDialog(contentPane, dr);
            temp.link = null;
        }
    }
});
btnDeleteRear.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnDeleteRear.setBounds(218, 160, 123, 23);
contentPane.add(btnDeleteRear);

JButton btnDeleteFront = new JButton("DELETE FRONT");
btnDeleteFront.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // delete front
        if (first == null) {
            JOptionPane.showMessageDialog(contentPane, "Deletion not
possible");
        }
        if (first.link == null) {
            String df = "Element is deleted is " + first.data;

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

        JOptionPane.showMessageDialog(contentPane, df);
        first = null;
    } else {
        String df = "Element deleted is " + first.data;
        JOptionPane.showMessageDialog(contentPane, df);
        first = first.link;
    }
}

});
btnDeleteFront.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnDeleteFront.setBounds(218, 208, 123, 23);
contentPane.add(btnDeleteFront);

JButton btnDisplay = new JButton("DISPLAY");
btnDisplay.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // display
        Node temp;
        if (first == null) {
            JOptionPane.showMessageDialog(contentPane, "Display not
possible");

            displaybox.setText("");
        } else if (first.link == null) {
            String msg = "";
            msg = msg + " " + Integer.toString(first.data);
            displaybox.setText(msg);
        } else {
            temp = first;
            String msg = "";
            while (temp != null) {
                msg = msg + " " + Integer.toString(temp.data);
                displaybox.setText(msg);
                temp = temp.link;
            }
        }
    }
});
btnDisplay.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnDisplay.setBounds(218, 257, 123, 23);
contentPane.add(btnDisplay);
displaybox = new JTextField();
displaybox.setHorizontalAlignment(SwingConstants.CENTER);
displaybox.setFont(new Font("Times New Roman", Font.BOLD, 20));
displaybox.setBackground(new Color(255, 228, 181));
displaybox.setBounds(10, 290, 532, 31);
contentPane.add(displaybox);
displaybox.setColumns(10);
}
}

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

## DOUBLY LINKED LIST DATA STRUCTURE

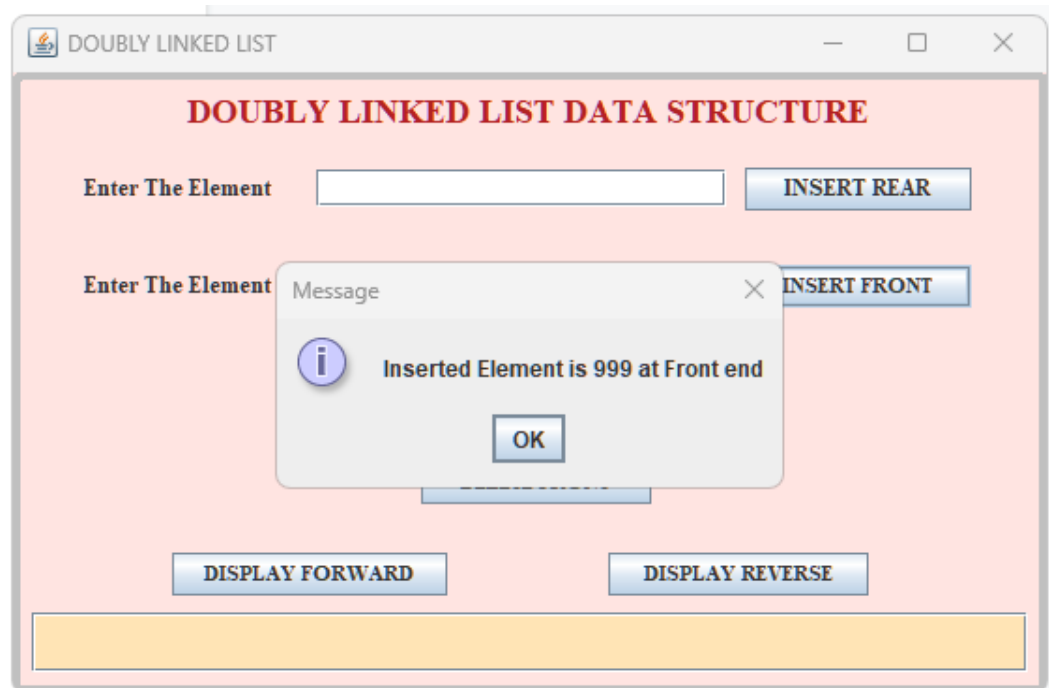
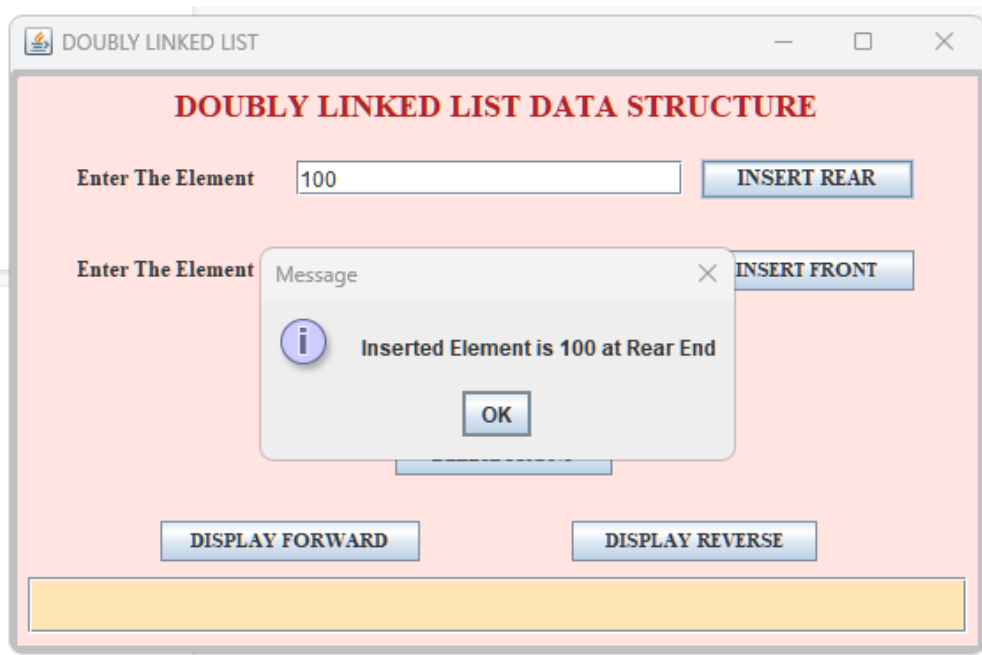
---

