

PROGRAM

```
import java.util.Random;

class OddThread extends Thread{
    private int num;
    public OddThread(int num){
        this.num=num;
    }
    public void run(){
        System.out.println("cube of "+num+"="+num*num*num);
    }
}

class EvenThread extends Thread{
    private int num;
    public EvenThread(int num){
        this.num=num;
    }
    public void run(){
        System.out.println("square of "+num+"="+num*num);
    }
}

class RandomThread extends Thread{
    public void run(){
        Random r =new Random();
        for(int i=0;i<10;i++){
            int num=r.nextInt(100);
            if(num%2==0)
                new EvenThread(num).start();
            else
                new OddThread(num).start();
        }
    }
}

public class TestMultithread {
    public static void main(String[] args) {
        RandomThread rt =new RandomThread();
        rt.start();
    }
}
```

OUTPUT

cube of 3=27

cube of 19=6859

cube of 35=42875

square of 32=1024

square of 94=8836

cube of 83=571787

cube of 9=729

square of 82=6724

square of 58=3364

cube of 95=857375

PROGRAM

```
class Display{
    public synchronized void print(String msg){
        System.out.print("[ "+msg);
        try{
            Thread.sleep(2000);
        }
        catch(Exception e){
            System.out.println(e.getMessage());
        }
        System.out.println("]");
    }
}

class SyncThread extends Thread {
    private Display d;
    private String msg;
    public SyncThread(Display d, String msg){
        this.d=d;
        this.msg=msg;
    }
    public void run(){
        d.print(msg);
    }
}

public class TestThreadSynchro{
    public static void main(String[] args){
        Display d =new Display();
        SyncThread t1 =new SyncThread(d,"Hello");
        SyncThread t2 =new SyncThread(d,"world");
        t1.start();
        t2.start();
    }
}
```

OUTPUT

```
[Hello]
[world]
```

PROGRAM

```
import java.util.Scanner;
public class DLinkedList{
    class Node{
        int data;
        Node next;
        Node prev;
        Node(int data){
            this.data=data;
            this.prev=null;
            this.next=null;
        }
    }
    public Node head;

    public void addNode(int data){
        Node newNode = new Node(data);
        if(head==null)
            head=newNode;
        else{
            Node temp =head;
            while(temp.next!=null)
                temp=temp.next;
            temp.next=newNode;
            newNode.prev=temp;
        }
    }
    public void removeNode(int data){
        if(head==null){
            System.out.println("List empty");
            return;
        }
        if(head.data==data){
            if(head.next!=null)
                head.next.prev=null;
            head=head.next;
            return;
        }
        Node temp =head;
        while(temp!=null){
            if(temp.data==data)
                break;
            temp=temp.next;
        }
        if(temp==null){
            System.out.println("data not found");
            return;
        }
    }
}
```

```

    }
    if(temp.next!=null)
        temp.next.prev=temp.prev;
    temp.prev.next=temp.next;
}

public void display(){
    if(head==null){
        System.out.println("Empty List ");
        return;
    }
    Node temp=head;
    System.out.print("List :");
    while(temp!=null){
        System.out.print(temp.data+" ");
        temp=temp.next;
    }
}

public static void main(String args[]){
    DLinkedList list =new DLinkedList();
    while(true){
        System.out.println("\nEnter \n1. add Node\n2 remove Node\n3 exit");
        Scanner sc =new Scanner(System.in);
        char ch= sc.nextLine().charAt(0);
        switch(ch){
            case '1': System.out.print("enter the data :");
                list.addNode(sc.nextInt());
                list.display();
                break;
            case '2':System.out.print("enter the data to be delete:");
                list.removeNode(sc.nextInt());
                list.display();
                break;
            case '3': return;
        }
    }
}
}

```

OUTPUT

```
Enter                               enter the data to be delete:21
1. add Node                         List :3 2 4
2 remove Node                       Enter
3 exit                             1. add Node
1                                  2 remove Node
enter the data :3                  3 exit
List :3                             1
Enter                               enter the data :5
1. add Node                         List :3 2 4 5
2 remove Node                       Enter
3 exit                             1. add Node
1                                  2 remove Node
enter the data :2                  3 exit
List :3 2                           3
Enter
1. add Node
2 remove Node
3 exit
1
enter the data :8
List :3 2 8
Enter
1. add Node
2 remove Node
3 exit
1
enter the data :21
List :3 2 8 21
Enter
1. add Node
2 remove Node
3 exit
1
enter the data :4
List :3 2 8 21 4
Enter
1. add Node
2 remove Node
3 exit
2
enter the data to be delete:8
List :3 2 21 4
Enter
1. add Node
2 remove Node
3 exit
2
```

PROGRAM

```
import java.util.Scanner;
public class QuickSort {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of names: ");
        int n = sc.nextInt();
        String[] names = new String[n];
        System.out.println("Enter the names: ");
        for (int i = 0; i < n; i++)
            names[i] = sc.next();
        quickSort(names, 0, n - 1);
        System.out.println("Sorted names: ");
        for (String name : names)
            System.out.println(name);
    }
    public static void quickSort(String[] arr, int left, int right) {
        if (left >= right)
            return;
        int pivotIndex = partition(arr, left, right);
        quickSort(arr, left, pivotIndex - 1);
        quickSort(arr, pivotIndex + 1, right);
    }
    public static int partition(String[] arr, int left, int right){
        String pivot = arr[left];
        int i = left+1;
        for (int j = left + 1; j <= right; j++)
            if (arr[j].compareTo(pivot) < 0) {
                swap(arr, i, j);
                i++;
            }
        swap(arr, left, i-1);
        return i-1;
    }
    public static void swap(String[] arr, int i, int j) {
        String temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
}
```

