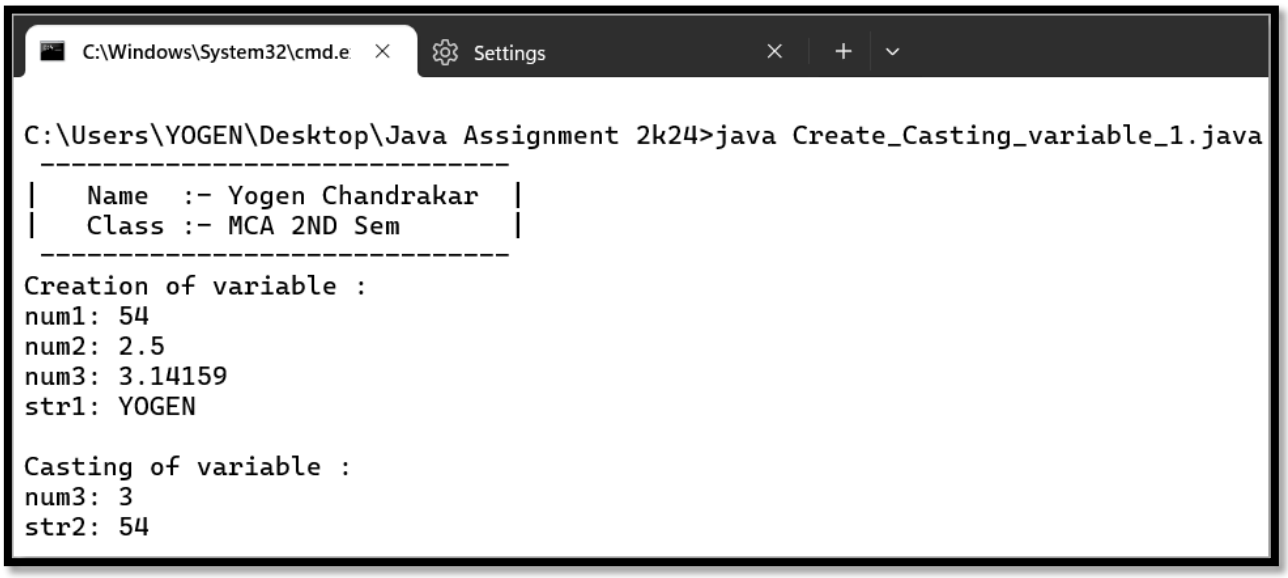
Program No.1

Write a Java program for Creation and Casting of Variables.

Code:-

```
import myinfo.Myinfo;
public class Create_Casting_variable_1 {
  public static void main(String[] args){
      Myinfo mc = new Myinfo();
      mc.display();
      System.out.println("Creation of variable : ");
      // create an integer variable

      int num1 = 54;
      System.out.println("num1: " + num1);
      // create a float variable
      float num2 = 2.5f;
      System.out.println("num2: " + num2);
      // create a double variable
      double num3 = 3.14159;
      System.out.println("num3: " + num3);
      // create a string variable
      String str1 = "YOGEN";
      System.out.println("str1: " + str1);

      System.out.println("\nCasting of variable : ");

      // cast the double variable to an integer
      int num4 = (int) num3;
      System.out.println("num3: " + num4);

      // cast the integer variable to a string
      String str2 = Integer.toString(num1);
      System.out.println("str2: " + str2);
      }
}
```

Output :-



Program No.2

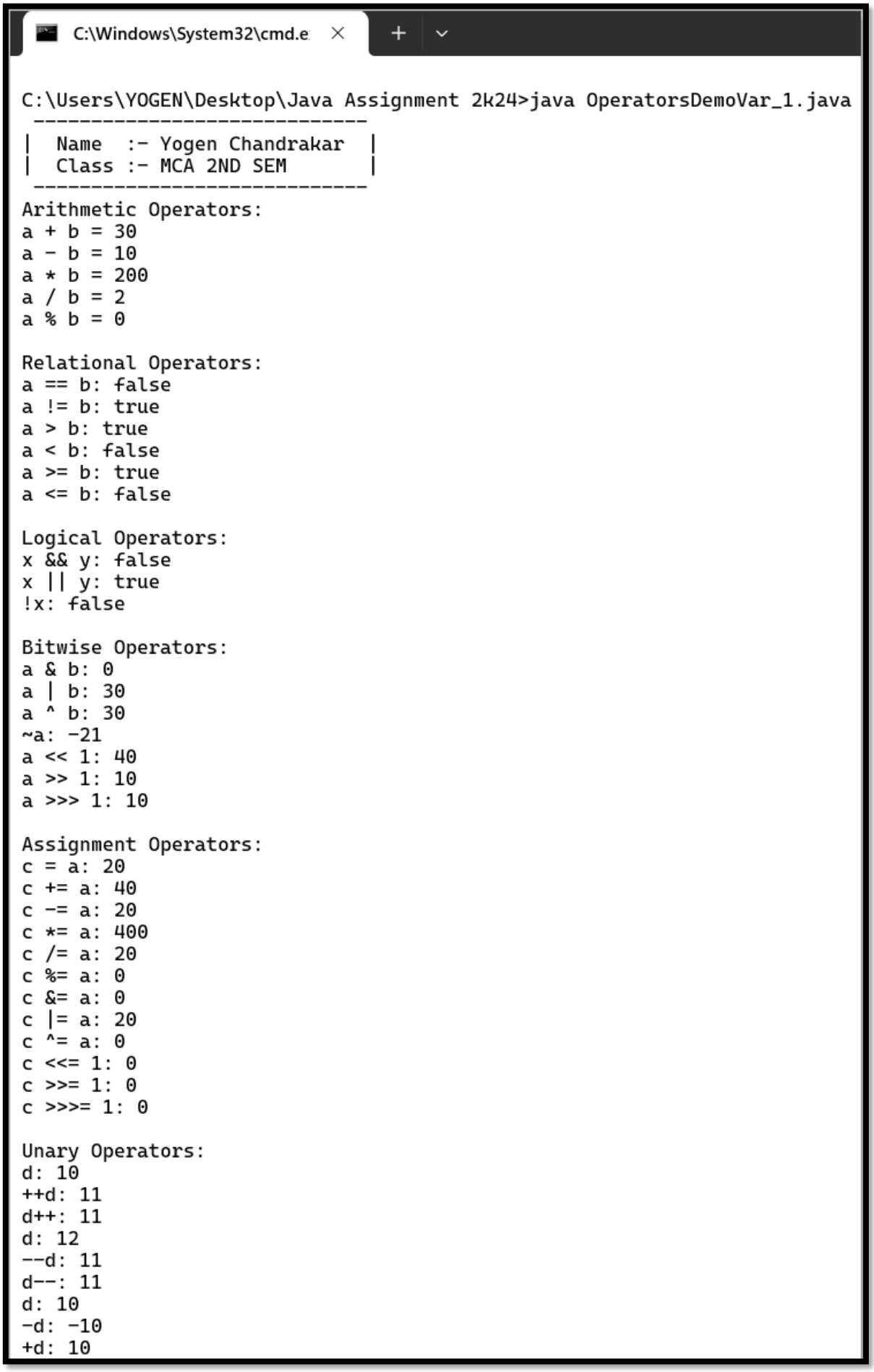
Write a Java program to demonstrate the various Operators.

Code:-

```
public class OperatorsDemoVar_1{
  public static void main(String[] args) {
    System.out.println("|  Name  :- Yogen Chandrakar  |");
    System.out.println("|  Class :- MCA 2ND SEM       |");

        // Arithmetic Operators
        int a = 20;
        int b = 10;
        System.out.println("Arithmetic Operators:");
        System.out.println("a + b = " + (a + b));
        System.out.println("a - b = " + (a - b));
        System.out.println("a * b = " + (a * b));
        System.out.println("a / b = " + (a / b));
        System.out.println("a % b = " + (a % b));

        // Relational Operators
        System.out.println("\nRelational Operators:");
        System.out.println("a == b: " + (a == b));
        System.out.println("a != b: " + (a != b));
        System.out.println("a > b: " + (a > b));
        System.out.println("a < b: " + (a < b));
        System.out.println("a >= b: " + (a >= b));
        System.out.println("a <= b: " + (a <= b));

        // Logical Operators
        boolean x = true;
        boolean y = false;
        System.out.println("\nLogical Operators:");
        System.out.println("x && y: " + (x && y));
        System.out.println("x || y: " + (x || y));
        System.out.println("!x: " + (!x));

        // Bitwise Operators
        System.out.println("\nBitwise Operators:");
        System.out.println("a & b: " + (a & b));
        System.out.println("a | b: " + (a | b));
        System.out.println("a ^ b: " + (a ^ b));
        System.out.println("~a: " + (~a));
        System.out.println("a << 1: " + (a << 1));
        System.out.println("a >> 1: " + (a >> 1));
        System.out.println("a >>> 1: " + (a >>> 1));

        // Assignment Operators
        int c;
        System.out.println("\nAssignment Operators:");

            c = a;
```

```
System.out.println("c = a: " + c);
 c += a;
 System.out.println("c += a: " + c);
 c -= a;
 System.out.println("c -= a: " + c);
 c *= a;
 System.out.println("c *= a: " + c);
 c /= a;
 System.out.println("c /= a: " + c);
 c %= a;
 System.out.println("c %= a: " + c);
 c &= a;
 System.out.println("c &= a: " + c);
 c |= a;
 System.out.println("c |= a: " + c);
 c ^= a;
 System.out.println("c ^= a: " + c);
 c <<= 1;
 System.out.println("c <<= 1: " + c);
 c >>= 1;
 System.out.println("c >>= 1: " + c);
 c >>>= 1;
 System.out.println("c >>>= 1: " + c);

 // Unary Operators
 System.out.println("\nUnary Operators:");
 int d = 10;
 System.out.println("d: " + d);
 System.out.println("++d: " + (++d));
 System.out.println("d++: " + (d++));
 System.out.println("d: " + d);
 System.out.println("--d: " + (--d));
 System.out.println("d--: " + (d--));
 System.out.println("d: " + d);
 System.out.println("-d: " + (-d));
 System.out.println("+d: " + (+d));
     }
 }
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java OperatorsDemoVar_1.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND SEM |
Arithmetic Operators:
a + b = 30
a - b = 10
a * b = 200
a / b = 2
a % b = 0
Relational Operators:
a == b: false
a != b: true
a > b: true
a < b: false
a >= b: true
a <= b: false
Logical Operators:
x && y: false
x || y: true
!x: false
Bitwise Operators:
a & b: 0
a | b: 30
a ^ b: 30
~a: -21
a << 1: 40
a >> 1: 10
a >>> 1: 10
Assignment Operators:
c = a: 20
c += a: 40
c -= a: 20
c *= a: 400
c /= a: 20
c %= a: 0
c &= a: 0
c |= a: 20
c ^= a: 0
c <<= 1: 0
c >>= 1: 0
c >>>= 1: 0
Unary Operators:
d: 10
++d: 11
d++: 11
d: 12
--d: 11
d--: 11
d: 10
-d: -10
+d: 10

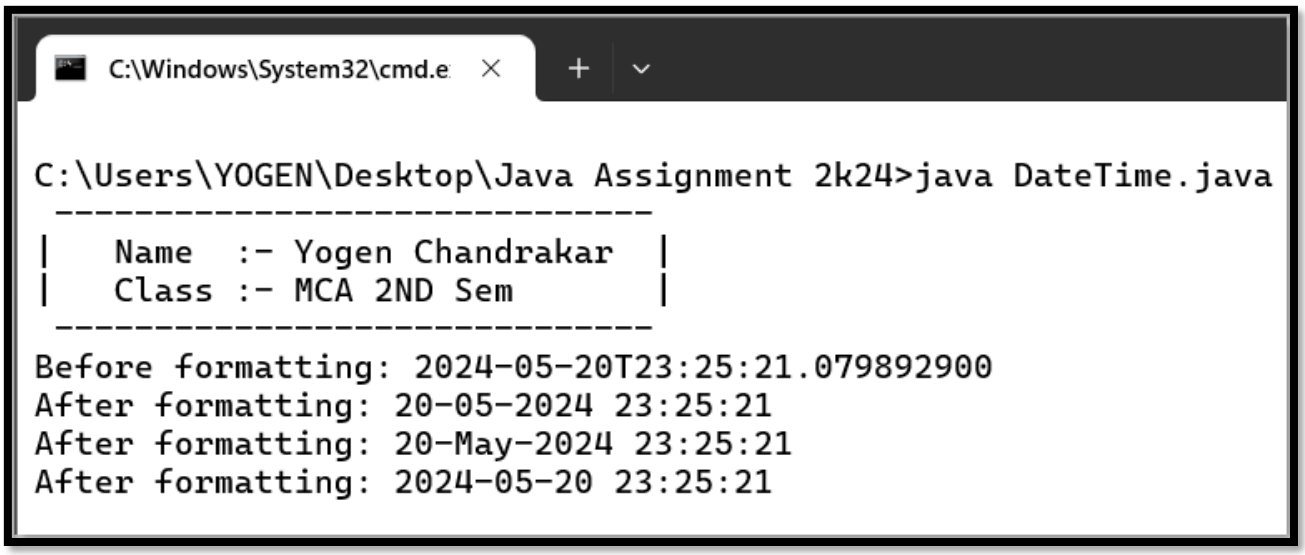
Program No.3

Write a Java program for printing the current date in different formats.

Code:-

```
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import myinfo.Myinfo;
public class DateTime{
public static void main(String[] args) {
   Myinfo mc = new Myinfo();
   mc.display();
   LocalDateTime myDateObj = LocalDateTime.now();
   System.out.println("Before formatting: " + myDateObj);