The screenshot shows a Java Swing window titled "DOUBLY LINKED LIST". The window has a light pink background and a title bar with standard Windows window controls. The main content area contains the following elements:

- DOUBLY LINKED LIST DATA STRUCTURE**: A title label in bold red text at the top center.
- Enter The Element**: A text label followed by a text input field.
- INSERT REAR**: A blue button with white text.
- Enter The Element**: A text label followed by a text input field.
- INSERT FRONT**: A blue button with white text.
- DELETE REAR**: A blue button with white text.
- DELETE FRONT**: A blue button with white text.
- DISPLAY FORWARD**: A blue button with white text.
- DISPLAY REVERSE**: A blue button with white text.
- A large yellow rectangular area at the bottom, likely a display for the list elements.

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan





Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

DOUBLY LINKED LIST

### DOUBLY LINKED LIST DATA STRUCTURE

Enter The Element

Enter The Element

99 999 100 200 300

DOUBLY LINKED LIST

### DOUBLY LINKED LIST DATA STRUCTURE

Enter The Element

Enter The Element

300 200 100 999 99

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

DOUBLY LINKED LIST

### DOUBLY LINKED LIST DATA STRUCTURE

Enter The Element

Enter The Element

i

Element deleted is 300

OK

300 200 100 999 99

DOUBLY LINKED LIST

### DOUBLY LINKED LIST DATA STRUCTURE

Enter The Element

Enter The Element

99 999 100 200

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan

```

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import java.awt.Color;
import javax.swing.JTextArea;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.SwingConstants;
import javax.swing.border.LineBorder;
public class DoublyLinkedList extends JFrame {

    private JPanel contentPane;
    private JTextField textField;
    private JTextField tf1;
    private JTextField tf2;
    private JTextField displaybox;
    class Node {
        int data;
        Node nextlink;
        Node prelink;
    }
    private Node first;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    DoublyLinkedList frame = new DoublyLinkedList();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public DoublyLinkedList() {
        setTitle("DOUBLY LINKED LIST");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // 568, 371
        setBounds(100, 100, 568, 371);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(255, 228, 225));
    }

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

contentPane.setBorder(new LineBorder(new Color(192, 192, 192), 4, true));

setContentPane(contentPane);
contentPane.setLayout(null);
JLabel lblNewLabel = new JLabel("DOUBLY LINKED LIST DATA STRUCTURE");
lblNewLabel.setForeground(new Color(178, 34, 34));
lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 18));
lblNewLabel.setBounds(93, 11, 364, 21);
contentPane.add(lblNewLabel);

tf1 = new JTextField();
tf1.setBounds(162, 52, 219, 20);
contentPane.add(tf1);
tf1.setColumns(10);

tf2 = new JTextField();
tf2.setColumns(10);
tf2.setBounds(162, 104, 219, 20);
contentPane.add(tf2);

JLabel lblNewLabel_1 = new JLabel("Enter The Element ");
lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_1.setBounds(38, 55, 103, 15);
contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_1_1 = new JLabel("Enter The Element ");
lblNewLabel_1_1.setFont(new Font("Times New Roman", Font.BOLD, 12));
lblNewLabel_1_1.setBounds(38, 107, 103, 15);
contentPane.add(lblNewLabel_1_1);

JButton btnNewButton = new JButton("INSERT REAR");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // insert rear
        Node temp;
        int elem = Integer.valueOf(tf1.getText());
        String insert_elem = "Inserted Element is " + elem + " at Rear End";
        JOptionPane.showMessageDialog(contentPane, insert_elem);
        Node newnode = new Node();
        newnode.data = elem;
        newnode.prelink = null;
        newnode.nextlink = null;
        if (first == null) {
            first = newnode;
        } else {
            temp = first;
            while (temp.nextlink != null) {
                temp = temp.nextlink;
            }
        }
    }
});

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

        temp.nextlink = newnode;
        newnode.prelink = temp;
    }
    tf1.setText("");
}

});
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnNewButton.setBounds(391, 51, 121, 23);
contentPane.add(btnNewButton);

JButton btnInsertFront = new JButton("INSERT FRONT");
btnInsertFront.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // insert front
        int elem = Integer.valueOf(tf2.getText());
        String insert_elem = "Inserted Element is " + elem + " at Front end";
