

GURU TEGH BAHADUR INSTITUTE OF TECHNOLOGY



SUBJECT CODE: CIC-258

JAVA PRACTICAL FILE

SEMESTER – IV

Submitted to,

Mr. Gaurav Sandhu

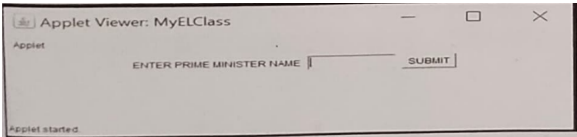
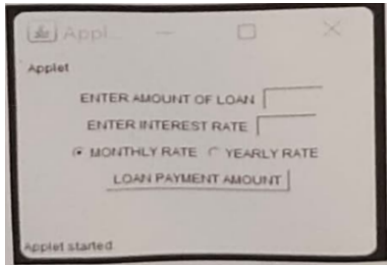
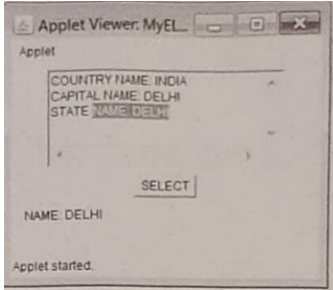
Submitted by,

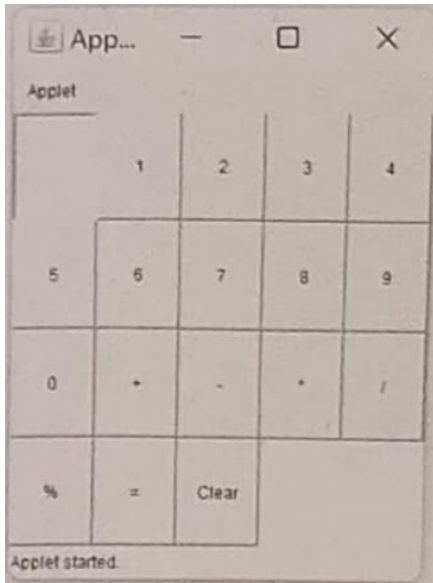
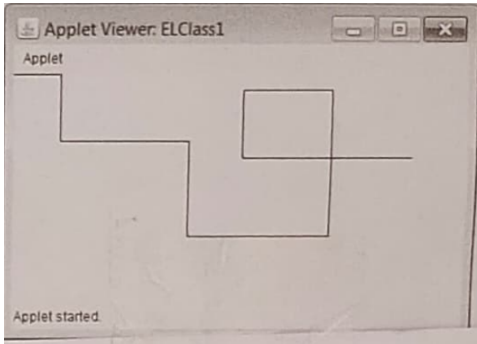
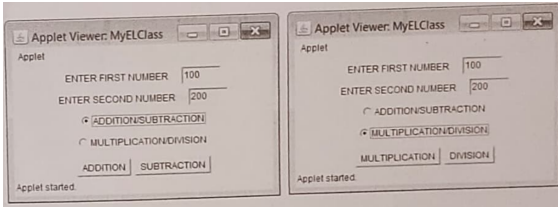
Abhay Raj (00976803122)

Index

S.No	Program	Page	Date	Signature
1.	Design a java program to find the biggest of three given integers.			
2.	Design a java program to define a class, describe its constructor, over-load the constructor and instantiate the object.			
3.	Design a java program to demonstrate the use of nested class.			
4.	Design a java program to print all the real solutions of the quadratic equation $ax^2+bx+c=0$, read in the values of a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.			
5.	Design a java program that uses both recursive and non-recursive methods to print the nth value of the Fibonacci series sequence.			
6.	Design a java program to prompt user for an integer and print all the prime numbers up to that integer.			
7.	Design a java program to check whether a string is palindrome or not.			
8.	Design a java program that reads a line of integers, then display each integer and the sum of all integers.			
9.	Design a java program to implement stack and queue concept.			
10.	Design a java program to create an abstract class named Shape, that contains an empty method called numberOfSides(). Provide four classes named Trapezoid, Triangle, Rectangle and Hexagon such that each one of the classes contains only the method numberOfSides(), that contains number of sides in each geometrical figure.			

11.	Design a java program to show dynamic polymorphism.			
12.	Design a java program to show interfaces.			
13.	Design a java program to create a customized exception and also make use of all 5 exception keywords.			
14.	Design a java program to sort list of names in ascending order.			
15.	Design a java program to create two threads: one for odd numbers and other for even numbers.			
16.	Design a java program to implement producer consumer problem using concept of inter thread communication.			
17.	Design a java program to create three threads runnable interface. First thread display "GOOD MORNING" at every 1 second. Second thread displays "HELLO" at every 2 seconds and third thread displays "WELCOME" at every 3 seconds.			
18.	Design a java program to create three threads. First thread display "GOOD MORNING" at every 1 second. Second thread displays "HELLO" at every 2 seconds and third thread displays "WELCOME" at every 3 seconds.			
19.	<p>Design an applet to do the following tasks:</p> <p>a) To output the question "WHO IS THE PRIME MINISTER OF INDIA?"</p> <p>b) To accept the answer and print "CORRECT" and then stop if the answer is correct.</p> <p>c) To print "TRY AGAIN" and if the answer is not correct.</p> <p>d) To display the correct answer, if the answer is wrong even after third attempt.</p>			

				
20.	<p>Design a java program to write an applet that computes the payment of loan which is based on the amount of loan, interest rate and number of months. It takes one parameter from browser which is called Monthly Rate. If it is True then interest rate is per month else it is annual. (Use label, textfield, button, check box, checkbox group).</p> 			
21.	<p>Design a java program to write an applet with following AWT components :textarea and button.</p> 			
22.	<p>Design a java program to write a calculator applet. Use grid layout to arrange the buttons for digits and for the +, -, *, /, % operations. Add textfield to display the result.</p>			

				
23.	<p>Design a java program to handle keyboard events.</p> 			
24.	<p>Design a program to Implement card layout in applets.</p> 			
25.	<p>Program In Java To Demonstrate FileWriter, FileReader, BufferedReader classes to Enter Multiple Strings And Numbers Into A File, Read File Using Loop And Display.</p>			

26.	Program to implement various functions giving details about a file.			
27.	Design a program in java to implement client-server TCP/IP socket application which exchange strings among them.			
28.	Program In Java To Make Database Connectivity Using JDBC, create statement object, create resultset object And Display (All Or Specific Records using first(), last(), previous(), next() and absolute() functions). (Pre requirements: install MySql software, create database: Shiksha, create table house with attributes as H_No, H_Locality, H_Rooms, H_Owner, insert rows of data into table house)			
29.	Program In Java To Make Database Connectivity Using JDBC, create statement object, create resultset object, insert a record using prepared statement and Display all records. (Pre requirements: install MySql software, create database: Shiksha, create table house with attributes as H_No, H_Locality, H_Rooms, H_Owner, insert rows of data into table house)			
30.	Program In Java To Make Database Connectivity Using JDBC, create statement object, create resultset object, update a record based on H_No and Display all records. (Pre requirements: install MySql software, create database: Shiksha, create table house with attributes as H_No, H_Locality, H_Rooms, H_Owner, insert rows of data into table house)			
31.	Program In Java To Make Database Connectivity Using JDBC, create statement object, create resultset object, delete a record based on H_No and Display all records. (Pre requirements: install MySql software, create database: Shiksha, create table house with attributes as H_No, H_Locality, H_Rooms, H_Owner, insert rows of data into table house)			

Program – 1

Aim: Design a java program to find the biggest of three given integers.

Code:

```
import java.util.*;

class BiggestNumberFinder {
    public static int findBiggest(int num1, int num2, int num3) {
        int max = num1;

        if (num2 > max) {
            max = num2;
        }

        if (num3 > max) {
            max = num3;
        }

        return max;
    }
}

public class MyELClass {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

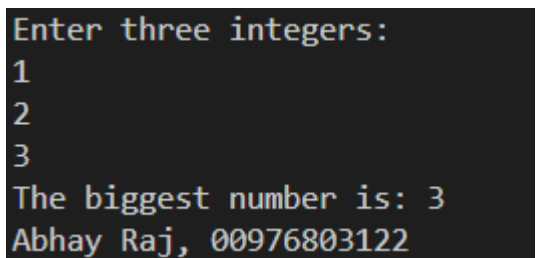
        System.out.println("Enter three integers:");
        int num1 = in.nextInt();
        int num2 = in.nextInt();
        int num3 = in.nextInt();

        int max = BiggestNumberFinder.findBiggest(num1, num2, num3);

        System.out.println("The biggest number is: " + max);
        System.out.println("Abhay Raj, 00976803122" );

        in.close();
    }
}
```

Output:

A screenshot of a terminal window showing the output of the Java program. The text is as follows:

```
Enter three integers:
1
2
3
The biggest number is: 3
Abhay Raj, 00976803122
```

Program – 2

Aim: Design a java program to define a class, describe its constructor, over-load the constructor and instantiate the object.