   DateTimeFormatter myFormatObj = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");
   String formattedDate1 = myDateObj.format(myFormatObj);
   System.out.println("After formatting: " + formattedDate1);


   DateTimeFormatter myFormatObj2 = DateTimeFormatter.ofPattern("dd-MMM-yyyy HH:mm:ss");
   String formattedDate2 = myDateObj.format(myFormatObj2);
   System.out.println("After formatting: " + formattedDate2);


   DateTimeFormatter myFormatObj3 = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
   String formattedDate3 = myDateObj.format(myFormatObj3);
   System.out.println("After formatting: " + formattedDate3);
 }
}
```

Output :-



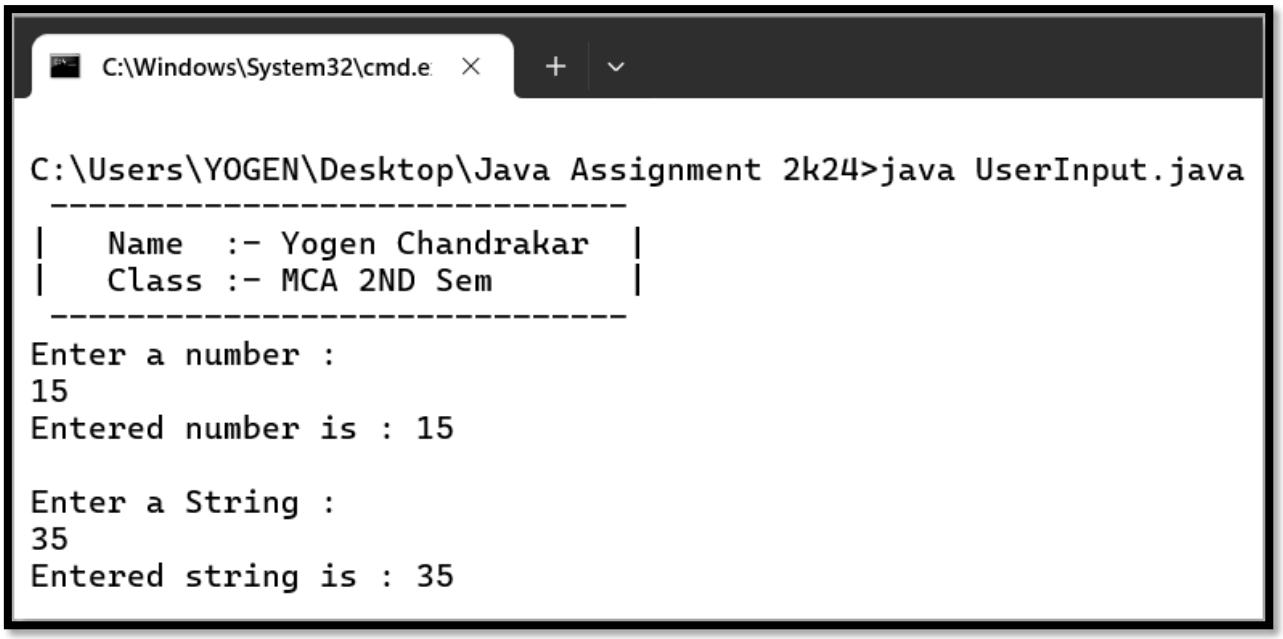
Program No.4

Write a Java program for Inputting Data From Keyboard through Scanner Class

Code:-

```
import java.util.Scanner;
import myinfo.Myinfo;
class UserInput_4{
   public static void main(String args[]){
      Myinfo m = new Myinfo();
      m.display();
      Scanner obj=new Scanner(System.in);
      System.out.println("Enter a number : ");
      int num = obj.nextInt();
      System.out.println("Entered number is : "+num);
      System.out.println("\nEnter a String : ");
      String str = obj.next();
      System.out.println("Entered string is : "+str);
   }
}
```

Output :-



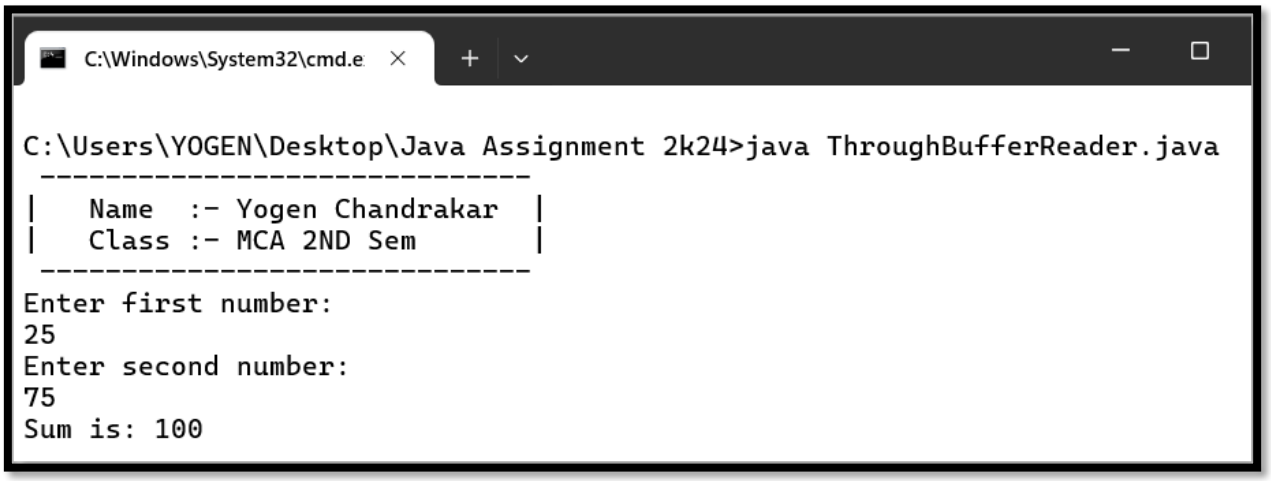
Program No.5

Write a Java program for Inputting Data From Keyboard through BufferedReader Class.

Code:-

```
import java.io.*;
import myinfo.Myinfo;
class ThroughBufferReader{
   public static void main(String args[])throws IOException{
      Myinfo m = new Myinfo();
      m.display();
      int a,b,sum;
      BufferedReader ob=new BufferedReader(new InputStreamReader(System.in));
      System.out.println("Enter first number: ");
      a = Integer.parseInt(ob.readLine());
      System.out.println("Enter second number: ");
      b = Integer.parseInt(ob.readLine());
      sum=a+b;
      System.out.println("Sum is: "+sum);
   }
}
```

Output :-



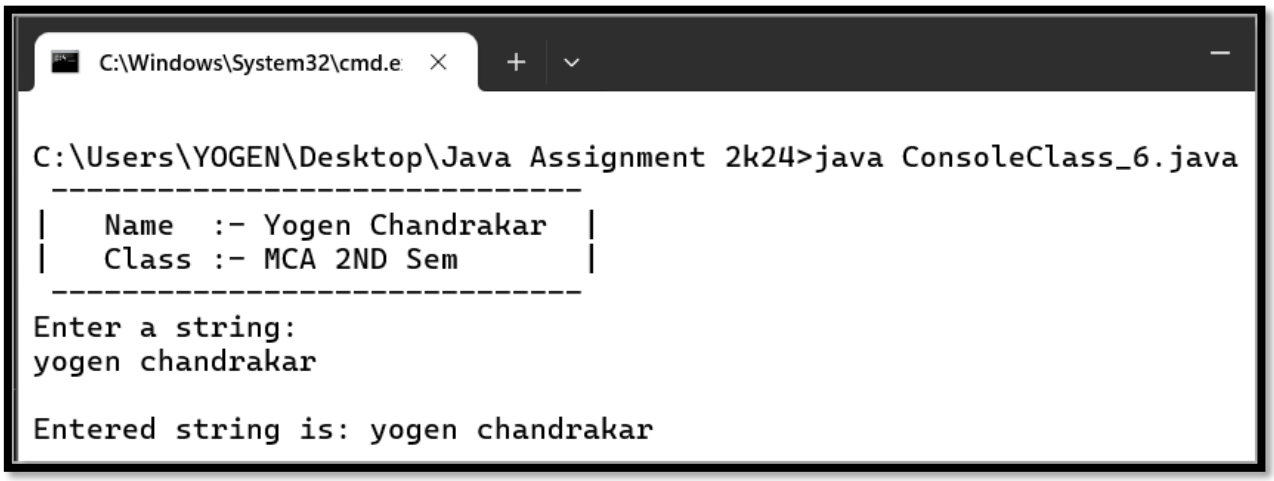
Program No.6

Write a Java program for Inputting Data from Keyboard through Console Class.

Code:-

```
import java.io.Console;
import myinfo.Myinfo;
class ConsoleClass_6{
   public static void main(String args[]){
      Myinfo m = new Myinfo();
      m.display();

      Console obj=System.console();
      System.out.println("Enter a string: ");
      String str=obj.readLine();
      System.out.println("\nEntered string is: "+str);
   }
}
```

Output :-



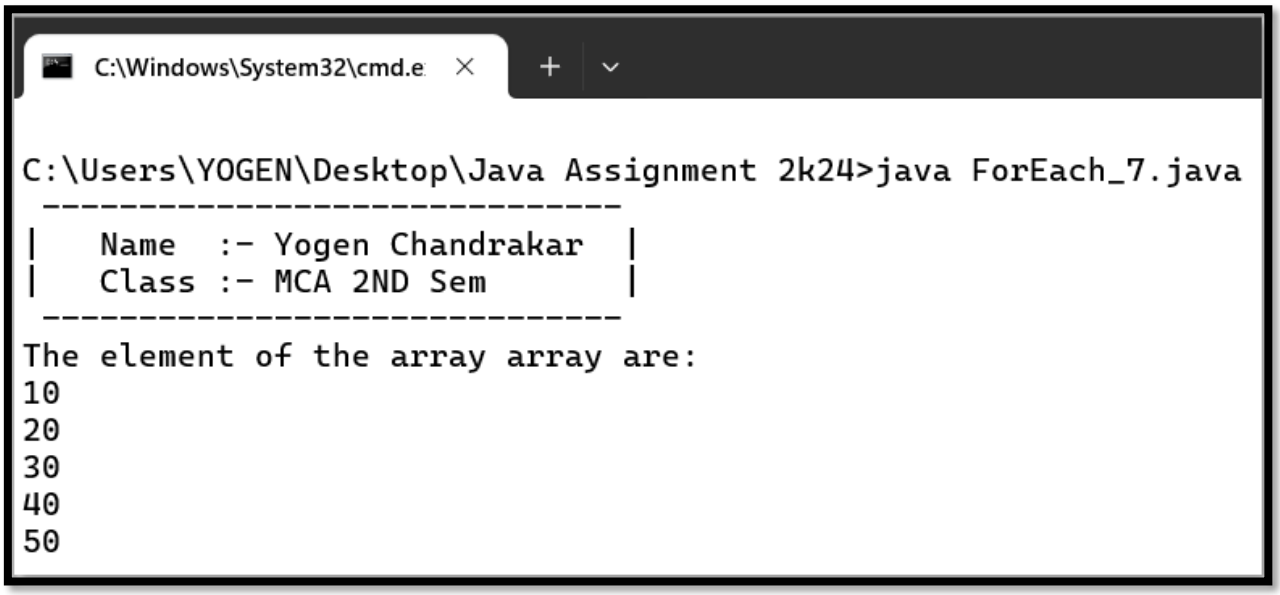
Program No.7

Write a Java program to demonstrate the use of for–each loop.

Code:-

```
import myinfo.Myinfo;
class ForEach_7{
  public static void main(String args[]){
        Myinfo m = new Myinfo();
        m.display();

   //Creating an array
   int arr[]={10,20,30,40,50};
   //traversing the array with for-each loop
   System.out.println("The element of the array are: ");
   for(int i:arr){
     System.out.println(i);
   }
 }
}
```

Output :-



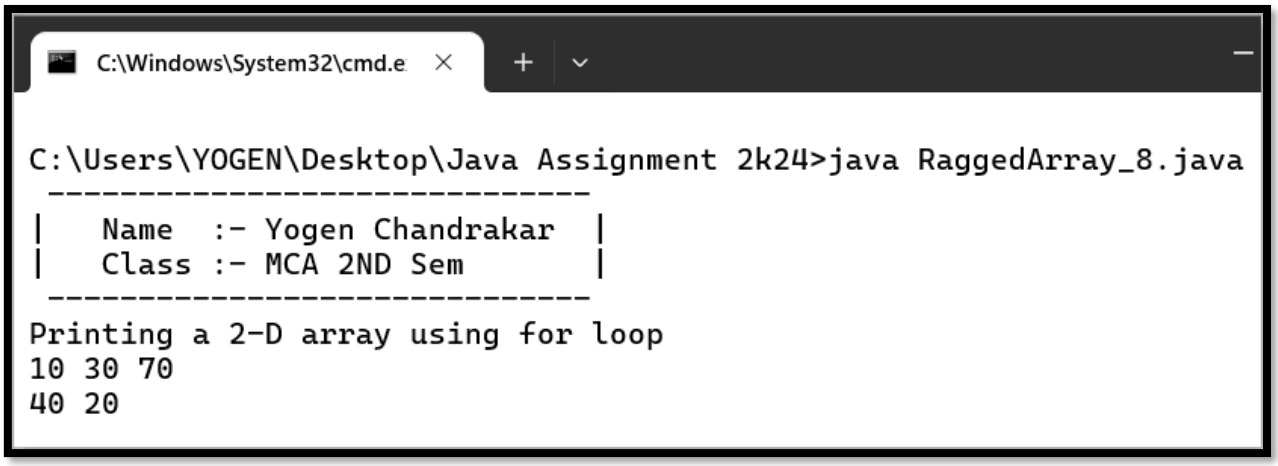
Program No.8

Write a Java program to demonstrate ragged arrays.

Code:-