        JOptionPane.showMessageDialog(contentPane, insert_elem);
        Node newnode = new Node();
        newnode.data = elem;
        newnode.prelink = null;
        newnode.nextlink = null;
        if (first == null) {
            first = newnode;
        } else {
            newnode.nextlink = first;
            first.prelink = newnode;
            first = newnode;
        }
        tf2.setText("");
    }
});
btnInsertFront.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnInsertFront.setBounds(391, 103, 121, 23);
contentPane.add(btnInsertFront);

JButton btnDeleteRear = new JButton("DELETE REAR");
btnDeleteRear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // delete rear
        Node temp;
        if (first == null) {
            JOptionPane.showMessageDialog(contentPane, "Deletion not
possible");
        } else if (first.nextlink == null) {
            String msg = "Element deleted is " + first.data;
            JOptionPane.showMessageDialog(contentPane, msg);
            first = null;
        } else {
            temp = first;

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

        while (temp.nextlink.nextlink != null) {
            temp = temp.nextlink;
        }
        String msg = "Element deleted is " + temp.nextlink.data;
        JOptionPane.showMessageDialog(contentPane, msg);
        temp.nextlink = null;
    }
}

});
btnDeleteRear.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnDeleteRear.setBounds(218, 160, 123, 23);
contentPane.add(btnDeleteRear);

JButton btnDeleteFront = new JButton("DELETE FRONT");
btnDeleteFront.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // delete front
        if (first == null) {
            JOptionPane.showMessageDialog(contentPane, "Deletion not
possible");

        } else if (first.nextlink == null) {
            String msg = "Element deleted is " + first.data;
            JOptionPane.showMessageDialog(contentPane, msg);
            first = null;
        } else {
            first = first.nextlink;
            first.prelink = null;
        }
    }
});
btnDeleteFront.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnDeleteFront.setBounds(218, 208, 123, 23);
contentPane.add(btnDeleteFront);

JButton btnDisplay = new JButton("DISPLAY FORWARD");
btnDisplay.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // display forward
        Node temp;
        if (first == null) {
            JOptionPane.showMessageDialog(contentPane, "Display not
possible");

            displaybox.setText("");
        } else if (first.nextlink == null) {
            String msg = "";
            msg = msg + " " + Integer.toString(first.data);
            displaybox.setText(msg);
        } else {
            temp = first;

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan

```

        String msg = "";
        while (temp != null) {
            msg = msg + " " + Integer.toString(temp.data);
            displaybox.setText(msg);
            temp = temp.nextlink;
        }
    }

});
btnDisplay.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnDisplay.setBounds(85, 257, 147, 23);
contentPane.add(btnDisplay);

textField = new JTextField();
textField.setBounds(166, 52, 215, 20);
contentPane.add(textField);
textField.setColumns(10);

JButton btnDisplayReverse = new JButton("DISPLAY REVERSE");
btnDisplayReverse.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // display reverse
        Node temp;
        if (first == null) {
            JOptionPane.showMessageDialog(contentPane, "Display not
possible");

            displaybox.setText("");
        }
        else if (first.nextlink == null) {
            String msg = "";
            msg = msg + " " + Integer.toString(first.data);
            displaybox.setText(msg);
        } else {
            temp = first;
            while (temp.nextlink != null) {
                temp = temp.nextlink;
            }
            String msg = "";
            while (temp != null) {
                msg = msg + " " + Integer.toString(temp.data);
                displaybox.setText(msg);
                temp = temp.prelink;
            }
        }
    }
});
btnDisplayReverse.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnDisplayReverse.setBounds(318, 257, 139, 23);
contentPane.add(btnDisplayReverse);

```

Name :-Harsh Chavan  
 Batch :-SEP 22 Batch  
 KODDCO044-Harsh Chavan



```
        displaybox = new JTextField();
        displaybox.setHorizontalAlignment(SwingConstants.CENTER);
        displaybox.setFont(new Font("Times New Roman", Font.BOLD, 20));
        displaybox.setBackground(new Color(255, 228, 181));
        displaybox.setBounds(10, 289, 532, 32);
        contentPane.add(displaybox);
        displaybox.setColumns(10);
    }
}
```

Name :-Harsh Chavan  
Batch :-SEP 22 Batch  
KODDCO044-Harsh Chavan