Code:

```
class Shapes
{
    double len;
    double bre;
    double result;

    Shapes()
    {
        len=0.0;
        bre=0.0;
    }

    Shapes(double l)
    {
        len=l;
        bre=0.0;
    }

    Shapes(double l, double b)
    {
        len=l;
        bre=b;
    }

    void area()
    {
        if(len!=0.0 && bre==0.0)
        {
            result=len*len;
            System.out.println("AREA SQUARE " + result);
        }
        if(len!=0.0 && bre!=0.0)
        {
            result=len*bre;
            System.out.println("AREA RECTANGLE " + result);
        }
        else if(len==0.0 && bre==0.0)
        {
            System.out.println("NO OUTPUT");
        }
    }
}
```



```
class MyELClass
{
    public static void main(String a[])
    {
        double para1, para2;
        para1=10.0;
        para2=20.0;
        Shapes s1=new Shapes();
        Shapes s2=new Shapes(para1);
        Shapes s3=new Shapes(para1,para2);
        s2.area();
        s3.area();
        s1.area();
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:

```
AREA SQUARE 100.0
AREA RECTANGLE 200.0
NO OUTPUT
Abhay Raj, 00976803122
```

Program – 3

Aim: Design a java program to demonstrate the use of nested class.

Code:

```
class Outer
{
    private int outer_x = 100;
    void test()
    {
        Inner inner = new Inner();
        inner.display();
        inner.hello();
    }

    class Inner
    {
        private void hello()
        {
            System.out.println("Private Method Hello() of Inner Class");
        }

        void display()
        {
            System.out.println("Display: outer_x = " + outer_x);
            Outer o=new Outer();
            System.out.println("Display: outer_x = " + o.outer_x);
        }
    }
}

class MyELClass
{
    public static void main(String args[])
    {
        Outer outer = new Outer();
        outer.test();
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:

```
Display: outer_x = 100
Display: outer_x = 100
Private Method Hello() of Inner Class
Abhay Raj, 00976803122
```

Program – 4

Aim: Design a java program to print all the real solutions of the quadratic equation $ax^2+bx+c=0$, read in the values of a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

Code:

```
import java.util.*;

class QuadraticEquationSolver {
    public static void solveQuadraticEquation(double a, double b, double c) {
        double discriminant = b * b - 4 * a * c;

        if (discriminant < 0) {
            System.out.println("No real solutions exist.");
        } else {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);

            System.out.println("The real solutions of the quadratic equation are:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        }
    }
}

public class MyELClass {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the coefficients of the quadratic equation  $ax^2 + bx + c = 0$ :");
        System.out.print("a: ");
        double a = scanner.nextDouble();
        System.out.print("b: ");
        double b = scanner.nextDouble();
        System.out.print("c: ");
        double c = scanner.nextDouble();

        QuadraticEquationSolver.solveQuadraticEquation(a, b, c);

        scanner.close();
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:

```
Enter the coefficients of the quadratic equation  $ax^2 + bx + c = 0$ :  
a: 1  
b: 2  
c: 1  
The real solutions of the quadratic equation are:  
Root 1: -1.0  
Root 2: -1.0  
Abhay Raj, 00976803122
```

Program – 5

Aim: Design a java program that uses both recursive and non-recursive methods to print the nth value of the Fibonacci series sequence.

Code:

```
import java.util.*;

class FibonacciCalculator {
    // Non-recursive method to find nth Fibonacci number
    public static long fibonacciNonRecursive(int n) {
        long a = 0;
        long b = 1;
        long fib = 0;

        if (n == 0) return a;
        if (n == 1) return b;

        for (int i = 2; i <= n; i++) {
            fib = a + b;
            a = b;
            b = fib;
        }

        return fib;
    }

    // Recursive method to find nth Fibonacci number
    public static long fibonacciRecursive(int n) {
        if (n <= 1) return n;
        return fibonacciRecursive(n - 1) + fibonacciRecursive(n - 2);
    }
}

public class MyELClass {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the value of n to find nth Fibonacci number: ");
        int n = scanner.nextInt();

        // Using non-recursive method
        System.out.println("Using non-recursive method:");
        long resultNonRecursive = FibonacciCalculator.fibonacciNonRecursive(n);
        System.out.println("The " + n + "th Fibonacci number is: " + resultNonRecursive);

        // Using recursive method
        System.out.println("\nUsing recursive method:");
        long resultRecursive = FibonacciCalculator.fibonacciRecursive(n);
        System.out.println("The " + n + "th Fibonacci number is: " + resultRecursive);

        scanner.close();
    }
}
```

```
        System.out.println("Abhay Raj, 00976803122" );  
    }  
}
```

Output:

```
Enter the value of n to find nth Fibonacci number: 10  
Using non-recursive method:  
The 10th Fibonacci number is: 55  
  
Using recursive method:  
The 10th Fibonacci number is: 55  
Abhay Raj, 00976803122
```

Program – 6

Aim: Design a java program to prompt user for an integer and print all the prime numbers up to that integer.

Code:

```
import java.util.*;

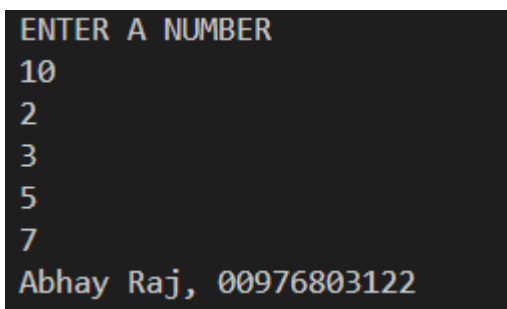
class Prime {
    int value, num;
    int i, flag;

    void inputNumber() {
        Scanner sc = new Scanner(System.in);
        System.out.println("ENTER A NUMBER ");
        value = sc.nextInt();
    }

    void displayPrime() {
        for (num = 2; num <= value; num++) {
            flag = 0;
            for (i = 2; i <= num - 1; i++) {
                if (num % i == 0) {
                    flag = 1;
                    break;
                }
            }
            if (flag == 0) {
                System.out.println(num);
            }
        }
    }
}

class MyELClass {
    public static void main(String a[]) {
        Prime o = new Prime();
        o.inputNumber();
        o.displayPrime();
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:



```
ENTER A NUMBER
10
2
3
5
7
Abhay Raj, 00976803122
```

Program – 7

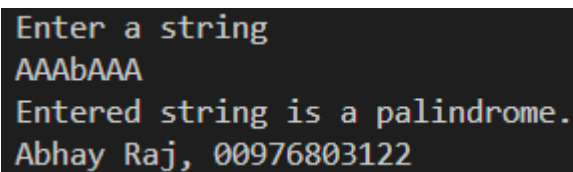
Aim: Design a java program to check whether a string is palindrome or not.

Code:

```
import java.util.*;

class P_Check {
    void check(String data) {
        String rdata = "";
        int i, len;
        len = data.length();
        for (i = len - 1; i >= 0; i--)
            rdata = rdata + data.charAt(i);
        if (data.equals(rdata))
            System.out.println("Entered string is a palindrome.");
        else
            System.out.println("Entered string is not a palindrome.");
    }
}

class MyELClass {
    public static void main(String a[]) {
        String name;
        P_Check pc = new P_Check();
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a string");
        name = in.nextLine();
        pc.check(name);
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

A screenshot of a terminal window showing the execution of the Java program. The output consists of four lines: 'Enter a string', 'AAAbAAA', 'Entered string is a palindrome.', and 'Abhay Raj, 00976803122'. The text is displayed in a monospaced font with a light blue/cyan color on a dark background.

```
Enter a string
AAAbAAA
Entered string is a palindrome.
Abhay Raj, 00976803122
```


Program – 8

Aim: Design a java program that reads a line of integers, then display each integer and the sum of all integers.

Code:

```
import java.util.*;

class IntData {
    int arr[];
    int i, sum;

    IntData(int size) {
        arr = new int[size];
        sum = 0;
    }

    void input() {
        Scanner sc = new Scanner(System.in);
        for (i = 0; i < arr.length; i++) {
            arr[i] = sc.nextInt();
        }
    }

    void display() {
        for (i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
    }

    void sumCal() {
        for (i = 0; i < arr.length; i++) {
            sum = sum + arr[i];
        }
        System.out.println("SUM IS " + sum);
    }
}

class MyELClass {
    public static void main(String a[]) {
        IntData obj = new IntData(5);
        obj.input();
        obj.display();
        obj.sumCal();
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:

```
1 2 3 4 5
1
2
3
4
5
SUM IS 15
Abhay Raj, 00976803122
```

Program – 9

Aim: Design a java program to implement stack and queue concept.