```
import myinfo.Myinfo;
public class RaggedArray{
   public static void main(String[] args) {
      Myinfo m = new Myinfo();
      m.display();

      int [][] rag_array = new int [2][];
      rag_array[0] = new int[]{10,30,70};
      rag_array[1] = new int[]{40,20};

      // Displaying the 2-D Array (for loop)
      System.out.println("Printing a 2-D array using for loop");
      for(int i=0;i<rag_array.length;i++){
         for(int j=0;j<rag_array[i].length;j++) {
            System.out.print(rag_array[i][j]);
            System.out.print(" ");
         }
         System.out.println("");
      }

   }
}
```

Output :-



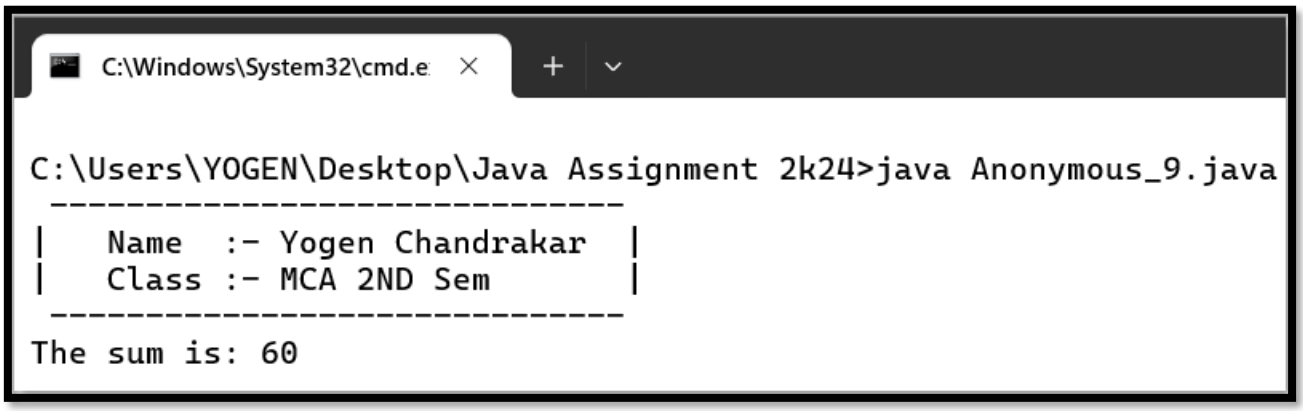
Program No.9

Write a Java program to demonstrate anonymous arrays.

Code:-

```
import myinfo.Myinfo;
class Anonymous_9 {
   public static void main(String[] args){
      Myinfo m = new Myinfo();
      m.display();
      // Anonymous array
      sum(new int[]{ 10, 20, 30 });
   }

   public static void sum(int[] a)
   {
      int total = 0;
      for (int i : a)
         total = total + i;

      System.out.println("The sum is: " + total);
   }
}
```

Output :-



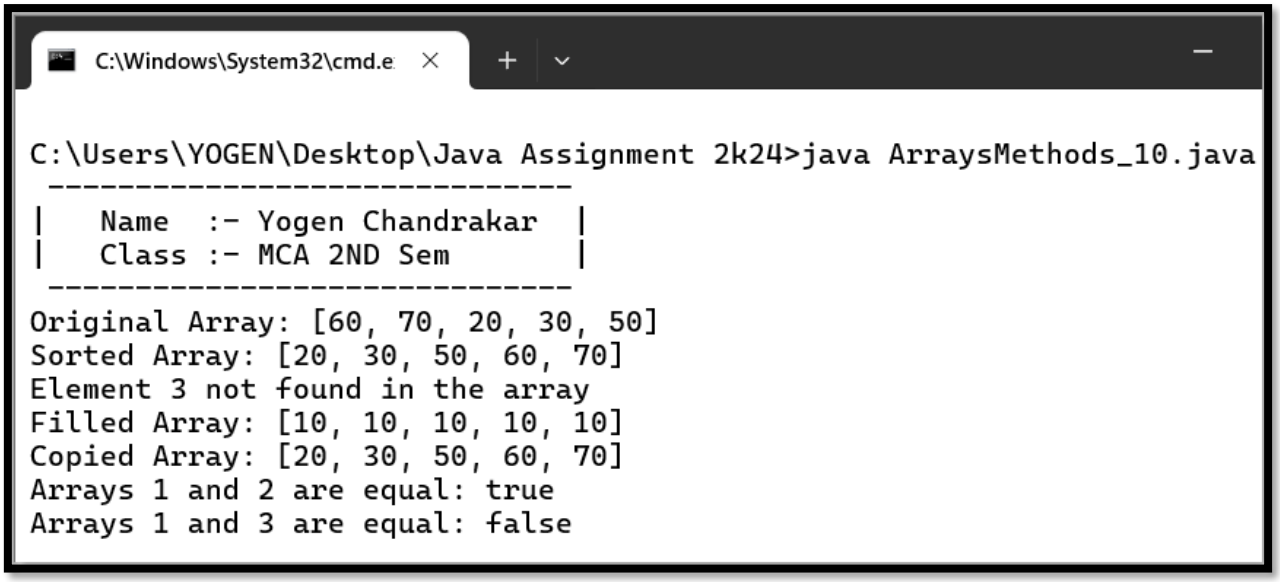
Program No.10

Write a Java program to demonstrate the methods of Arrays Class.

Code:-

```
import java.util.Arrays;
import myinfo.Myinfo;
public class ArraysMethods_10{

   public static void main(String[] args) {
      Myinfo m = new Myinfo();
      m.display();

      int[] numbers = {60,70,20,30,50};
      System.out.println("Original Array: " + Arrays.toString(numbers));

      // Sorting
      Arrays.sort(numbers);
      System.out.println("Sorted Array: " + Arrays.toString(numbers));

      // Search
      int searchElement = 3;
      int index = Arrays.binarySearch(numbers, searchElement);
      if (index >= 0) {
         System.out.println("Element " + searchElement + " found at index " + index);
      } else {
         System.out.println("Element " + searchElement + " not found in the array");
      }

      // Filling the array with a specific value
      int[] filledArray = new int[5];
      Arrays.fill(filledArray, 10);
      System.out.println("Filled Array: " + Arrays.toString(filledArray));

      // Copy
      int[] copiedArray = Arrays.copyOf(numbers, numbers.length);
      System.out.println("Copied Array: " + Arrays.toString(copiedArray));

      // Compare
      int[] array1 = {1, 2, 3};
      int[] array2 = {1, 2, 3};
      int[] array3 = {1, 2, 4};
      System.out.println("Arrays 1 and 2 are equal: " + Arrays.equals(array1, array2));
      System.out.println("Arrays 1 and 3 are equal: " + Arrays.equals(array1, array3));
   }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java ArraysMethods_10.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
Original Array: [60, 70, 20, 30, 50]
Sorted Array: [20, 30, 50, 60, 70]
Element 3 not found in the array
Filled Array: [10, 10, 10, 10, 10]
Copied Array: [20, 30, 50, 60, 70]
Arrays 1 and 2 are equal: true
Arrays 1 and 3 are equal: false

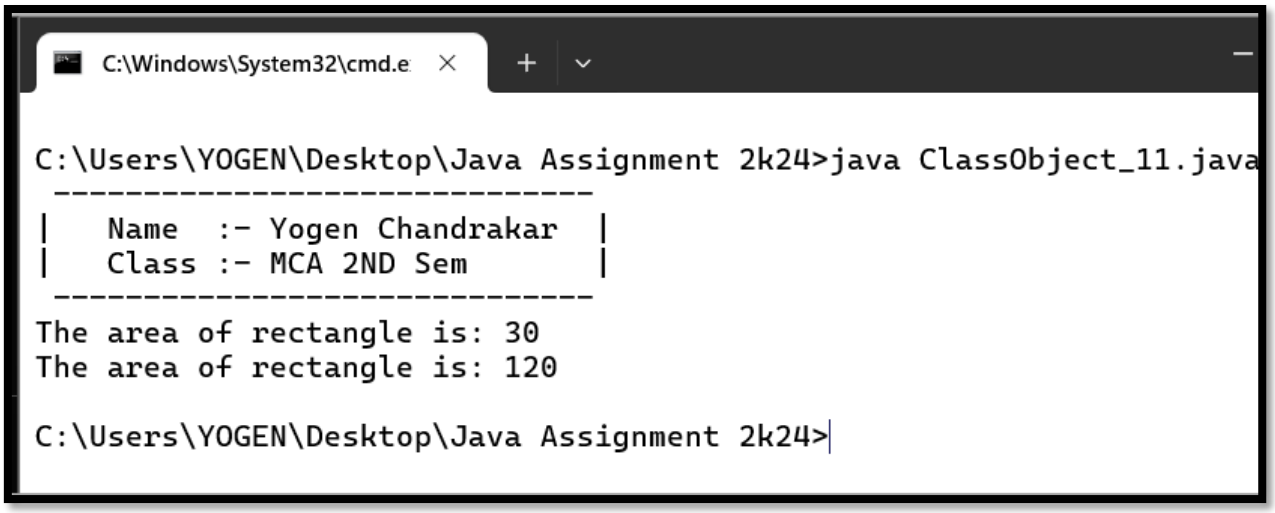
Program No.11

Write a Java program for Application Of Classes And Objects.

Code:-

```
import myinfo.Myinfo;
class ClassObject_11 {
    public static void main(String args[]) {
        Myinfo m = new Myinfo();
        m.display();
        // Creating Rectangle objects with constructor initialization
        Rectangle r1 = new Rectangle(5, 6);
        Rectangle r2 = new Rectangle(12, 10);

        // Calling calculateArea method
        r1.calculateArea();
        r2.calculateArea();
    }
}
class Rectangle {
    int length;
    int width;

    // Constructor to initialize length and width
    Rectangle(int l, int w) {
        length = l;
        width = w;
    }
    void calculateArea() {
        System.out.println("The area of rectangle is: " + length * width);
    }
}
```

Output :-



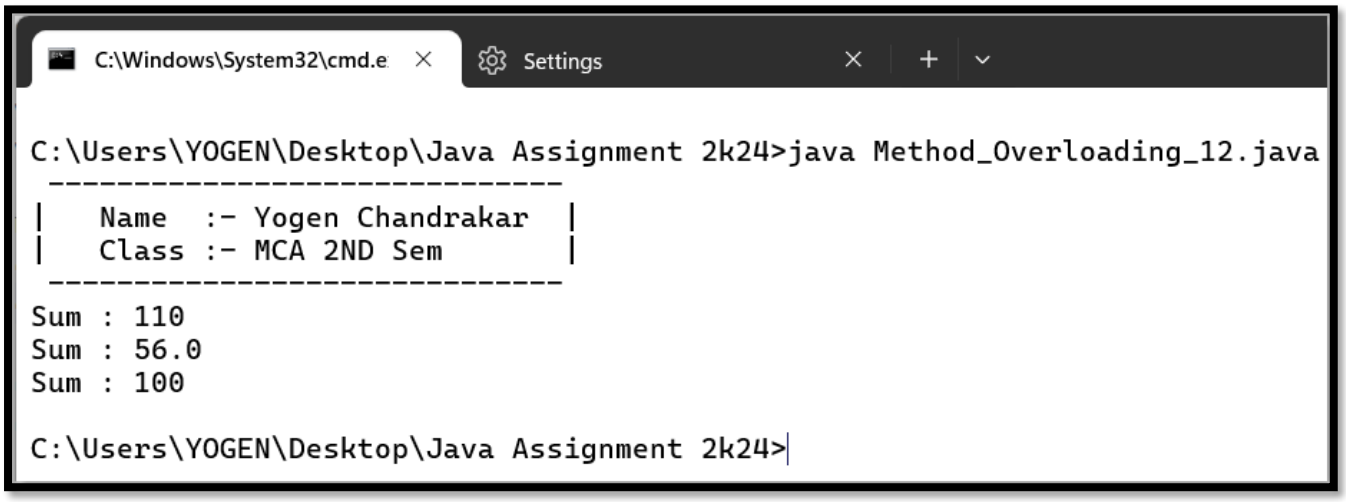
Program No.12

Write a Java program to demonstrate method overloading.

Code:-

```
import myinfo.Myinfo;
class Method_Overloading_12{
   public static void main(String args[]){
      Myinfo mc = new Myinfo();
      mc.display();
      Calculator c = new Calculator();
      System.out.println("Sum : "+c.sum(11,99));
      System.out.println("Sum : "+c.sum((float)11.55,(float)44.45));
      System.out.println("Sum : "+c.sum(11,79,10));
   }
}
class Calculator{
   int sum(int a,int b){
      return a+b;
   }
   float sum(float a,float b){
      return a+b;
   }
   int sum(int a,int b,int c){
      return a+b+c;
   }
}
```

Output :-



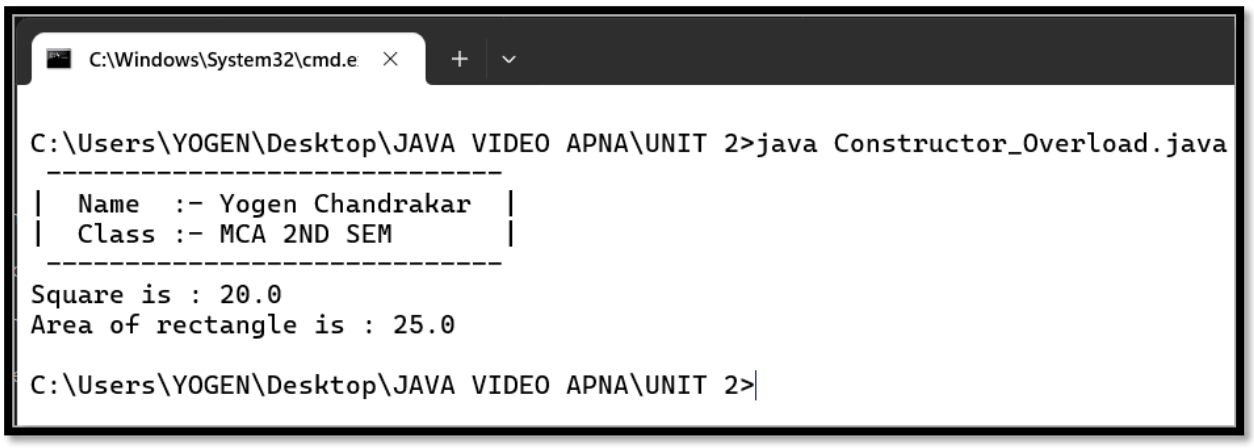
Program No.13

Write a Java program to demonstrate constructor overloading.