OUTPUT

Enter the number of names: 5

Enter the names:

raju

appu

manju

babu

ammu

Sorted names:

ammu

appu

babu

manju

raju

PROGRAM

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

public class Calculator implements ActionListener{
    JFrame frame;
    JTextField textField;
    JButton[] numberButtons= new JButton[10];
    JButton[] functionButton=new JButton[8];
    JButton addButton,subButton,mulButton,divButton;
    JButton decButton,equButton,delButton,clrButton;
    JPanel panel;
    double num1=0,num2=0,result=0;
    char operator;

    Calculator(){
        Font myFont = new Font("Ink Free",Font.BOLD,35);
        frame =new JFrame("CALCULATOR");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setBounds(500,150,420,550);
        frame.setLayout(null);
        frame.getContentPane().setBackground(new Color(147, 161, 96));
        textField = new JTextField();
        textField.setBounds(50,25,300,50);
        textField.setFont(myFont);
        textField.setEditable(false);

        addButton =new JButton("+");
        subButton =new JButton("-");
        mulButton =new JButton("*");
        divButton =new JButton("/");
        decButton =new JButton(".");
        equButton =new JButton("=");
        delButton =new JButton("delete");
        clrButton =new JButton("clear");

        functionButton[0] =addButton;
        functionButton[1] =subButton;
        functionButton[2] =mulButton;
        functionButton[3] =divButton;
        functionButton[4] =decButton;
        functionButton[5] =equButton;
        functionButton[6] =delButton;
        functionButton[7] =clrButton;
        for(int i=0;i<8;i++){
            functionButton[i].addActionListener(this);
```

```

        functionButton[i].setFont(myFont);
        functionButton[i].setFocusable(false);
    }
    for(int i=0;i<10;i++){
        numberButtons[i] =new JButton(""+i);
        numberButtons[i].addActionListener(this);
        numberButtons[i].setFont(myFont);
        numberButtons[i].setFocusable(false);
    }
    delButton.setBounds(50,430,145,50);
    clrButton.setBounds(205,430,145,50);
    panel =new JPanel();
    panel.setBounds(50,100,300,300);
    panel.setLayout(new GridLayout(4,4,10,10));
    panel.add(numberButtons[7]);
    panel.add(numberButtons[8]);
    panel.add(numberButtons[9]);
    panel.add(addButton);

    panel.add(numberButtons[4]);
    panel.add(numberButtons[5]);
    panel.add(numberButtons[6]);
    panel.add(subButton);

    panel.add(numberButtons[1]);
    panel.add(numberButtons[2]);
    panel.add(numberButtons[3]);
    panel.add(mulButton);

    panel.add(decButton);
    panel.add(numberButtons[0]);
    panel.add(equButton);
    panel.add(divButton);

    frame.add(panel);
    frame.add(delButton);
    frame.add(clrButton);
    frame.add(textField);
    frame.setResizable(false);
    frame.setVisible(true);
}

public static void main(String argd[]){
    Calculator calc =new Calculator();
}
public void actionPerformed(ActionEvent e){
    for(int i=0;i<10;i++)
        if(e.getSource()==numberButtons[i])

```

```

        textField.setText(textField.getText() + i);

if(e.getSource()==clrButton)
    textField.setText("");

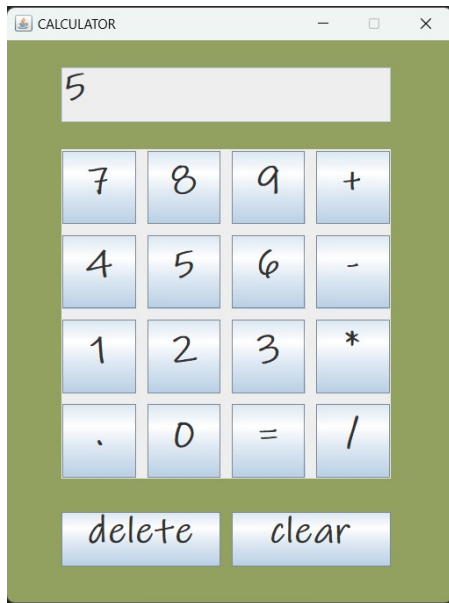
if(e.getSource()==decButton)
    textField.setText(textField.getText() + ".");

if(e.getSource()==addButton){
    num1= Double.parseDouble(textField.getText());
    operator='+';
    textField.setText("");
}
if(e.getSource()==subButton){
    num1= Double.parseDouble(textField.getText());
    operator='-';
    textField.setText("");
}
if(e.getSource()==mulButton){
    num1= Double.parseDouble(textField.getText());
    operator='*';
    textField.setText("");
}
if(e.getSource()==divButton){
    num1= Double.parseDouble(textField.getText());
    operator='/';
    textField.setText("");
}
if(e.getSource()==equButton){
    num2=Double.parseDouble(textField.getText());
    switch(operator){
        case '+':result=num1+num2;break;
        case '-':result=num1-num2;break;
        case '*':result=num1*num2;break;
        case '/':result=num1/num2;break;
    }
    textField.setText(""+result);
    num1=result;
}
if(e.getSource()==delButton){
    String string =textField.getText();
    textField.setText("");
    for(int i=0;i<string.length()-1;i++)
        textField.setText(textField.getText()+string.charAt(i));
}

}
}

```

OUTPUT



$$5 + 5 = 10.0$$

$$25 / 2 = 12.5$$

$$50 * 4 = 200.0$$

$$12 - 3 = 9.0$$

$$12.1 - .1 = 12.0$$