Code:

```
class Stack {
    int stck[];
    int tos;

    Stack(int size) {
        stck = new int[size];
        tos = -1;
    }

    void push(int item) {
        if (tos == stck.length - 1)
            System.out.println("Stack is full.");
        else
            stck[++tos] = item;
    }

    int pop() {
        if (tos < 0) {
            System.out.println("Stack underflow.");
            return 0;
        } else
            return stck[tos--];
    }
}

class MyELClass {
    public static void main(String args[]) {
        Stack mystack1 = new Stack(5);
        Stack mystack2 = new Stack(8);

        for (int i = 0; i < 5; i++)
            mystack1.push(i);

        for (int i = 0; i < 8; i++)
            mystack2.push(i);

        System.out.println("Stack in mystack1:");
        for (int i = 0; i < 5; i++)
            System.out.println(mystack1.pop());

        System.out.println("Stack in mystack2:");
        for (int i = 0; i < 8; i++)
            System.out.println(mystack2.pop());
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:

```
Stack in mystack1:
4
3
2
1
0
Stack in mystack2:
7
6
5
4
3
2
1
0
Abhay Raj, 00976803122
```

Program – 10

Aim: Design a java program to create an abstract class named Shape, that contains an empty method called numberOfSides(). Provide four classes named Trapezoid, Triangle, Rectangle and Hexagon such that each one of the classes contains only the method numberOfSides(), that contains number of sides in each geometrical figure.

Code:

```
abstract class Shape {
    abstract void numberOfSides();
}

class Trapezoid extends Shape {
    void numberOfSides() {
        System.out.println("A trapezoid has 4 sides.");
    }
}

class Triangle extends Shape {
    void numberOfSides() {
        System.out.println("A triangle has 3 sides.");
    }
}

class Rectangle extends Shape {
    void numberOfSides() {
        System.out.println("A rectangle has 4 sides.");
    }
}

class Hexagon extends Shape {
    void numberOfSides() {
        System.out.println("A hexagon has 6 sides.");
    }
}

public class MyELClass {
    public static void main(String a[]) {
        Shape trapezoid = new Trapezoid();
        trapezoid.numberOfSides();

        Shape triangle = new Triangle();
        triangle.numberOfSides();

        Shape rectangle = new Rectangle();
        rectangle.numberOfSides();

        Shape hexagon = new Hexagon();
        hexagon.numberOfSides();
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

}

Output:

```
A trapezoid has 4 sides.  
A triangle has 3 sides.  
A rectangle has 4 sides.  
A hexagon has 6 sides.  
Abhay Raj, 00976803122
```

Program – 11

Aim: Design a java program to show dynamic polymorphism.

Code:

```
class MyArea {
    void area() {
        System.out.println("A concrete function called Area()");
    }
}

class SQArea extends MyArea {
    double s;

    void area() { // Same name function in base class
        double areasq;
        s = 10.0;
        areasq = s * s;
        System.out.println("Area of Square " + areasq);
    }
}

class RECTArea extends MyArea {
    double l, b;

    void area() { // Same name function in base class
        double arearect;
        l = 10.0;
        b = 2.0;
        arearect = l * b;
        System.out.println("Area of Rectangle " + arearect);
    }
}

class MyELClass {
    public static void main(String a[]) {
        MyArea ma; // Base class object
        ma = new SQArea(); // Referring to SQArea
        ma.area(); // Calling SQArea method (Dynamic Polymorphism)
        ma = new RECTArea(); // Referring to RECTArea
        ma.area(); // Calling RECTArea method (Dynamic Polymorphism)
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:

```
Area of Square 100.0
Area of Rectangle 20.0
Abhay Raj, 00976803122
```

Program – 12

Aim: Design a java program to show interfaces.

Code:

```
interface IntStack {
    void push(int item); // Store an item.
    int pop(); // Retrieve an item.
}

class DynStack implements IntStack {
    private int stk[];
    private int tos;

    // Allocate and initialize stack.
    DynStack(int size) {
        stk = new int[size];
        tos = -1;
    }

    // Push an item onto the stack.
    public void push(int item) {
        // If stack is full, allocate a larger stack.
        if (tos == stk.length - 1) {
            int temp[] = new int[stk.length * 2]; // Making a double size array.
            for (int i = 0; i < stk.length; i++)
                temp[i] = stk[i];
            stk = temp;
            stk[++tos] = item;
        } else
            stk[++tos] = item;
    }

    // Pop an item from the stack.
    public int pop() {
        if (tos < 0) {
            System.out.println("Stack underflow.");
            return 0;
        } else
            return stk[tos--];
    }
}

class ELClass1 {
    public static void main(String args[]) {
        DynStack mystack1 = new DynStack(5);
        DynStack mystack2 = new DynStack(8);

        // These loops cause each stack to grow.
        for (int i = 0; i < 12; i++)
            mystack1.push(i);
    }
}
```



```

        for (int i = 0; i < 20; i++)
            mystack2.push(i);

        System.out.println("Stack in mystack1:");
        for (int i = 0; i < 12; i++)
            System.out.println(mystack1.pop());

        System.out.println("Stack in mystack2:");
        for (int i = 0; i < 20; i++)
            System.out.println(mystack2.pop());
        System.out.println("Abhay Raj, 00976803122" );
    }
}

```

Output:

```

Stack in mystack1:
11
10
9
8
7
6
5
4
3
2
1
0
Stack in mystack2:
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
Abhay Raj, 00976803122

```

Program – 13

Aim: Design a java program to create a customized exception and also make use of all 5 exception keywords.

Code:

```
class NSalException extends Exception {
    public NSalException(String s) {
        super(s); // CALLING CONSTRUCTOR Exception()
    }
}

class PSalException extends RuntimeException {
    public PSalException(String s) {
        super(s); // CALLING CONSTRUCTOR RuntimeException()
    }
}

class Employee {
    public void decideSal(String s1) throws NSalException, PSalException,
    NumberFormatException {
        int sal = Integer.parseInt(s1);
        if (sal <= 0) {
            NSalException no = new NSalException("Invalid Salary");
            throw(no);
        } else {
            PSalException po = new PSalException("Valid Salary");
            throw(po);
        }
    }
}

class MyELClass {
    public static void main(String args[]) {
        try {
            Employee e = new Employee();
            e.decideSal(args[0]);
        } catch (NumberFormatException nfe) {
            System.err.println("Please enter integer values.");
        } catch (NSalException no) {
            System.err.println("Negative Salary.");
        } catch (PSalException po) {
            System.err.println("Valid Positive Salary.");
        } finally {
            System.out.println("Finally Block executing");
        }
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:

```
Valid Positive Salary.  
Finally Block executing  
Abhay Raj, 00976803122
```

Program – 14

Aim: Design a java program to sort list of names in ascending order.

Code:

```
import java.util.*;

class SortNames {
    String name[] = new String[5];
    int i, n = 5;

    SortNames() {
        for (i = 0; i < n; i++) {
            name[i] = null;
        }
    }

    void inputNames() {
        name[0] = "Yogesh";
        name[1] = "Deepak";
        name[2] = "Aman";
        name[3] = "Srishti";
        name[4] = "Baljinder";
    }

    void bubbleSort() {
        int j;
        String temp;
        for (i = 0; i < n; i++) {
            for (j = 0; j < n - 1 - i; j++) {
                if (name[j].compareTo(name[j + 1]) > 0) {
                    temp = name[j];
                    name[j] = name[j + 1];
                    name[j + 1] = temp;
                }
            }
        }
    }

    void displayNames() {
        for (i = 0; i < n; i++) {
            System.out.println(name[i]);
        }
    }
}

class MyELClass {
    public static void main(String a[]) {
        SortNames sn = new SortNames();
        sn.inputNames();
    }
}
```

```
        sn.bubbleSort();  
        sn.displayNames();  
        System.out.println("Abhay Raj, 00976803122" );  
    }  
}
```

Output:

```
Aman  
Baljinder  
Deepak  
Srishti  
Yogesh  
Abhay Raj, 00976803122
```

Program – 15

Aim: Design a java program to create two threads: one for odd numbers and other for even numbers.

Code:

```
class EvenThread extends Thread {
    int num;

    void set() {
        this.num = 0;
    }

    public void run() {
        for (int i = 1; i <= 10; i++) {
            try {
                even();
                Thread.sleep(500);
            } catch (InterruptedException e) {
                System.out.println("INTERRUPTED EXCEPTION ");
            }
        }
    }

    void even() {
        num = num + 2;
        System.out.println("Even : " + num);
    }
}

class OddThread implements Runnable {
    int num;

    void set() {
        this.num = 1;
    }

    public void run() {
        for (int i = 1; i <= 10; i++) {
            try {
                odd();
                Thread.sleep(500);
            } catch (InterruptedException e) {
                System.out.println("INTERRUPTED EXCEPTION ");
            }
        }
    }

    void odd() {
        num = num + 2;
        System.out.println("Odd : " + num);
    }
}
```

```
    }  
}  
  
class JavaLab {  
    public static void main(String a[]) {  
        int i;  
        EvenThread et = new EvenThread();  
        OddThread ot = new OddThread();  
        Thread t = new Thread(ot);  
        et.set();  
        ot.set();  
        et.start();  
        t.start();  
        System.out.println("Abhay Raj, 00976803122" );  
    }  
}
```

Output:

```
Abhay Raj, 00976803122  
Even : 2  
Odd : 3  
Even : 4  
Odd : 5  
Odd : 7  
Even : 6  
Even : 8  
Odd : 9  
Even : 10  
Odd : 11  
Odd : 13  
Even : 12  
Odd : 15  
Even : 14  
Even : 16  
Odd : 17  
Odd : 19  
Even : 18  
Odd : 21  
Even : 20
```

Program – 16

Aim: Design a java program to implement producer consumer problem using concept of inter thread communication.