Code:-

```
class Student{
   String name;
   int num1;
   int num2;

   Student(int num1,int num2){
      this.num1 = num1;
      this.num2=num2;
   }
   Student(int num1){
      this.num1 = num1;
   }
   int SqureDisplay(){
     return num1*num1;
   }
    int AreaRectangle(){
      return num1*num2;
   }
}
class Constructor_Overload{
   public static void main(String args[]){
      System.out.println(" ----------------------------");
      System.out.println("|  Name  :- Yogen Chandrakar  |");
      System.out.println("|  Class :- MCA 2ND SEM       |");
      System.out.println(" ----------------------------");
      Student obj1= new Student(4,5);
      Student obj2 = new Student(5);

      float result1 = obj1.AreaRectangle();
      float result2 = obj2.SqureDisplay();

      System.out.println("Square is : "+result1);
      System.out.println("Area of rectangle is : "+result2);
   }}
```

Output :-



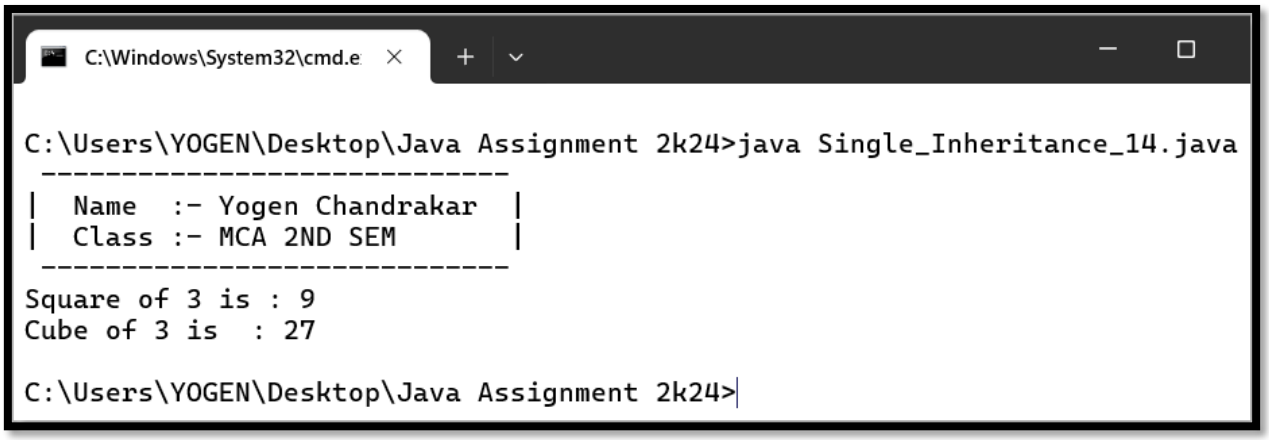
Program No.14

Write a Java program Using Single Inheritance.

Code:-

```
class Single_Inheritance_14{
    public static void main (String args[]){
          System.out.println(" ----------------------------");
          System.out.println("|  Name  :- Yogen Chandrakar  |");
           System.out.println("|  Class :- MCA 2ND SEM       |");
           System.out.println(" ----------------------------");
           b = new B();
           b.square();
           b.cube();
             }
  }
    class A{
      int x = 3 ;
      public void square(){
              System.out.println("Square of "+x+" is : "+(x*x));
            }
        }
    class B extends A{              // extends Class A
           public  void cube(){
             System.out.println("Cube of "+x+" is  : "+(x*x*x));
            }
     }
```

Output :-



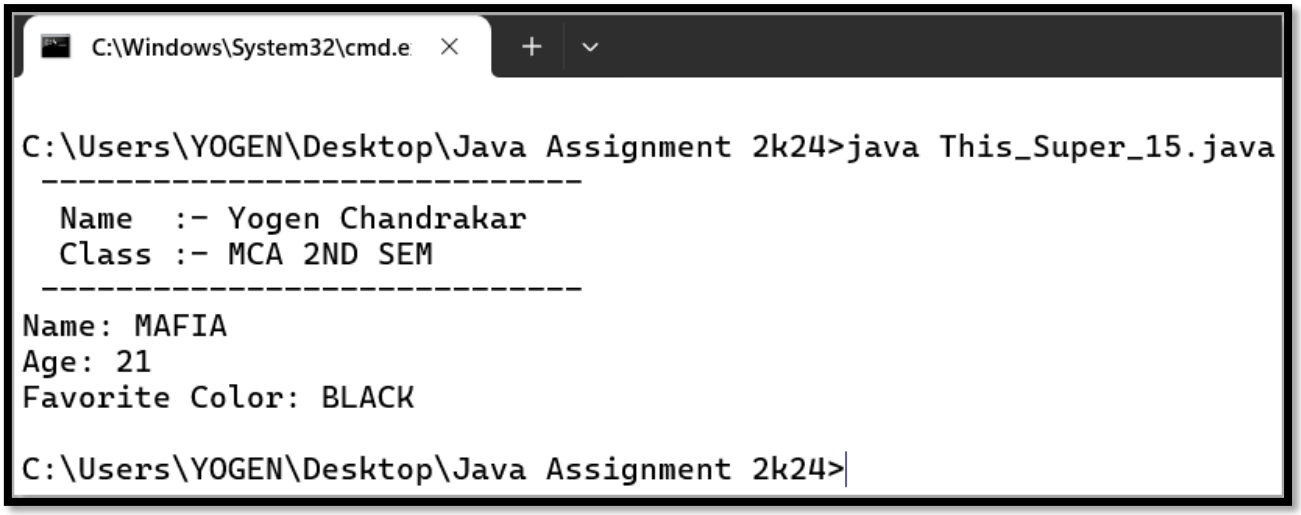
Program No.15

Write a Java program Using Super And This Keyword.

Code:-

```
public class This_Super_15{
 public static void main(String[] args) {
       System.out.println("  Name  :- Yogen Chandrakar  ");
       System.out.println("  Class :- MCA 2ND SEM       ");
       System.out.println(" ----------------------------");

 Child child = new Child("MAFIA", 21, "BLACK");
   child.display();       // Call the display() method of the Child class
   }
}
// The Parent class
class Parent {
   String name;
   Parent(String name) {   // use for name initialization
   this.name = name;     // this point current object
 }
   void display() {
   System.out.println("Name: " + name);
 }
}
// The Child class
class Child extends Parent {
   int age;
   String favoriteColor;
   Child(String name, int age, String favoriteColor) {
   super(name);
   this.age = age;
   this.favoriteColor = favoriteColor;
   }
   void display() {
      super.display();
      System.out.println("Age: " + age);
      System.out.println("Favorite Color: " + favoriteColor);
   }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java This_Super_15.java
Name :- Yogen Chandrakar
Class :- MCA 2ND SEM
Name: MAFIA
Age: 21
Favorite Color: BLACK
C:\Users\YOGEN\Desktop\Java Assignment 2k24>

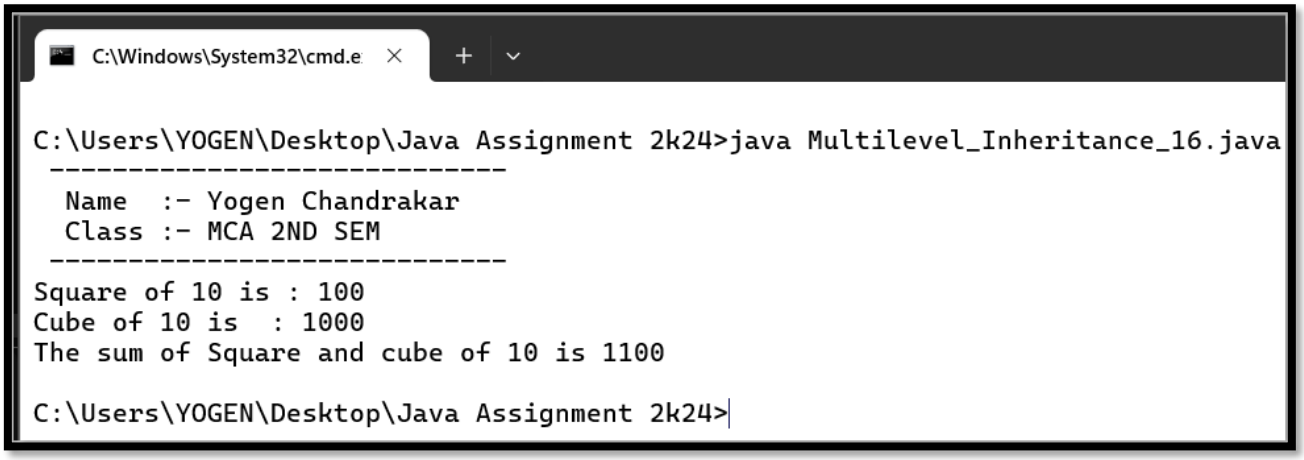
Program No.16

Write a Java program to demonstrate multilevel inheritance.

Code:-

```
class Multilevel_Inheritance_16{
      public static void main (String args[]){
         System.out.println(" ----------------------------");
         System.out.println("  Name  :- Yogen Chandrakar  ");
         System.out.println("  Class :- MCA 2ND SEM       ");
         System.out.println(" ----------------------------");
         C c = new C();
         c.squareDisp();
         c.cubeDisp();
         c.sum();
      }
   }
   class A{
      int x = 10 ;
         int square =x*x;
         public void squareDisp(){
         System.out.println("Square of "+x+" is : "+square);
      }
   }
   class B extends A{
       int cube = x*x*x;
       public  void cubeDisp(){
       System.out.println("Cube of "+x+" is  : "+cube);
      }
   }
   class C extends B{
      public void sum(){
         int sum = square + cube;
         System.out.println("The sum of Square and cube of "+x+" is "+sum);
      }
   }
```

Output :-



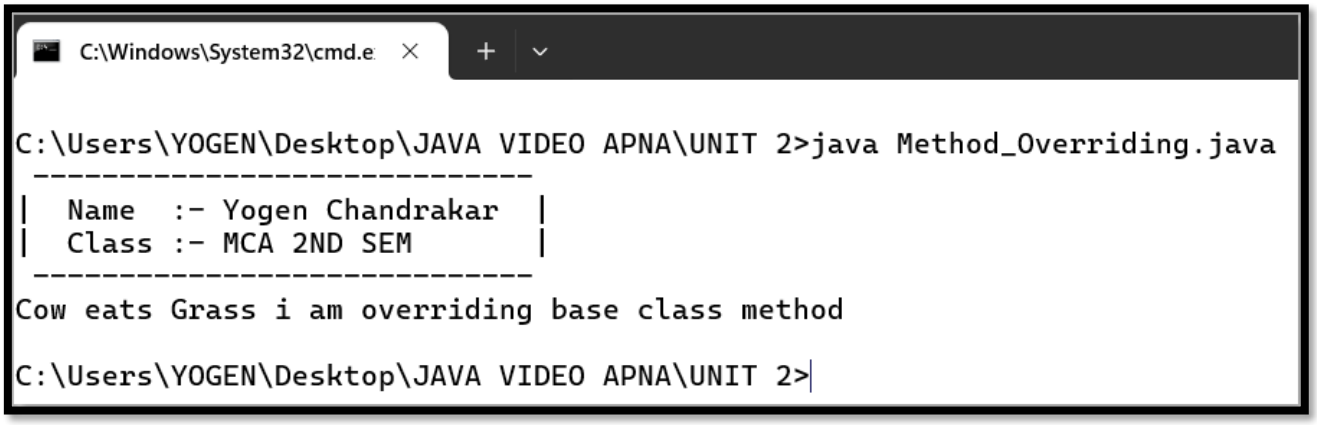
Program No.17

Write a Java program to demonstrate method overriding.

Code:-