Code:

```
class Q {
    int n;
    boolean valueSet = false;

    synchronized int get() {
        while (valueSet == false)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        System.out.println("Got by Consumer: " + n);
        valueSet = false;
        notify();
        return n;
    }

    synchronized void put(int n) {
        while (valueSet == true)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        this.n = n;
        valueSet = true;
        System.out.println("Put by Producer: " + n);
        notify();
    }
}

class Producer implements Runnable {
    Q q;

    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }

    public void run() {
        int i = 0;
        while (true) {
            q.put(i++);
        }
    }
}
```



```

    }

    class Consumer implements Runnable {
        Q q;

        Consumer(Q q) {
            this.q = q;
            new Thread(this, "Consumer").start();
        }

        public void run() {
            while (true) {
                q.get();
            }
        }
    }

    class MyELClass {
        public static void main(String args[]) {
            Q q = new Q();
            new Producer(q);
            new Consumer(q);
            System.out.println("Press Control-C to stop.");
            System.out.println("Abhay Raj, 00976803122" );
        }
    }
}

```

Output:

```

Press Control-C to stop.
Abhay Raj, 00976803122
Put by Producer: 0
Got by Consumer: 0
Put by Producer: 1
Got by Consumer: 1
Put by Producer: 2
Got by Consumer: 2
Put by Producer: 3
Got by Consumer: 3
Put by Producer: 4
Got by Consumer: 4
Put by Producer: 5
Got by Consumer: 5
Put by Producer: 6
Got by Consumer: 6
Put by Producer: 7
Got by Consumer: 7
Put by Producer: 8
Got by Consumer: 8
Put by Producer: 9
Got by Consumer: 9
Put by Producer: 10
Got by Consumer: 10
Put by Producer: 11

```

Program – 17

Aim: Design a java program to create three threads runnable interface. First thread display "GOOD MORNING" at every 1 second. Second thread displays "HELLO" at every 2 seconds and third thread displays "WELCOME" at every 3 seconds.

Code:

```
class Callme {
    void call(String msg, int t) {
        System.out.println(msg);
        try {
            Thread.sleep(t * 1000);
        } catch (InterruptedException e) {
            System.out.println("Interrupted");
        }
    }
}

class Caller implements Runnable {
    String msg;
    int time;
    Callme target;
    Thread t;

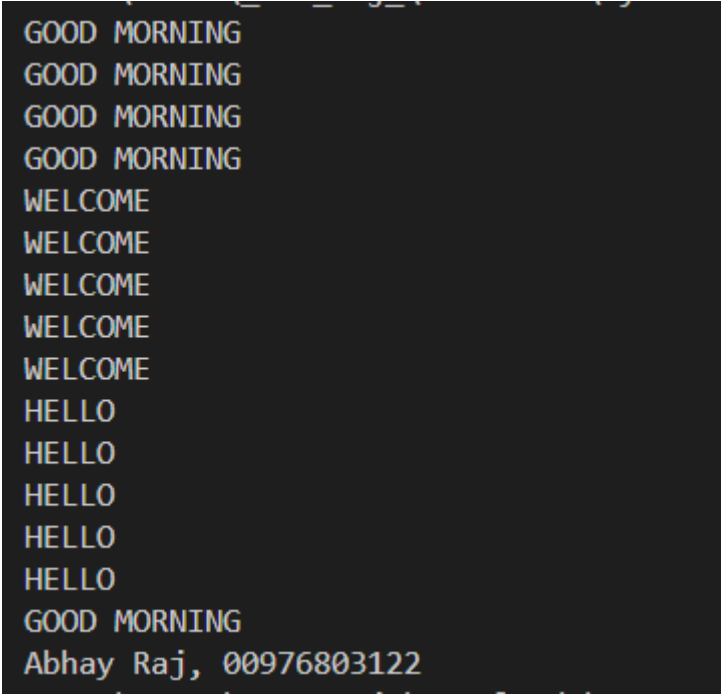
    public Caller(Callme targ, String s, int et) {
        target = targ;
        msg = s;
        time = et;
        t = new Thread(this);
        t.start();
    }

    public void run() {
        int i;
        for (i = 1; i <= 5; i++) {
            synchronized (target) {
                target.call(msg, time);
            }
        }
    }
}

class MyELClass {
    public static void main(String args[]) {
        Callme target = new Callme();
        Caller ob1 = new Caller(target, "GOOD MORNING", 1);
        Caller ob2 = new Caller(target, "HELLO", 2);
        Caller ob3 = new Caller(target, "WELCOME", 3);
        try {
            ob1.t.join();
            ob2.t.join();
        }
    }
}
```

```
        ob3.t.join();
    } catch (InterruptedException e) {
        System.out.println("Interrupted");
    }
    System.out.println("Abhay Raj, 00976803122" );
}
}
```

Output:



```
GOOD MORNING
GOOD MORNING
GOOD MORNING
GOOD MORNING
WELCOME
WELCOME
WELCOME
WELCOME
WELCOME
HELLO
HELLO
HELLO
HELLO
HELLO
GOOD MORNING
Abhay Raj, 00976803122
```

Program – 18

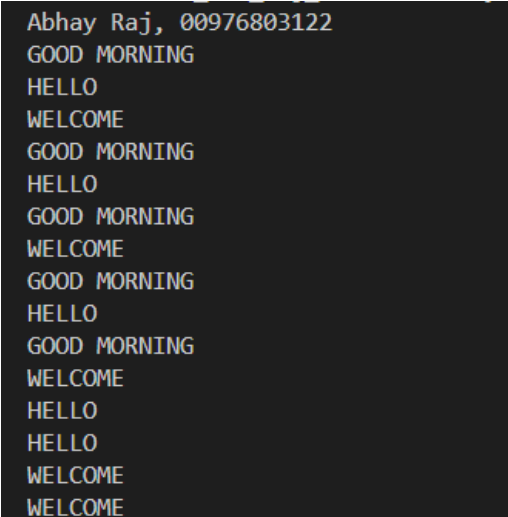
Aim: Design a java program to create three threads. First thread display "GOOD MORNING" at every 1 second. Second thread displays "HELLO" at every 2 seconds and third thread displays "WELCOME" at every 3 seconds.

Code:

```
class DisplayMessage extends Thread {
    private String message;
    private int interval;
    private int count;
    public DisplayMessage(String message, int interval) {
        this.message = message;
        this.interval = interval;
        this.count = 0;
    }
    public void run() {
        while (count < 5) { // Run for five times
            System.out.println(message);
            try {
                Thread.sleep(interval * 1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            count++;
        }
    }
}

public class MyELClass {
    public static void main(String[] args) {
        DisplayMessage thread1 = new DisplayMessage("GOOD MORNING", 1);
        DisplayMessage thread2 = new DisplayMessage("HELLO", 2);
        DisplayMessage thread3 = new DisplayMessage("WELCOME", 3);
        thread1.start();
        thread2.start();
        thread3.start();
        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. The output shows the program's execution results. It starts with the author's name and phone number, followed by a sequence of messages from the three threads: 'GOOD MORNING', 'HELLO', and 'WELCOME'. These messages are printed in a staggered fashion, reflecting their respective intervals of 1, 2, and 3 seconds.

```
Abhay Raj, 00976803122
GOOD MORNING
HELLO
WELCOME
GOOD MORNING
HELLO
GOOD MORNING
WELCOME
GOOD MORNING
HELLO
GOOD MORNING
WELCOME
HELLO
HELLO
WELCOME
WELCOME
```

Program – 19

Aim: Design an applet to do the following tasks:

- a) To output the question "WHO IS THE PRIME MINISTER OF INDIA?"
- b) To accept the answer and print "CORRECT" and then stop if the answer is correct.
- c) To print "TRY AGAIN" and if the answer is not correct.
- d) To display the correct answer, if the answer is wrong even after third attempt.

Code:

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

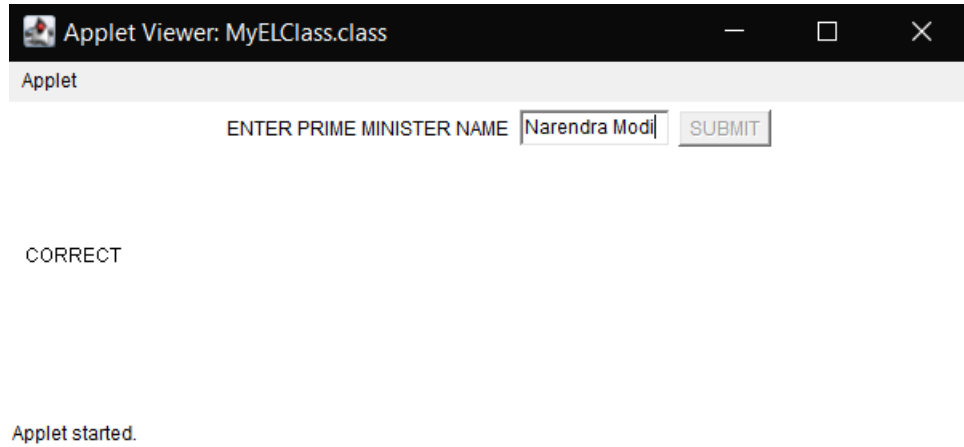
// <applet code=MyELClass width=600 height=600></applet>
public class MyELClass extends Applet implements ActionListener {
    String ans = "";
    Button sub;
    TextField tf1;
    int count = 0;

    public void init() {
        Label f = new Label("ENTER PRIME MINISTER NAME", Label.RIGHT);
        tf1 = new TextField(10);
        add(f);
        add(tf1);
        sub = new Button("SUBMIT");
        add(sub);
        sub.addActionListener(this);
    }

    public void actionPerformed(ActionEvent ae) {
        String name = tf1.getText();
        if (ae.getSource() == sub) {
            if (name.equals("Narendra Modi")) {
                ans = "CORRECT";
                sub.setEnabled(false);
            } else {
                ans = "INCORRECT";
                count++;
                if (count == 3) {
                    ans = "NAME OF PRIME MINISTER IS Sh. NARENDRA MODI";
                }
            }
        }
        repaint();
    }
}
```

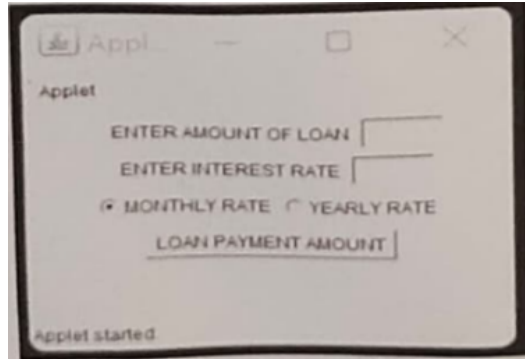
```
public void paint(Graphics g) {  
    g.drawString(ans, 10, 100);  
}  
}
```

Output:



Program – 20

Aim: Design a java program to write an applet that computes the payment of loan which is based on the amount of loan, interest rate and number of months. It takes one parameter from browser which is called Monthly Rate. If it is True then interest rate is per month else it is annual. (Use label, textfield, button, check box, checkbox group).



Code:

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
```

```
/* <applet code=MyELClass width=300 height=300></applet>
   <param name=N1 value=5>
*/
```

```
public class MyELClass extends Applet implements ItemListener, ActionListener {
    String s1;
    double loanAmount;
    double interestRate;
    double loanPayment;
    int n1; // Add declaration for n1
    TextField tf1, tf2;
    Button result;
    Checkbox cb1, cb2;
    CheckboxGroup cbg;
```

```
    public void init() {
        Label f = new Label("ENTER AMOUNT OF LOAN", Label.RIGHT);
        Label s = new Label("ENTER INTEREST RATE", Label.RIGHT);
        tf1 = new TextField(10);
        tf2 = new TextField(10);
        add(f);
        add(tf1);
        add(s);
        add(tf2);
```

```

cbg = new CheckboxGroup();
cb1 = new Checkbox("MONTHLY RATE", cbg, true);
cb2 = new Checkbox("YEARLY RATE", cbg, false);
result = new Button("CALCULATE LOAN PAYMENT AMOUNT");
add(cb1);
add(cb2);
add(result);
cb1.addItemListener(this);
cb2.addItemListener(this);
result.addActionListener(this);
s1 = getParameter("N1");
if (s1 != null) {
    n1 = Integer.parseInt(s1);
}
}

```

```

public void itemStateChanged(ItemEvent ie) {
    // No need for implementation here, as we're handling everything in actionPerformed
}

```

```

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == result) {

        loanAmount = Double.parseDouble(tf1.getText());
        interestRate = Double.parseDouble(tf2.getText());

        if (interestRate == 0 ) {
            showStatus("Please enter valid interest rate and loan term." + loanAmount +
interestRate);
            return;
        }

        if (cb1.getState()) {
            interestRate /= 12/100;
        }
        else{
            interestRate /= 100;
        }

        loanPayment = loanAmount * interestRate * 1;

        repaint();

    }
}

```



```
public void paint(Graphics g) {  
    g.drawString("LOAN PAYMENT AMOUNT: $" + String.format("%.2f",  
loanPayment), 20, 120);  
}  
}
```

Output:

Applet Viewer: MyELClass.class

Applet

ENTER AMOUNT OF LOAN ENTER INTEREST RATE

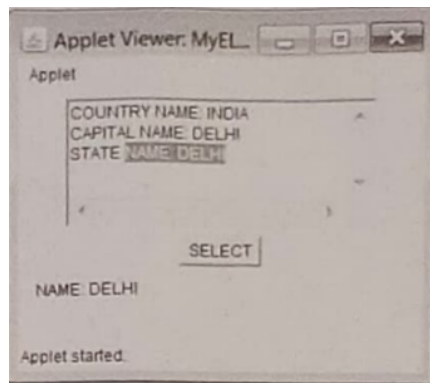
☐ MONTHLY RATE ☒ YEARLY RATE

LOAN PAYMENT AMOUNT: \$10.00

Applet started.

Program – 21

Aim: Design a java program to write an applet with following AWT components :textarea and button.



Code:

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
```

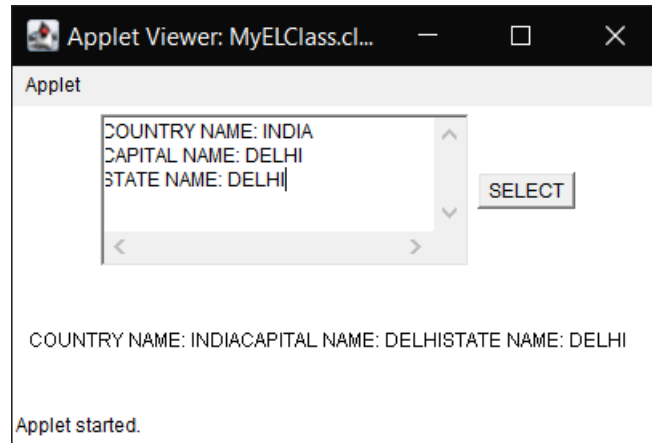
```
/* <applet code=MyELClass width=300 height=300></applet> */
public class MyELClass extends Applet implements ActionListener {
    String msg = "";
    String data = "";
    Button select;
    TextArea ta;

    public void init() {
        data = "COUNTRY NAME: INDIA\n" + "CAPITAL NAME: DELHI\n" +
            "STATE NAME: DELHI";
        ta = new TextArea(data, 5, 30);
        add(ta);
        select = new Button("SELECT");
        add(select);
        select.addActionListener(this);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == select) {
            msg = ta.getSelectedText();
        }
        repaint();
    }
}
```

```
public void paint(Graphics g) {  
    g.drawString(msg, 10, 150);  
}  
}
```

Output:



Program – 22

Aim: Design a java program to write a calculator applet. Use grid layout to arrange the buttons for digits for the +,-,*,/,% operations. Add text-field to display the result.

Code:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class MyELClass extends Applet implements ActionListener {

    TextField result;
    Button buttons[];
    String operator = "";

    public void init() {
        result = new TextField(20);
        result.setEditable(false);
        add(result);

        buttons = new Button[19];
        int k = 0;
        for (int i = 0; i < 10; i++) {
            buttons[k] = new Button("" + i);
            buttons[k].addActionListener(this);
            add(buttons[k]);
            k++;
        }

        buttons[k] = new Button("0");
        buttons[k].addActionListener(this);
        add(buttons[k]);
        k++;

        buttons[k] = new Button(".");
        buttons[k].addActionListener(this);
        add(buttons[k]);
        k++;

        buttons[k] = new Button("+");
        buttons[k].addActionListener(this);
        add(buttons[k]);
        k++;

        buttons[k] = new Button("-");
        buttons[k].addActionListener(this);
        add(buttons[k]);
        k++;

        buttons[k] = new Button("*");
        buttons[k].addActionListener(this);
```

```

add(buttons[k]);
k++;

buttons[k] = new Button("/");
buttons[k].addActionListener(this);
add(buttons[k]);
k++;

buttons[k] = new Button("%");
buttons[k].addActionListener(this);
add(buttons[k]);
k++;

buttons[k] = new Button("=");
buttons[k].addActionListener(this);
add(buttons[k]);
k++;

buttons[k] = new Button("C");
buttons[k].addActionListener(this);
add(buttons[k]);
k++;

setLayout(new GridLayout(5, 4));
}

public void actionPerformed(ActionEvent e) {
    String action = e.getActionCommand();
    if (Character.isDigit(action.charAt(0))) {
        result.setText(result.getText() + action);
    } else if (action.equals("+") || action.equals("-") || action.equals("*") || action.equals("/")
|| action.equals("%")) {
        operator = action;
        result.setText(result.getText() + action);
    } else if (action.equals("C")) {
        result.setText("");
        operator = "";
    } else if (action.equals(".")) {
        if (result.getText().indexOf(".") == -1) {
            result.setText(result.getText() + action);
        }
    } else if (action.equals("=")) {
        double number1 = Double.parseDouble(result.getText().substring(0,
result.getText().indexOf(operator)));
        double number2 =
Double.parseDouble(result.getText().substring(result.getText().indexOf(operator) + 1));

        double resultValue = 0;
        if (operator.equals("+")) {
            resultValue = number1 + number2;
        } else if (operator.equals("-")) {
            resultValue = number1 - number2;

```