```
class Method_Overriding {
   public static void main(String args[]){
      System.out.println(" ----------------------------");
      System.out.println("|  Name  :- Yogen Chandrakar  |");
      System.out.println("|  Class :- MCA 2ND SEM       |");
      System.out.println(" ----------------------------");
      Cow c = new Cow();
      c.eat();  // Eats Grass
   }
}
class Animals{
   void eat(){
      System.out.println("Eats anything");
      }
   }
class Cow extends Animals{
   void eat(){
      System.out.println("Cow eats Grass i am overriding base class method ");
   }
}
```

Output :-



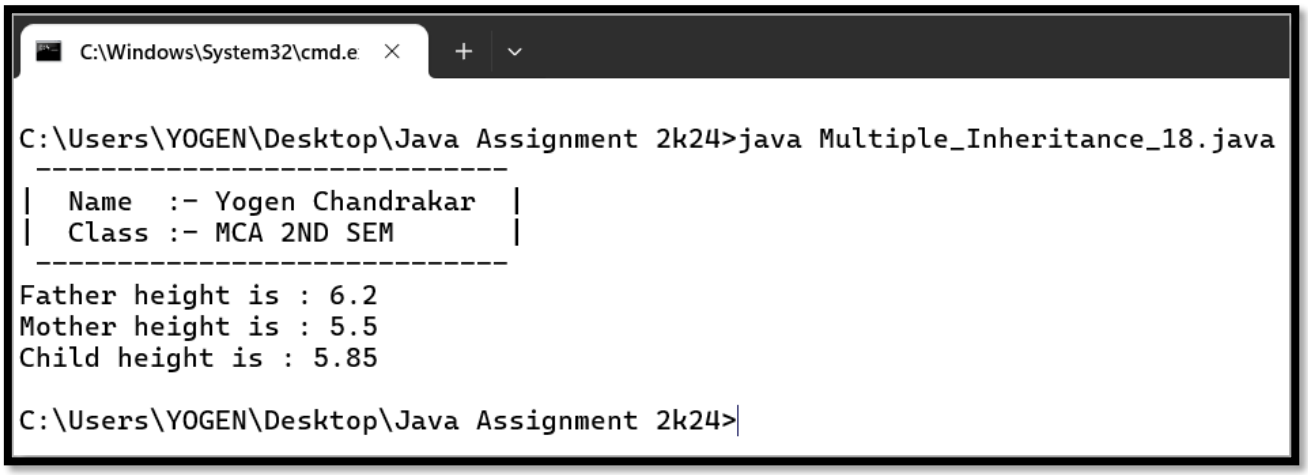
Program No.18

Write a Java program Using Multiple Inheritance Concept through interfaces.

Code:-

```
class Multiple_Inheritance_18{
    public static void main(String args[]){
        System.out.println(" ----------------------------");
        System.out.println("|  Name  :- Yogen Chandrakar  |");
        System.out.println("|  Class :- MCA 2ND SEM       |");
        System.out.println(" ----------------------------");
        Child ch =new Child();
        ch.height();
    }
}
interface Father{
    float HT = 6.2f;
    void height();
}
interface Mother{
    float HT =5.5f;
    void height();
}
class Child implements Father,Mother{
    public void height(){    // imlimentation of inteface method
        float ht = (Father.HT+Mother.HT)/2;
        System.out.println("Father height is : "+Father.HT);
        System.out.println("Mother height is : "+Mother.HT);
        System.out.println("Child height is : "+ht);
    }
}
```

Output :-



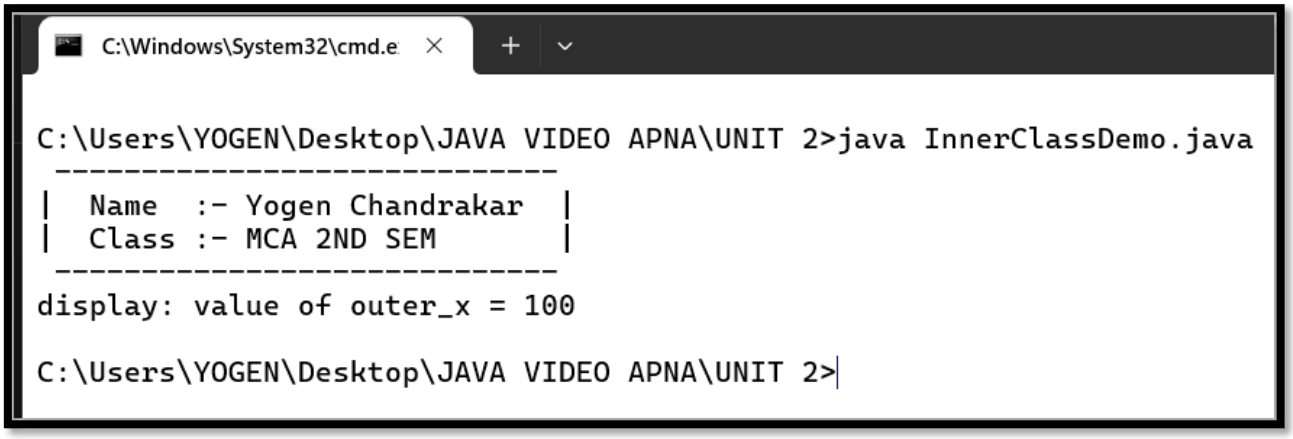
Program No.19

Write a Java program to demonstrate the concept of inner class.

Code:-

```
class InnerClassDemo {
    public static void main(String args[]) {
      System.out.println(" ----------------------------");
      System.out.println("|  Name  :- Yogen Chandrakar  |");
      System.out.println("|  Class :- MCA 2ND SEM       |");
      System.out.println(" ----------------------------");
    Outer outer = new Outer();
    outer.test();
 }
}
class Outer {
    int outer_x = 100;
    void test() {
    Inner inner = new Inner();  // object create of inner class
    inner.display();
 }
 // this is an inner class
 class Inner {
    void display() {
    System.out.println("display: value of outer_x = " + outer_x); //access the variable outer_x i.e x
    }
 }
```

Output :-



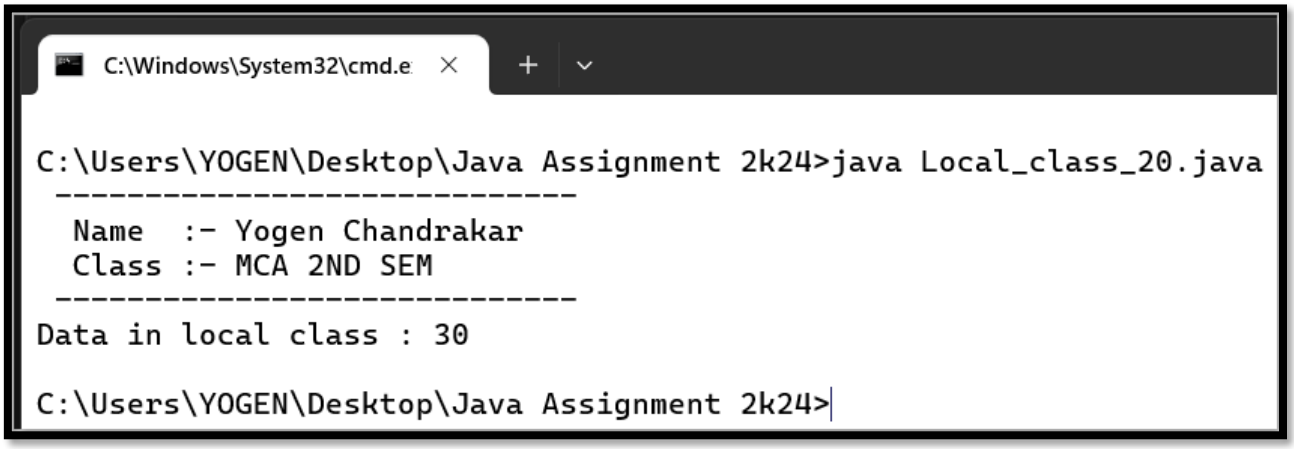
Program No.20

Write a Java program to demonstrate the concept of local class.

Code:-

```
public class Local_class_20{
 private int data=30;//instance variable
 void display(){
  class Local{
   void msg(){
    System.out.println("Data in local class : "+data);
   }
  }
  Local l=new Local();
  l.msg();
 }
 public static void main(String args[]){
       System.out.println(" ----------------------------");
       System.out.println("  Name  :- Yogen Chandrakar  ");
       System.out.println("  Class :- MCA 2ND SEM       ");
       System.out.println(" ----------------------------");

  Local_class_20 obj=new Local_class_20();
  obj.display();
 }
}
```

Output :-



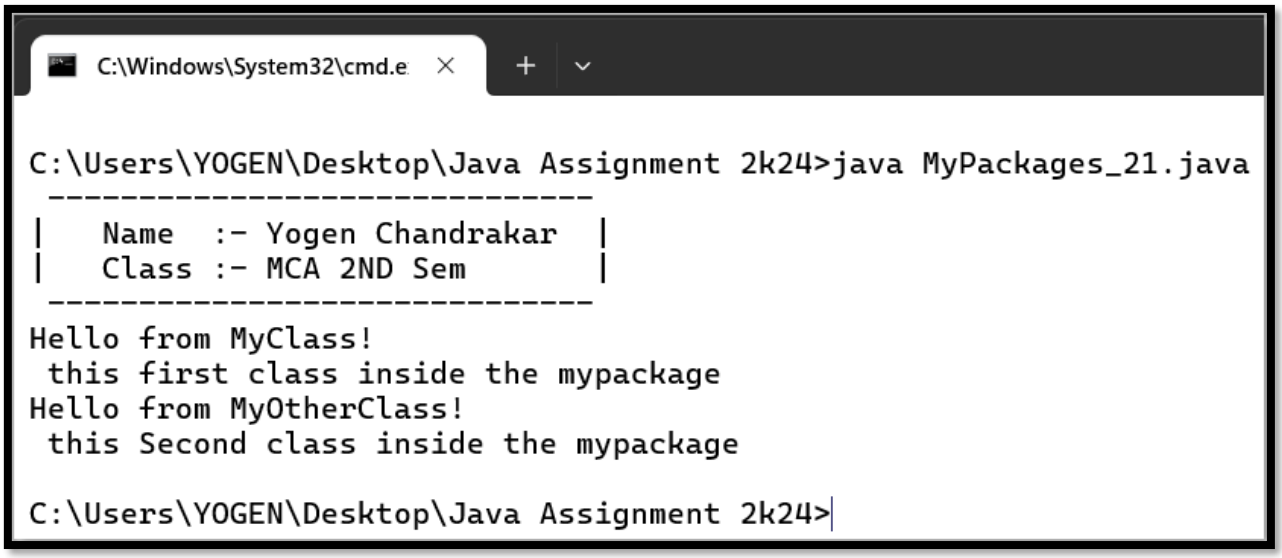
Program No.21

Write a Java program that creates its own package containing two classes.

Code:-

```
package myPackage;
public class MyClass {  // first class
 public void printMessage() {
     System.out.println("Hello from MyClass!\n this first class inside the mypackage");
 }
}
package myPackage;

public class MyOtherClass {  //  second class
 public void printMessage() {
     System.out.println("Hello from MyOtherClass! \n this Second class inside the mypackage");
 }
}
import myPackage.*;   // import package
import myinfo.Myinfo;
public class MyPackages_21 {
 public static void main(String[] args) {
 MyClass obj1 = new MyClass();
 MyOtherClass obj2 = new MyOtherClass();
 Myinfo obj3 = new Myinfo();

 obj3.display();
 obj1.printMessage();
 obj2.printMessage();
 }
}
```

Output :-



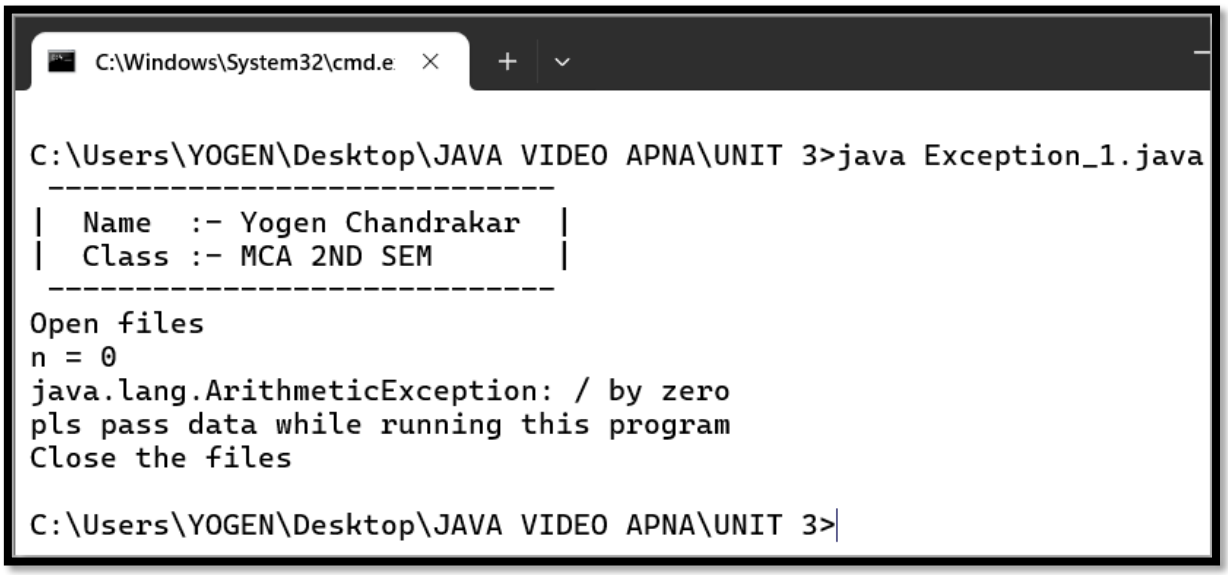
Program No.22

Write a Java program Using Try And Catch Statement.

Code:-