```

    } else if (operator.equals("*")) {
        resultValue = number1 * number2;
    } else if (operator.equals("/")) {
        resultValue = number1 / number2;
    } else if (operator.equals("%")) {
        resultValue = number1 % number2;
    }

    result.setText(Double.toString(resultValue));
}
}
}

```

Output:



Program – 23

Aim: Design a java program to handle keyboard events.

Code:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class MyELClass extends Applet implements KeyListener {

    String message;

    public void init() {
        addKeyListener(this);
        message = "";
    }

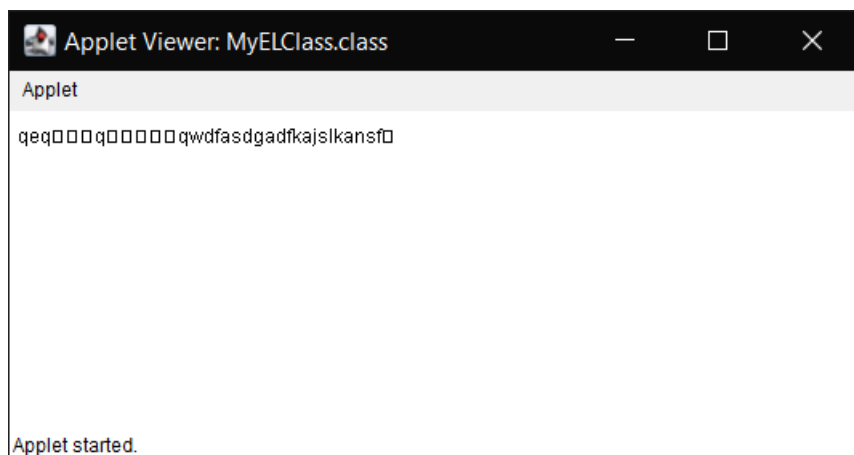
    public void keyPressed(KeyEvent e) {
        message = message + e.getKeyChar();
        repaint();
    }

    public void keyReleased(KeyEvent e) {
    }

    public void keyTyped(KeyEvent e) {
    }

    public void paint(Graphics g) {
        g.drawString(message, 5, 20);
    }
}
```

Output:



Program – 24

Aim: Design a program to Implement card layout in applets.

Code:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class MyELClass extends Applet implements ActionListener {
    CardLayout cardLayout;
    Panel cardPanel, inputPanel, resultPanel;
    TextField numField1, numField2, resultField;
    Button addButton, subtractButton, multiplyButton, divideButton;

    public void init() {
        cardLayout = new CardLayout();
        setLayout(cardLayout);

        cardPanel = new Panel(cardLayout);
        inputPanel = new Panel(new GridLayout(5, 1)); // Changed to 5 rows and 1 column
        resultPanel = new Panel(new GridLayout(2, 1));
        numField1 = new TextField();
        numField2 = new TextField();

        addButton = new Button("Addition");
        subtractButton = new Button("Subtraction");
        multiplyButton = new Button("Multiplication");
        divideButton = new Button("Division");

        addButton.addActionListener(this);
        subtractButton.addActionListener(this);
        multiplyButton.addActionListener(this);
        divideButton.addActionListener(this);

        inputPanel.add(new Label("Number 1:"));
        inputPanel.add(numField1);
        inputPanel.add(new Label("Number 2:"));
        inputPanel.add(numField2);
        inputPanel.add(addButton);
        inputPanel.add(subtractButton);
        inputPanel.add(multiplyButton);
        inputPanel.add(divideButton);

        resultField = new TextField();
        resultField.setEditable(false);

        resultPanel.add(new Label("Result:"));
        resultPanel.add(resultField);
    }
}
```



```

cardPanel.add(inputPanel, "input");
cardPanel.add(resultPanel, "result");

add(cardPanel);
}

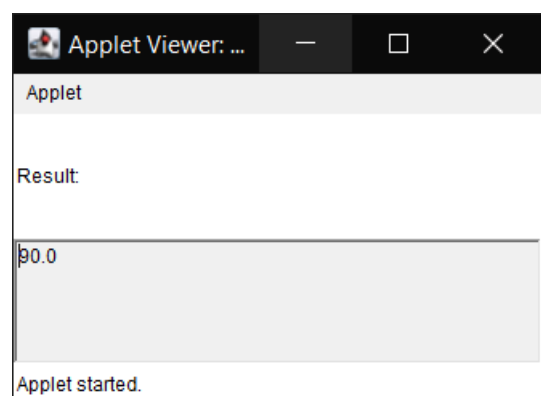
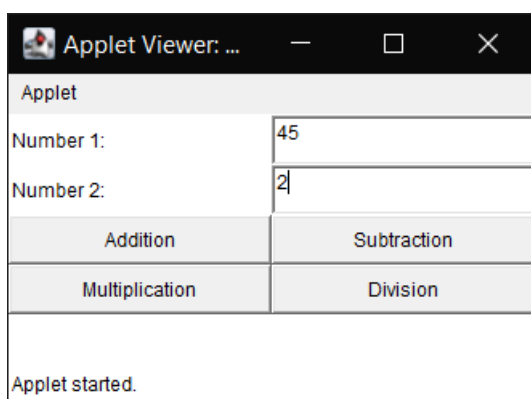
public void actionPerformed(ActionEvent e) {
    double num1 = Double.parseDouble(numField1.getText());
    double num2 = Double.parseDouble(numField2.getText());
    double result = 0;

    if (e.getSource() == addButton) {
        result = num1 + num2;
    } else if (e.getSource() == subtractButton) {
        result = num1 - num2;
    } else if (e.getSource() == multiplyButton) {
        result = num1 * num2;
    } else if (e.getSource() == divideButton) {
        if (num2 != 0) {
            result = num1 / num2;
        } else {
            resultField.setText("Error: Divide by zero");
            return;
        }
    }
}

resultField.setText(Double.toString(result));
cardLayout.show(cardPanel, "result");
}
}

```

Output:



Program – 25

Aim: Program In Java To Demonstrate FileWriter, FileReader, BufferedReader classes to Enter Multiple Strings And Numbers Into A File, Read File Using Loop And Display.

Code:

```
import java.io.*;

public class MyELClass {
    public static void main(String[] args) {
        String fileName = "data.txt";

        try (FileWriter writer = new FileWriter(fileName)) {
            BufferedWriter bufferedWriter = new BufferedWriter(writer);

            bufferedWriter.write("Hello, World!\n");
            bufferedWriter.write("This is a Java program.\n");

            bufferedWriter.write("123\n");
            bufferedWriter.write("456\n");

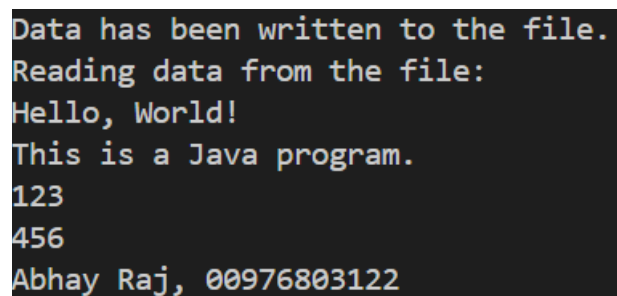
            bufferedWriter.close();
            System.out.println("Data has been written to the file.");
        } catch (IOException e) {
            System.err.println("Error writing to the file: " + e.getMessage());
        }

        try (FileReader reader = new FileReader(fileName)) {
            BufferedReader bufferedReader = new BufferedReader(reader);
            String line;

            System.out.println("Reading data from the file:");
            while ((line = bufferedReader.readLine()) != null) {
                System.out.println(line);
            }

            bufferedReader.close();
            System.out.println("Abhay Raj, 00976803122" );
        } catch (IOException e) {
            System.err.println("Error reading from the file: " + e.getMessage());
        }
    }
}
```

Output:

A screenshot of a terminal window showing the output of the Java program. The text is displayed in a monospaced font on a dark background. The output consists of several lines: 'Data has been written to the file.', 'Reading data from the file:', 'Hello, World!', 'This is a Java program.', '123', '456', and 'Abhay Raj, 00976803122'.

Program – 26

Aim: Program to implement various functions giving details about a file.

Code:

```
import java.io.File;

public class MyELClass {
    public static void main(String[] args) {
        String filePath = "LAB 1.txt";

        File file = new File(filePath);

        if (file.exists()) {
            System.out.println("File Name: " + file.getName());
            System.out.println("Absolute Path: " + file.getAbsolutePath());
            System.out.println("File Size in bytes: " + file.length());
            System.out.println("Readable: " + file.canRead());
            System.out.println("Writable: " + file.canWrite());
            System.out.println("File Type: " + getFileType(file));
        } else {
            System.out.println("File does not exist.");
        }

        System.out.println("Abhay Raj, 00976803122" );
    }

    public static String getFileType(File file) {
        String fileName = file.getName();
        int dotIndex = fileName.lastIndexOf('.');
        if (dotIndex > 0 && dotIndex < fileName.length() - 1) {
            return fileName.substring(dotIndex + 1).toUpperCase();
        }
        return "Unknown";
    }
}
```

Output:

```
File Name: LAB 1.txt
Absolute Path: C:\Users\_Not_Raj_\downloads\myjava\LAB 1.txt
File Size in bytes: 2750
Readable: true
Writable: true
File Type: TXT
Abhay Raj, 00976803122
```

Program – 27

Aim: Design a program in java to implement client-server TCP\IP socket application which exchange strings among them.

Code:

TCPServer.java

```
import java.io.*;
import java.net.*;

public class TCPServer {
    public static void main(String[] args) {
        final int PORT = 12345;

        try (ServerSocket serverSocket = new ServerSocket(PORT)) {
            System.out.println("Server waiting for client connection...");

            Socket clientSocket = serverSocket.accept();
            System.out.println("Client connected: " + clientSocket);

            // Setup input and output streams for communication
            BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

            String message;
            while ((message = in.readLine()) != null) {
                System.out.println("Received from client: " + message);
                out.println("Server received: " + message);
            }

            // Close resources
            in.close();
            out.close();
            clientSocket.close();
            System.out.println("Abhay Raj, 00976803122" );
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
Server waiting for client connection...
Client connected: Socket[addr=/127.0.0.1,port=50910,localport=12345]
Received from client: Hello from client!
Abhay Raj, 00976803122
```

TCPClient.java

```
import java.io.*;
import java.net.*;

public class TCPClient {
    public static void main(String[] args) {
        final String SERVER_IP = "localhost";
        final int SERVER_PORT = 12345;

        try (Socket socket = new Socket(SERVER_IP, SERVER_PORT);
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

            // Send a message to the server
            out.println("Hello from client!");

            // Receive and print server response
            String response = in.readLine();
            System.out.println("Server response: " + response);
            System.out.println("Abhay Raj, 00976803122" );
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
Server response: Server received: Hello from client!
Abhay Raj, 00976803122
```

Program – 28

Aim: Program In Java To Make Database Connectivity Using JDBC, create statement object, create resultset object And Display (All Or Specific Records using first(), last(), previous(), next() and absolute() functions). (Pre requirements: install MySql software, create database: Shiksha, create table house with attributes as H_No, H_Locality, H_Rooms, H_Owner, insert rows of data into table house)

Code:

```
import java.sql.*;
class DBConnectivity
{
    public Connection CO;
    void makeConnection()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            CO=DriverManager.getConnection("jdbc:mysql://localhost:3306/shiksha", "root", "root");
        }
        catch(Exception Ee)
        {
            System.out.println(Ee);
        }
    }

    void displayFirst()
    {
        try
        {
            ResultSet RS;
            Statement SMT;
            SMT =
            CO.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
            String QU="SELECT * FROM HOUSE";
            RS=SMT.executeQuery(QU);
            RS.first();
            System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
            RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));

            RS.close();
            SMT.close();
        }
        catch(Exception Ee)
        {
            System.out.println(Ee);
        }
    }

    void displayThird()
    {
        try
        {
            ResultSet RS;
            Statement SMT;
            SMT = CO.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
```

```

        String QU="SELECT * FROM HOUSE";
        RS=SMT.executeQuery(QU);
        RS.absolute(3);
        System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
            RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));

        RS.close();
        SMT.close();
    }
    catch(Exception Ee)
    {
        System.out.println(Ee);
    }
}

void displayLast()
{
    try
    {
        ResultSet RS;
        Statement SMT;
        SMT = CO.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
        String QU="SELECT * FROM HOUSE";
        RS=SMT.executeQuery(QU);
        RS.last();
        System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
            RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));

        RS.close();
        SMT.close();
    }
    catch(Exception Ee)
    {
        System.out.println(Ee);
    }
}

void closeConnection()
{
    try
    {
        CO.close();
    }
    catch(Exception Ee)
    {
        System.out.println(Ee);
    }
}

void displayNextAll()
{
    try
    {
        Statement SMT;
        ResultSet RS;
        SMT = CO.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
        String QU="SELECT * FROM HOUSE";
        RS=SMT.executeQuery(QU);
        RS.first();
        System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
            RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));
    }
}

```

```

        while(RS.next())
        {
            System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
            RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));
        }

        RS.close();
        SMT.close();
    }
    catch(Exception Ee)
    {
        System.out.println("Exception Caught for Next.....");
    }
}

```

```

void displayPreviousAll()
{
    try
    {
        Statement SMT;
        ResultSet RS;
        SMT = CO.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_READ_ONLY);
        String QU="SELECT * FROM HOUSE";
        RS=SMT.executeQuery(QU);
        RS.last();
        System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
        RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));

        while(RS.previous())
        {
            System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
            RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));
        }

        RS.close();
        SMT.close();
    }
    catch(Exception Ee)
    {
        System.out.println("Exception Caught for Next.....");
    }
}
}

```

```

public class MyELClass
{
    public static void main(String args[])
    {
        DBConnectivity Obj=new DBConnectivity();
        Obj.makeConnection();
    }
}

```



```
System.out.println("DIPSLAYING FIRST RECORD ");
Obj.displayFirst();
System.out.println("DIPSLAYING THIRD RECORD ");
Obj.displayThird();
System.out.println("DIPSLAYING LAST RECORD ");
Obj.displayLast();
System.out.println("DIPSLAYING ALL RECORDS ");
Obj.displayNextAll();
System.out.println("DIPSLAYING ALL RECORDS ");
Obj.displayPreviousAll();
Obj.closeConnection();

System.out.println("Abhay Raj, 00976803122" );
}
}
```

Output:

```
DIPSLAYING FIRST RECORD
1ABC Street3John Doe
DIPSLAYING THIRD RECORD
3PQR Road4Michael Johnson
DIPSLAYING LAST RECORD
4LMN Lane5Emily Brown
DIPSLAYING ALL RECORDS
1ABC Street3John Doe
2XYZ Avenue2Jane Smith
3PQR Road4Michael Johnson
4LMN Lane5Emily Brown
DIPSLAYING ALL RECORDS
4LMN Lane5Emily Brown
3PQR Road4Michael Johnson
2XYZ Avenue2Jane Smith
1ABC Street3John Doe
Abhay Raj, 00976803122
```

Program – 29

Aim: Program In Java To Make Database Connectivity Using JDBC, create statement object, create resultset object, insert a record using prepared statement and Display all records. (Pre requirements: install MySql software, create database: Shiksha, create table house with attributes as H_No, H_Locality, H_Rooms, H_Owner, insert rows of data into table house)

Code:

```
import java.sql.*;
import java.util.*;

class DBConnectivity
{
    public Connection CO;

    void makeConnection()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            CO=DriverManager.getConnection("jdbc:mysql://localhost:3306/shiksha", "root", "root");
        }
        catch(Exception Ee)
        {
            System.out.println(Ee);
        }
    }

    void closeConnection()
    {
        try
        {
            CO.close();
        }

        catch(Exception Ee)
        {
            System.out.println(Ee);
        }
    }

    void displayNextAll()
    {
        try
        {
            Statement SMT;
            ResultSet RS;
            SMT = CO.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
            String QU="SELECT * FROM HOUSE";
            RS=SMT.executeQuery(QU);
            RS.first();
            System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
            RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));

            while(RS.next())
            {
```

```

        System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
        RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));
    }

    RS.close();
    SMT.close();
}
catch(Exception Ee)
{
    System.out.println("Exception Caught for Next.....");
}
}

```

```

void insertRecord()

```

```

{
    int Hnum,Hrooms;
    String Hloc, Howner;
    Scanner sc=new Scanner(System.in);
    try
    {
        Statement SMT;
        String QU;
        SMT = CO.createStatement();
        System.out.println("ENTER HOUSE NUMBER ");
        Hnum=sc.nextInt();
        System.out.println("ENTER HOUSE LOCALITY ");
        Hloc=sc.next();
        System.out.println("ENTER Mr. / Mrs. ");
        String salu=sc.next();
        System.out.println("ENTER HOUSE OWNER NAME ");
        Howner=sc.next();
        Howner=salu+" "+Howner;
        System.out.println("ENTER HOUSE ROOMS ");
        Hrooms=sc.nextInt();
        QU="INSERT INTO HOUSE VALUES(" + Hnum + "," + Hloc + "," +
        Hrooms + "," + Howner + ")";
        SMT.executeUpdate(QU);
        SMT.close();
    }
    catch(Exception Ee)
    {
        System.out.println("Exception Caught During Insert .....");
    }
}

```