```
class Exception_1{
   public static void main (String args[]){
      System.out.println(" ----------------------------");
      System.out.println("|  Name  :- Yogen Chandrakar  |");
      System.out.println("|  Class :- MCA 2ND SEM       |");
      System.out.println(" ----------------------------");
      try{
         System.out.println("Open files");
         int n = args.length;
         System.out.println("n = "+n);
         int a = 45 /n;
         System.out.println("a = "+a);
         int b[]= {10,20,30};
         b[50] = 100;
      }
      catch(ArithmeticException ae){  // catch block
         System.out.println(ae);           // Display the exception details
         System.out.println("pls pass data while running this program");   // display message to the user
      }
      catch(ArrayIndexOutOfBoundsException aie){       // Display exception details
            // System.out.println(aie); // optional
         aie.printStackTrace();  // print instruction
         // display a message to user
         System.out.println("Pls see that the array index is within the range ");
      }
      finally{
         // close the files
         System.out.println("Close the files ");
      }
   }
}
```

Output :-



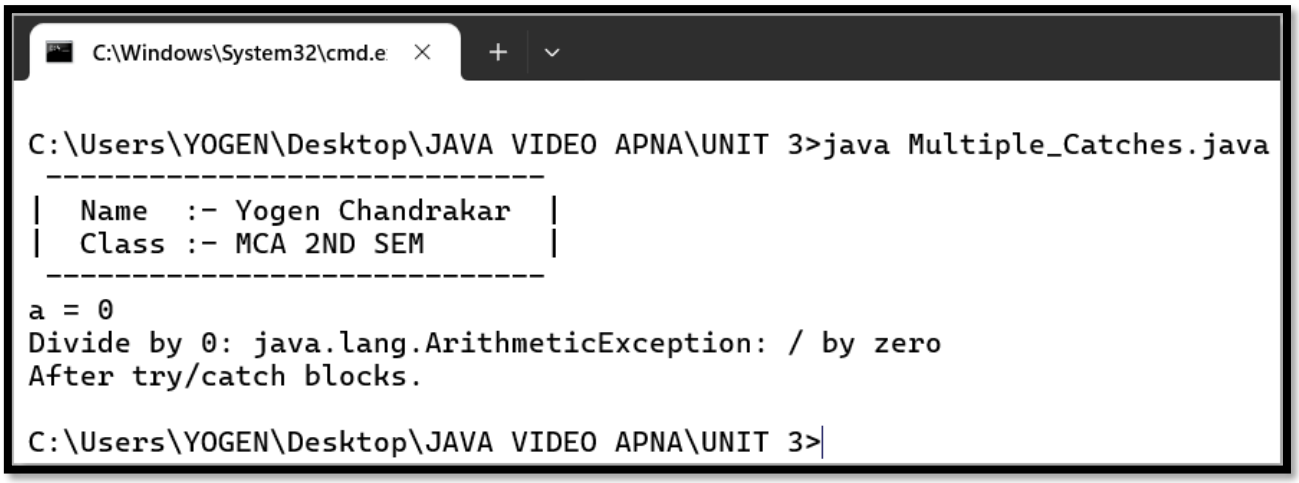
Program No.23

Write a Java program Using Multiple Catch Statements.

Code:-

```
class Multiple_Catches {
 public static void main(String args[]) {
       System.out.println(" ----------------------------");
       System.out.println("|  Name  :- Yogen Chandrakar  |");
       System.out.println("|  Class :- MCA 2ND SEM       |");
       System.out.println(" ----------------------------");
 try {
   int a = args.length;
   System.out.println("a = " + a);
   int b = 42 / a;  // passes no command line arguments
   int c[] = { 1 };
   c[42] = 99;   // passes command line arguments then Index 42 out of bounds for length 1 if args pass
 }
catch(ArithmeticException e) {
   System.out.println("Divide by 0: " + e);
 }
catch(ArrayIndexOutOfBoundsException e) {
   System.out.println("Array index oob: " + e);
 }
 System.out.println("After try/catch blocks.");
  }
```

Output :-



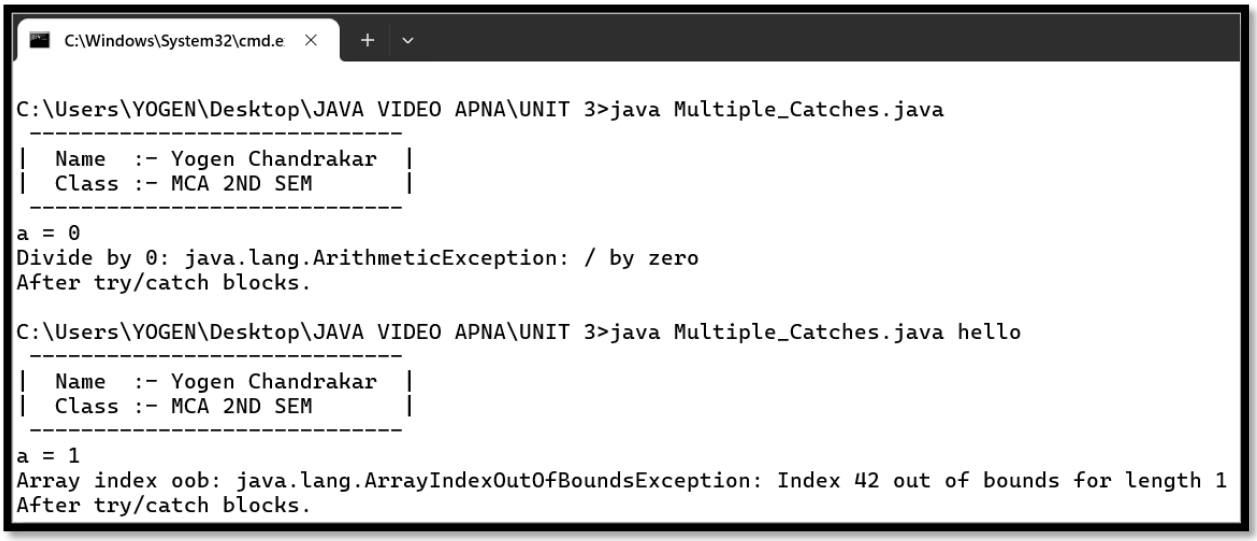
Program No.24

Write a Java program to demonstrate the MultiCatch feature.

Code:-

```
class Multiple_Catches {
 public static void main(String args[]) {
       System.out.println(" ----------------------------");
       System.out.println("|  Name  :- Yogen Chandrakar  |");
       System.out.println("|  Class :- MCA 2ND SEM       |");
       System.out.println(" ----------------------------");
 try {
   int a = args.length;
   System.out.println("a = " + a);
   int b = 42 / a;  // passes no command line arguments
   int c[] = { 1 };
   c[42] = 99;   // passes command line arguments then Index 42
 }
catch(ArithmeticException e) {
   System.out.println("Divide by 0: " + e);
 }
catch(ArrayIndexOutOfBoundsException e) {
   System.out.println("Array index oob: " + e);
 }
 System.out.println("After try/catch blocks.");
  }
}
```

Output :-



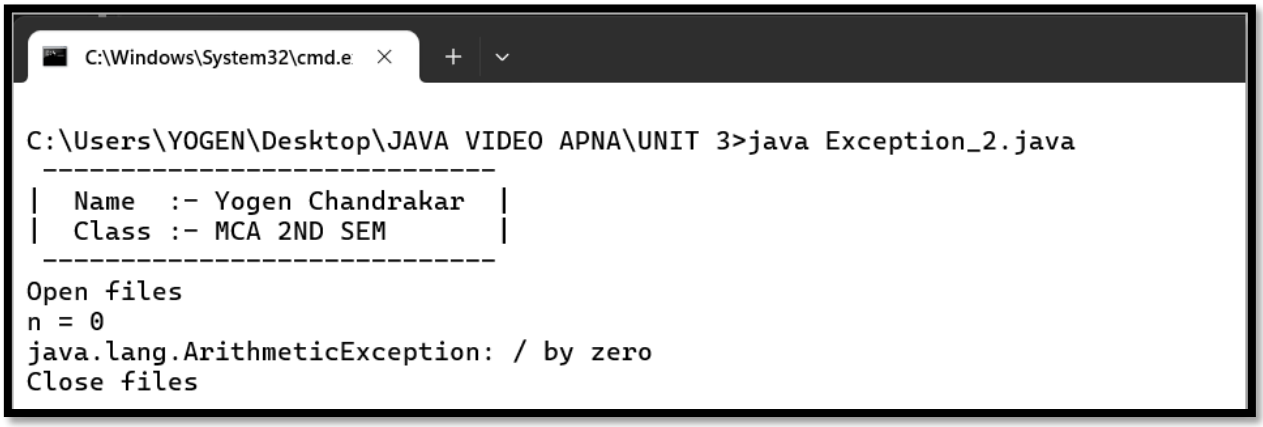
Program No.25

Write a Java program to demonstrate the use of finally block

Code:-

```
class Exception_2{
   public static void main (String args[]){
      System.out.println(" ----------------------------");
      System.out.println("|  Name  :- Yogen Chandrakar  |");
      System.out.println("|  Class :- MCA 2ND SEM       |");
      System.out.println(" ----------------------------");
      try{
         System.out.println("Open files");
         int n = args.length;
         System.out.println("n = "+n);
         int a = 45 /n;     //   by zero
         System.out.println("a = "+a);
         int b[]= {10,20,30};
         b[50] = 100;    //  Index 50 out of bounds for length 3
      }
      catch(ArithmeticException | ArrayIndexOutOfBoundsException ae){
         // Display the exception details
         System.out.println(ae);
         // display message to the user
      }
      finally{  // using finally block
         System.out.println("Close files");
      }
   }
}
```

Output :-



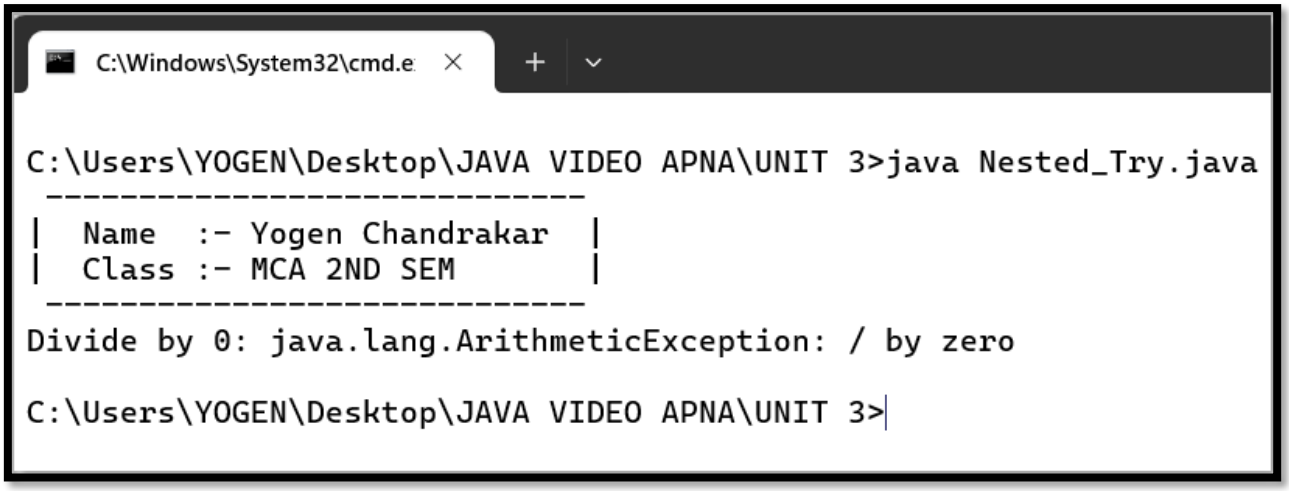
Program No.26

Write a Java program Using Nested Try Statements.

Code:-

```
class Nested_Try {
 public static void main(String args[]){
       System.out.println(" ------------------------------------");
       System.out.println("|  Name  :- Yogen Chandrakar  |");
       System.out.println("|  Class :- MCA 2ND SEM       |");
       System.out.println(" ------------------------------------");
 try {
   int a = args.length;
   int b = 42 / a;
   System.out.println("a = " + a);
   System.out.println("b = " + b);
      try { // nested try block
         if(a==1) a = a/(a-a); // division by zero
         if(a==2) {
             int c[] = { 1 };
             c[42] = 99; // generate an out-of-bounds exception
          }
         }
       catch(ArrayIndexOutOfBoundsException e) {
       System.out.println("Array index out-of-bounds: " + e);
       }
 }
catch(ArithmeticException e) {
     System.out.println("Divide by 0: " + e);
       }
    }
}
```

Output :-



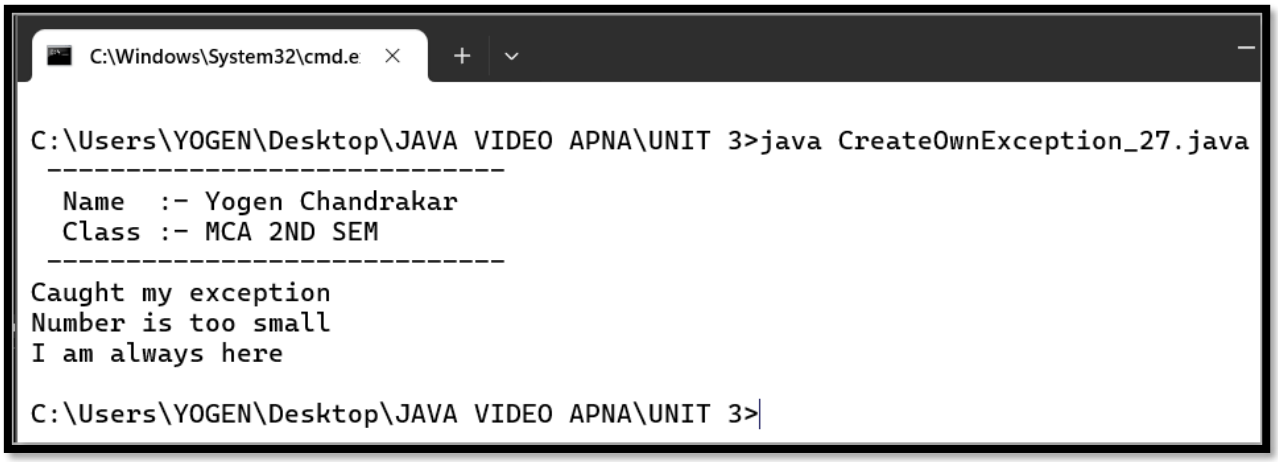
Program No.27

Write a Java program To Create Your Own Exception Class And Display Corresponding Error Message.

Code:-