```

void insertRecordUsingPrepared()

```

```

{
    int Hnum,Hrooms;
    String Hloc, Howner;
    Scanner sc=new Scanner(System.in);
    try

```

```

    {
        PreparedStatement PSMT;
        String QU;
        PSMT = CO.prepareStatement("INSERT INTO HOUSE VALUES(?,?,?,?)");
        System.out.println("ENTER HOUSE NUMBER ");
        Hnum=sc.nextInt();
        System.out.println("ENTER HOUSE LOCALITY ");
        Hloc=sc.next();
        System.out.println("ENTER Mr. / Mrs. ");
        String salu=sc.next();
        System.out.println("ENTER HOUSE OWNER NAME ");
        Howner=sc.next();
        Howner=salu+" "+Howner;
        System.out.println("ENTER HOUSE ROOMS ");
        Hrooms=sc.nextInt();
        PSMT.setInt(1,Hnum);
        PSMT.setString(2,Hloc);
        PSMT.setInt(3,Hrooms);
        PSMT.setString(4,Howner);
        PSMT.executeUpdate();
        PSMT.close();
    }
    catch(Exception Ee)
    {
        System.out.println("Exception Caught During Insert .....");
        System.out.println(Ee);
    }
}
}

public class MyELClass
{
    public static void main(String args[])
    {
        DBConnectivity Obj=new DBConnectivity();
        Obj.makeConnection();
        //Obj.insertRecord();
        Obj.insertRecordUsingPrepared();
        Obj.displayNextAll();
        Obj.closeConnection();

        System.out.println("Abhay Raj, 00976803122" );
    }
}

```

Output:

```
ENTER HOUSE NUMBER
13
ENTER HOUSE LOCALITY
ZZZ Avenue
ENTER Mr. / Mrs.
ENTER HOUSE OWNER NAME
ROSHAN
ENTER HOUSE ROOMS
4
1ABC Street3John Doe
2XYZ Avenue2Jane Smith
3PQR Road4Michael Johnson
4LMN Lane5Emily Brown
10ZZZ1DELHI RAJ
12ZZZ1STREET MOHAN
13ZZZ4Avenue ROSHAN
23RRR3Street Rohan
Abhay Raj, 00976803122
```

Program – 30

Aim: Program In Java To Make Database Connectivity Using JDBC, create statement object, create resultset object, update a record based on H_No and Display all records. (Pre requirements: install MySql software, create database: Shiksha, create table house with attributes as H_No, H_Locality, H_Rooms, H_Owner, insert rows of data into table house)

Code:

```
import java.sql.*;
import java.util.*;

class DBConnectivity
{
    public Connection CO;
    void makeConnection()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            CO=DriverManager.getConnection("jdbc:mysql://localhost:3306/shiksha", "root", "root");
        }
        catch(Exception Ee)
        {
            System.out.println(Ee);
        }
    }

    void closeConnection()
    {
        try
        {
            CO.close();
        }
        catch(Exception Ee)
        {
            System.out.println(Ee);
        }
    }

    void displayNextAll()
    {
        try
        {
            Statement SMT;
            ResultSet RS;
            SMT = CO.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
            String QU="SELECT * FROM HOUSE";
            RS=SMT.executeQuery(QU);
            RS.first();
            System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
            RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));

            while(RS.next())
            {
                System.out.println(RS.getInt("H_NO")+RS.getString("H_LOCALITY")+
                RS.getInt("H_ROOMS")+RS.getString("H_OWNER"));
            }
        }
    }
}
```

```

        RS.close();
        SMT.close();
    }
    catch(Exception Ee)
    {
        System.out.println("Exception Caught for Next.....");
    }
}

void updateRecord()
{
    int Hnum,Hrooms;
    String Hloc, Howner;
    Scanner sc=new Scanner(System.in);
    try
    {
        Statement SMT;
        String QU;
        SMT = CO.createStatement();

        System.out.println("ENTER HOUSE NUMBER TO UPDATE ");
        Hnum=sc.nextInt();

        System.out.println("ENTER HOUSE LOCALITY ");
        Hloc=sc.next();
        System.out.println("ENTER Mr. / Mrs. ");
        String salu=sc.next();
        System.out.println("ENTER HOUSE OWNER NAME ");
        Howner=sc.next();
        Howner=salu+" "+Howner;
        System.out.println("ENTER HOUSE ROOMS ");
        Hrooms=sc.nextInt();

        QU="UPDATE HOUSE SET H_LOCALITY = '" + Hloc + "', H_ROOMS = " + Hrooms +
        ", H_OWNER = '" + Howner + "' WHERE H_NO = " + Hnum + " ";
        SMT.executeUpdate(QU);
        SMT.close();
    }

    catch(Exception Ee)
    {
        System.out.println("Exception Caught During Update .....");
    }
}

public class MyELClass
{
    public static void main(String args[])
    {
        DBConnectivity Obj=new DBConnectivity();
    }
}

```

```
        Obj.makeConnection();
        Obj.updateRecord();
        Obj.displayNextAll();
        Obj.closeConnection();

        System.out.println("Abhay Raj, 00976803122" );
    }
}
```

Output:

```
ENTER HOUSE NUMBER TO UPDATE
23
ENTER HOUSE LOCALITY
DDD House
ENTER Mr. / Mrs.
ENTER HOUSE OWNER NAME
RANU
ENTER HOUSE ROOMS
5
1ABC Street3John Doe
2XYZ Avenue2Jane Smith
3PQR Road4Michael Johnson
4LMN Lane5Emily Brown
10ZZZ1DELHI RAJ
12ZZZ1STREET MOHAN
13ZZZ4Avenue ROSHAN
23DDD5House RANU
Abhay Raj, 00976803122
```


Program – 31

Aim: Program In Java To Make Database Connectivity Using JDBC, create statement object, create resultset object, delete a record based on H_No and Display all records. (Pre requirements: install MySql software, create database: Shiksha, create table house with attributes as H_No, H_Locality, H_Rooms, H_Owner, insert rows of data into table house)

Code:

```
import java.io.*;
import java.util.*;
import java.sql.*;

class DBConnectivity
{
    public Connection CO;
    void makeConnection()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            CO=DriverManager.getConnection("jdbc:mysql://localhost:3306/shiksha", "root", "root");
        }
        catch(Exception Ee)
        {
            System.out.println(Ee);
        }
    }

    void closeConnection()
    {
        try
        {
            CO.close();
        }
        catch(Exception Ee)
        {
            System.out.println(Ee);
        }
    }

    void displayNextAll()
    {
        try
        {
            Statement SMT;
            ResultSet RS;
            SMT = CO.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
            String QU="SELECT * FROM HOUSE";
            RS=SMT.executeQuery(QU);
            RS.first();
            System.out.println(RS.getInt("H_NO")+“ ”+RS.getString("H_LOCALITY")
+“ ”+
            RS.getInt("H_ROOMS")+“ ”+RS.getString("H_OWNER"));

            while(RS.next())
            {
```

```

+“ ”+
        System.out.println(RS.getInt("H_NO")+“ ”+RS.getString("H_LOCALITY")
        RS.getInt("H_ROOMS")+“ ”+RS.getString("H_OWNER"));
    }

    RS.close();
    SMT.close();
}
catch(Exception Ee)
{
    System.out.println("Exception Caught for Next.....");
}
}

```

```

void deleteRecord()
{
    int Hnum;
    Scanner sc=new Scanner(System.in);
    try
    {
        Statement SMT;
        String QU;
        SMT = CO.createStatement();

        System.out.println("ENTER HOUSE NUMBER, TO DELETE ");
        Hnum=sc.nextInt();

        QU="DELETE FROM HOUSE WHERE H_NO = "+Hnum+"";
        SMT.executeUpdate(QU);

        SMT.close();
    }
    catch(Exception Ee)
    {
        System.out.println("Exception Caught During Delete .....");
    }
}
}

```

```

public class MyELClass
{
    public static void main(String args[])
    {
        DBConnectivity Obj=new DBConnectivity();
        Obj.makeConnection();
        Obj.deleteRecord();
        Obj.displayNextAll();
        Obj.closeConnection();

        System.out.println("Abhay Raj, 00976803122" );
    }
}

```

Output:

```
ENTER HOUSE NUMBER, TO DELETE  
1  
2 XYZ Avenue 2 Jane Smith  
3 PQR Road 4 Michael Johnson  
4 LMN Lane 5 Emily Brown  
10 ZZZ 1 DELHI RAJ  
12 ZZZ 1 STREET MOHAN  
13 ZZZ 4 Avenue ROSHAN  
Abhay Raj, 00976803122
```