```
import java.lang.Exception;  //  Exception is a base class

class CreateOwnException_27{
   public static void main(String args[]){
      System.out.println(" ----------------------------");
      System.out.println("  Name  :- Yogen Chandrakar  ");
      System.out.println("  Class :- MCA 2ND SEM       ");
      System.out.println(" ----------------------------");

   int x = 5, y = 1000;
   try{
      float z =  (float) x / (float) y;
       if(z < 0.01){
              throw new MyException("Number is too small");
       }
   }
   catch (MyException e){
      System.out.println("Caught my exception");
       System.out.println(e.getMessage());     // Number is too small
   }
   finally{
      System.out.println ("I am always here");
   }
   }
}
class MyException extends Exception{    // creating user defined exception
   MyException (String message){   // creating constructor
      super (message);
   }
}
```

Output :-



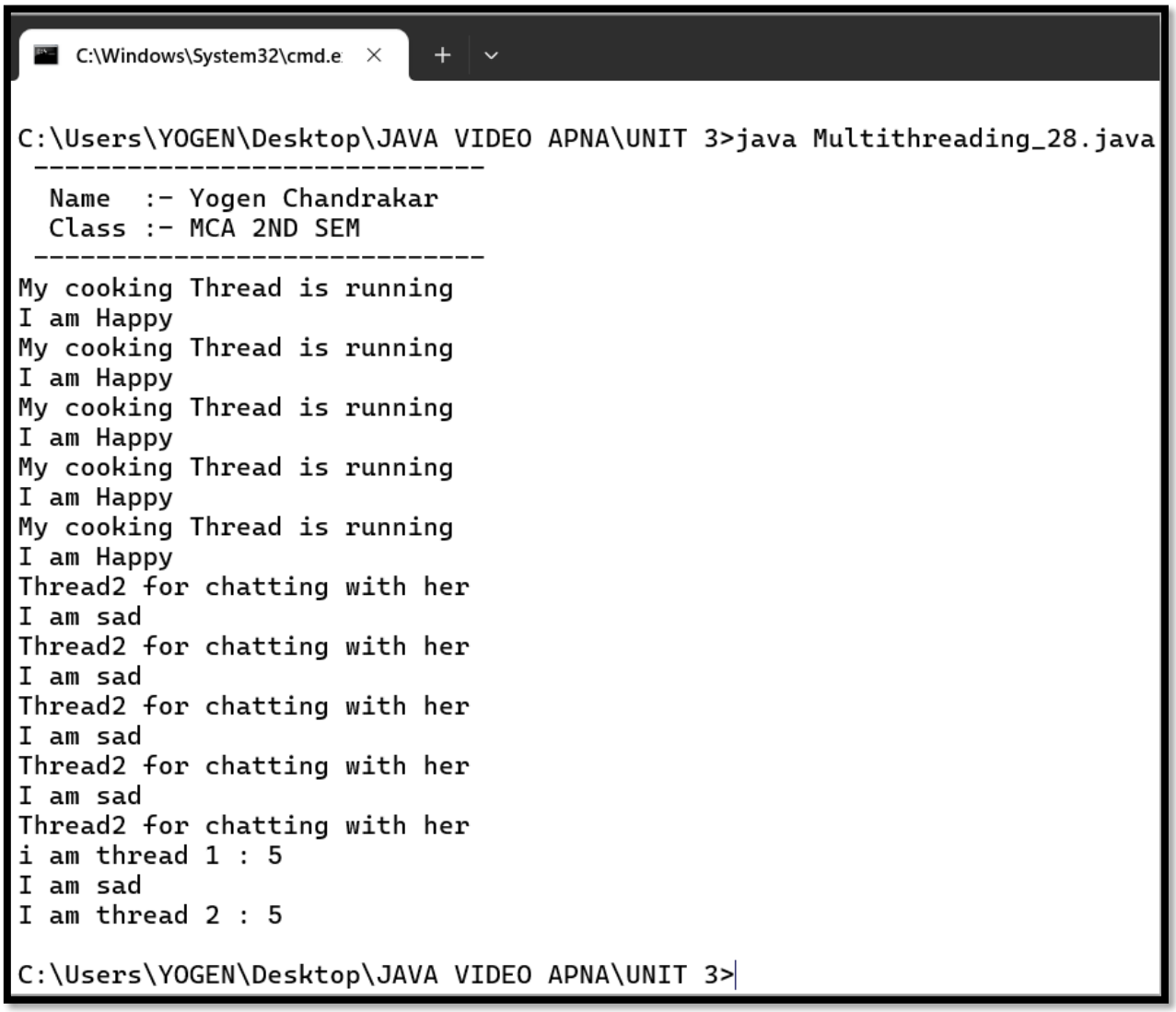
Program No.28

Write a Java program For Creating And Executing Threads by extending the Thread class.

Code:-

```
class Multithreading_28{
   public static void main(String args[]){
      System.out.println(" ----------------------------");
      System.out.println("  Name  :- Yogen Chandrakar  ");
      System.out.println("  Class :- MCA 2ND SEM       ");
      System.out.println(" ----------------------------");

      Mythread1 t1 = new Mythread1();   // Thread 1 goes to New_born state
      Mythread2 t2 = new Mythread2();
      t1.start();  // Now thread t1 goes to runnable state
      t2.start();
   }
}
class Mythread1 extends Thread{
   public void run (){  // Here run() is overriden in child class
      int i = 0;
      while(i<5){
         System.out.println("My cooking Thread is running ");
         System.out.println("I am Happy ");
         i++;
      }
      System.out.println("i am thread 1 : "+i);
   }
   }
class Mythread2 extends Thread{
   public void run (){
      int i = 0;
      while(i<5){
         System.out.println("Thread2 for chatting with her ");
         System.out.println("I am sad ");
         i++;
      }
      System.out.println("I am thread 2 : "+i);
   }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\JAVA VIDEO APNA\UNIT 3>java Multithreading_28.java
Name :- Yogen Chandrakar
Class :- MCA 2ND SEM
My cooking Thread is running
I am Happy
My cooking Thread is running
I am Happy
My cooking Thread is running
I am Happy
My cooking Thread is running
I am Happy
My cooking Thread is running
I am Happy
Thread2 for chatting with her
I am sad
Thread2 for chatting with her
I am sad
Thread2 for chatting with her
I am sad
Thread2 for chatting with her
I am sad
Thread2 for chatting with her
i am thread 1 : 5
I am sad
I am thread 2 : 5
C:\Users\YOGEN\Desktop\JAVA VIDEO APNA\UNIT 3>

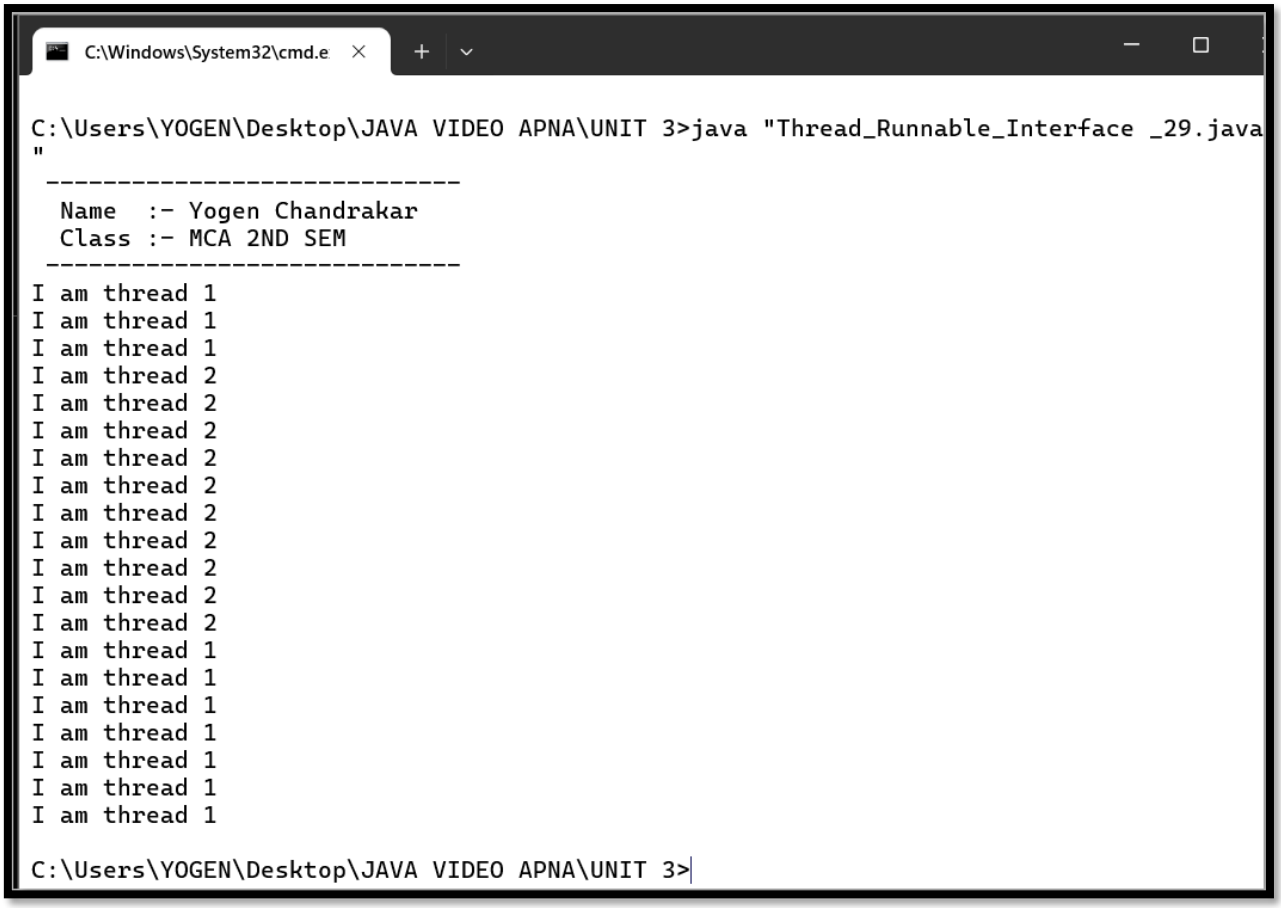
Program No.29

Write a Java program To run Three Threads by implementing the Runnable Interface

Code:-

```
public class Thread_Runnable_Interface{
   public static void main(String args[]){
      System.out.println(" ----------------------------");
      System.out.println("  Name  :- Yogen Chandrakar  ");
      System.out.println("  Class :- MCA 2ND SEM       ");
      System.out.println(" ----------------------------");
      MythreadRunnable_1 bullet1 = new MythreadRunnable_1();
      Thread gun1 = new Thread(bullet1);

      MythreadRunnable_2 bullet2 = new MythreadRunnable_2();
      Thread gun2 = new Thread(bullet2);

      gun1.start();
      gun2.start();
   }
}
class MythreadRunnable_1 implements Runnable{
   public void run(){
      for (int i=0;i<10;i++){
         System.out.println("I am thread 1 ");
      }
   }
}
class MythreadRunnable_2 implements Runnable{
   public void run(){
      for (int i=0;i<10;i++){
         System.out.println("I am thread 2 ");
      }
   }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\JAVA VIDEO APNA\UNIT 3>java "Thread_Runnable_Interface _29.java
"
Name :- Yogen Chandrakar
Class :- MCA 2ND SEM
I am thread 1
I am thread 1
I am thread 1
I am thread 2
I am thread 2
I am thread 2
I am thread 2
I am thread 2
I am thread 2
I am thread 2
I am thread 2
I am thread 2
I am thread 2
I am thread 1
I am thread 1
I am thread 1
I am thread 1
I am thread 1
I am thread 1
I am thread 1
C:\Users\YOGEN\Desktop\JAVA VIDEO APNA\UNIT 3>

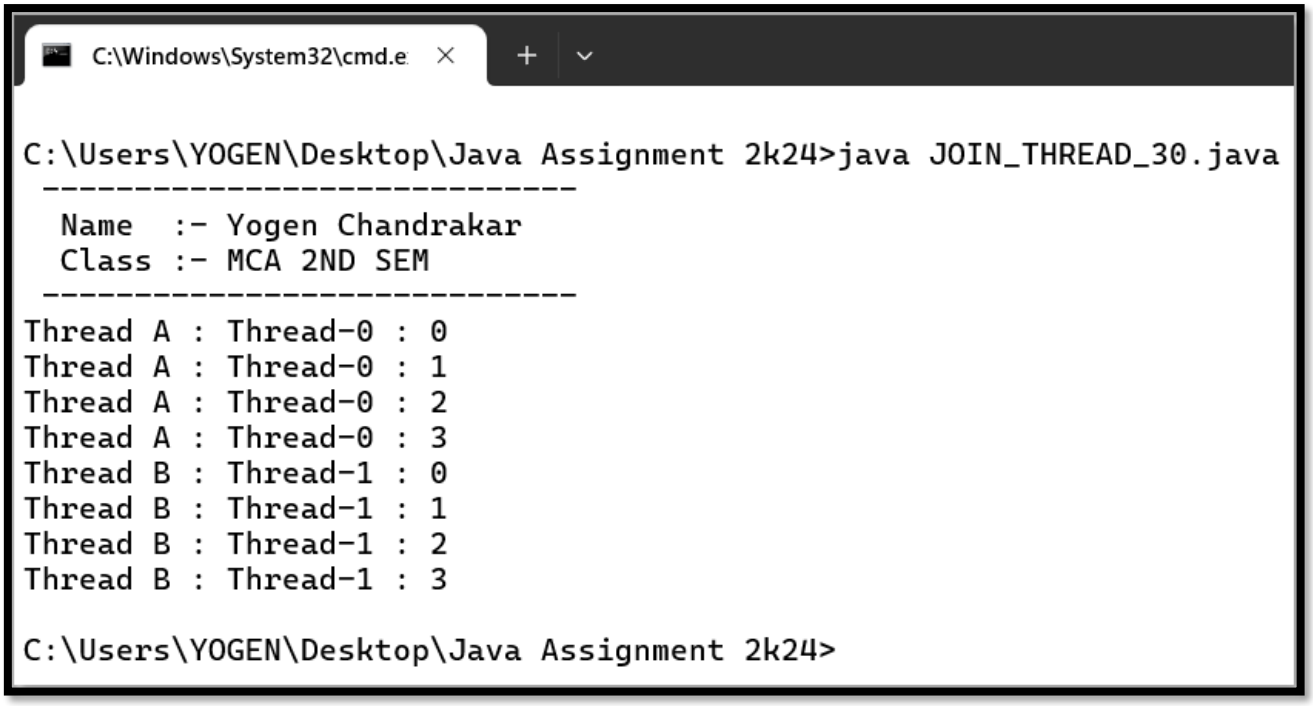
Program No.30

Write a Java program to demonstrate the use of join() method.

Code:-

```
public class JOIN_THREAD_30{
   public static void main(String args[]){
      System.out.println(" ----------------------------");
      System.out.println("  Name  :- Yogen Chandrakar  ");
      System.out.println("  Class :- MCA 2ND SEM       ");
      System.out.println(" ----------------------------");

      ThreadCreate obj = new ThreadCreate("Thread A");
      ThreadCreate obj2 = new ThreadCreate("Thread B");
      obj.start();
      try{
         obj.join();
      }
      catch(Exception e){
         System.out.println(e);
      }
      obj2.start();
      try{
         obj2.join();
      }
      catch(Exception e){
         System.out.println(e);
      }
   }
}
class ThreadCreate extends Thread{
   String name;
   ThreadCreate(String name){
      this.name=name;
   }
   public void run(){
      for(int i=0;i<=3;i++){
        try{
            System.out.println(this.name+":"+Thread.currentThread().getName()+" : "+i);
         }
        catch(Exception e){
            System.out.println(e);
         }
      }
   }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java JOIN_THREAD_30.java
Name :- Yogen Chandrakar
Class :- MCA 2ND SEM
Thread A : Thread-0 : 0
Thread A : Thread-0 : 1
Thread A : Thread-0 : 2
Thread A : Thread-0 : 3
Thread B : Thread-1 : 0
Thread B : Thread-1 : 1
Thread B : Thread-1 : 2
Thread B : Thread-1 : 3
C:\Users\YOGEN\Desktop\Java Assignment 2k24>

Program No.31

Write a Java program to demonstrate Multithreading using wait () & notify().

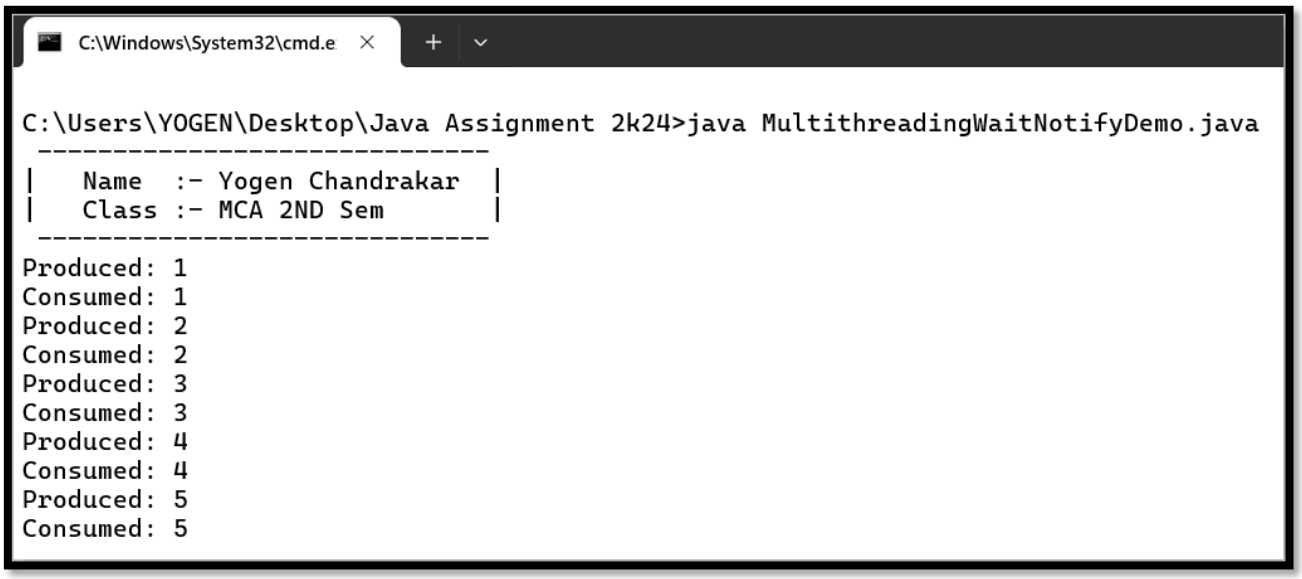
Code:-

```
public class MultithreadingWaitNotifyDemo {
    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        SharedResource resource = new SharedResource();
        Producer producer = new Producer(resource);
        Consumer consumer = new Consumer(resource);

        producer.start();
        consumer.start();
    }
}
class SharedResource {
    private int data;
    private boolean dataAvailable = false;

    public synchronized void produce(int value) {
        while (dataAvailable) {
            try {
                wait();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
                System.out.println("Producer interrupted");
            }
        }
        data = value;
        dataAvailable = true;
        System.out.println("Produced: " + value);
        notify();
    }

    public synchronized int consume() {
        while (!dataAvailable) {
            try {
                wait();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
                System.out.println("Consumer interrupted");
            }
```

```
            }
            dataAvailable = false;
            System.out.println("Consumed: " + data);
            notify();
            return data;
        }
    }

    class Producer extends Thread {
        private final SharedResource resource;

        public Producer(SharedResource resource) {
            this.resource = resource;
        }

        @Override
        public void run() {
            for (int i = 1; i <= 5; i++) {
                resource.produce(i);
                try {
                    Thread.sleep(100); // Simulate time taken to produce an item
                } catch (InterruptedException e) {
                    Thread.currentThread().interrupt();
                    System.out.println("Producer thread interrupted");
                }
            }
        }
    }

    class Consumer extends Thread {
        private final SharedResource resource;

        public Consumer(SharedResource resource) {
            this.resource = resource;
        }

        @Override
        public void run() {
            for (int i = 1; i <= 5; i++) {
                resource.consume();
                try {
                    Thread.sleep(150); // Simulate time taken to consume an item
                } catch (InterruptedException e) {
                    Thread.currentThread().interrupt();
                    System.out.println("Consumer thread interrupted");
```

```
        }
      }
    }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java MultithreadingWaitNotifyDemo.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
Produced: 1
Consumed: 1
Produced: 2
Consumed: 2
Produced: 3
Consumed: 3
Produced: 4
Consumed: 4
Produced: 5
Consumed: 5

Program No.32

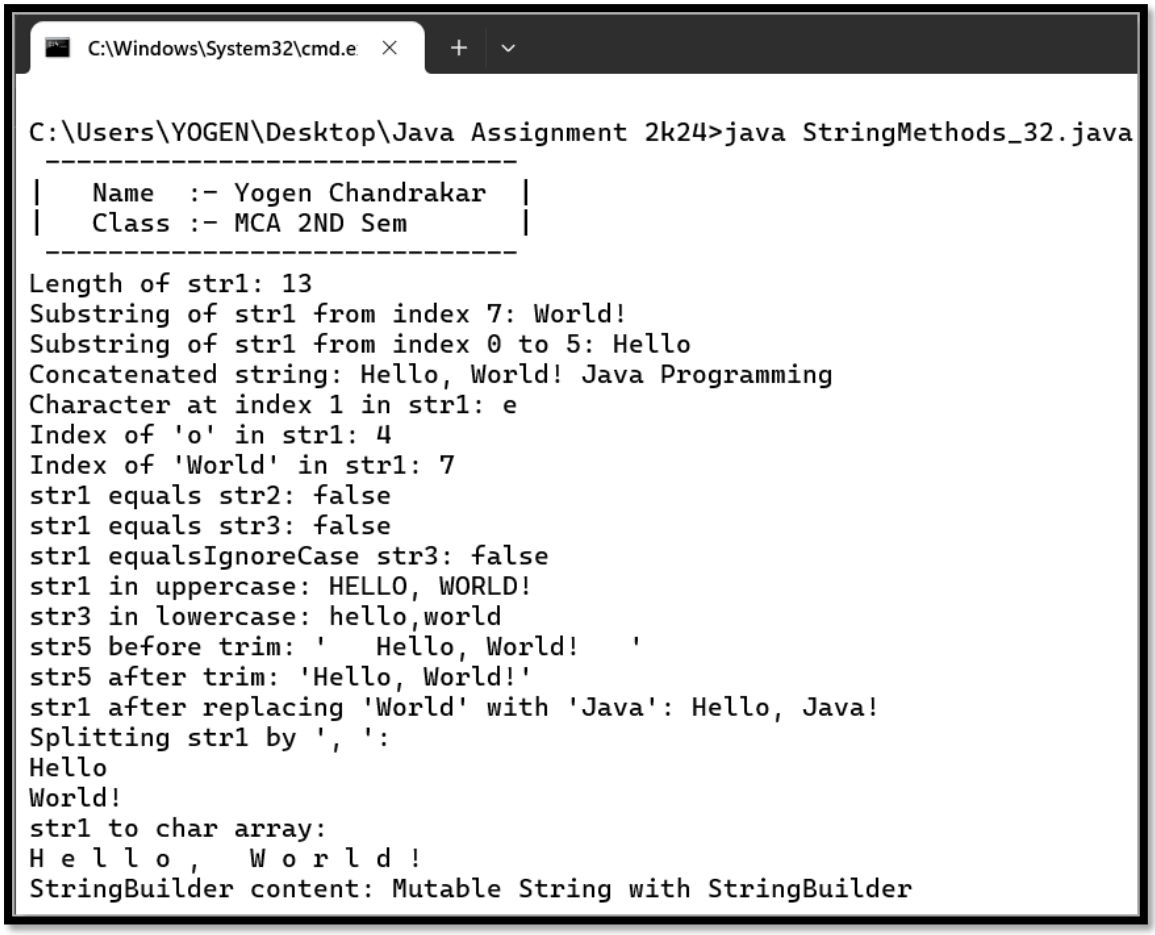
Write a Java program to demonstrate The String Class & its methods.

Code:-

```
import myinfo.Myinfo;
public class StringMethods_32 {
   public static void main(String[] args) {
      Myinfo m = new Myinfo();
      m.display();
      // Creating strings
      String str1 = "Hello, World!";
      String str2 = new String("Java Programming");
      char [] ch  ={'H','E','L','L','O',',','W','O','R','L','D'};
      String str3 = new String(ch);  // using array

      // Length of a string
      System.out.println("Length of str1: " + str1.length());  //13

      // Substring
      System.out.println("Substring of str1 from index 7: " + str1.substring(7));  //World!
      System.out.println("Substring of str1 from index 0 to 5: " + str1.substring(0, 5));  //Hello

      String str4 = str1.concat(" ").concat(str2);        // Concatenation
      System.out.println("Concatenated string: " + str4);  // Hello, World! Java Programming

      // Character at a specific index
      System.out.println("Character at index 1 in str1: " + str1.charAt(1));   // e

      // Index of a character or substring
      System.out.println("Index of 'o' in str1: " + str1.indexOf('o'));   //  4
      System.out.println("Index of 'World' in str1: " + str1.indexOf("World"));// 7

      // Comparison
      System.out.println("str1 equals str2: " + str1.equals(str2)); // false
      System.out.println("str1 equals str3: " + str1.equals(str3)); // false
      System.out.println("str1 equalsIgnoreCase str3: " + str1.equalsIgnoreCase(str3)); // true

      // Case conversion
      System.out.println("str1 in uppercase: " + str1.toUpperCase()); // HELLO, WORLD!
      System.out.println("str3 in lowercase: " + str3.toLowerCase());  // hello, world!

      String str5 = "   Hello, World!   ";  // Trim
      System.out.println("str5 before trim: '" + str5 + "'"); // '   Hello, World!   '
      System.out.println("str5 after trim: '" + str5.trim() + "'");  // 'Hello, World!'
```

```
        // Replace
        System.out.println("str1 after replacing 'World' with 'Java': " + str1.replace("World", "Java"));
                // Hello, Java!
        String[] words = str1.split(", ");          // Split
        System.out.println("Splitting str1 by ', ':");
        for (String word : words) {
           System.out.println(word);
        }
        // String to char array
        char[] charArray = str1.toCharArray();
        System.out.println("str1 to char array:");
        for (char c : charArray) {
           System.out.print(c + " ");   // H e l l o ,  W o r l d !
        }
        System.out.println();

        // StringBuilder for mutable strings
        StringBuilder sb = new StringBuilder("Mutable String");
        sb.append(" with StringBuilder");
        System.out.println("StringBuilder content: " + sb.toString()); // Mutable String with StringBuilder
    }
}
```

Output :-



Program No.33

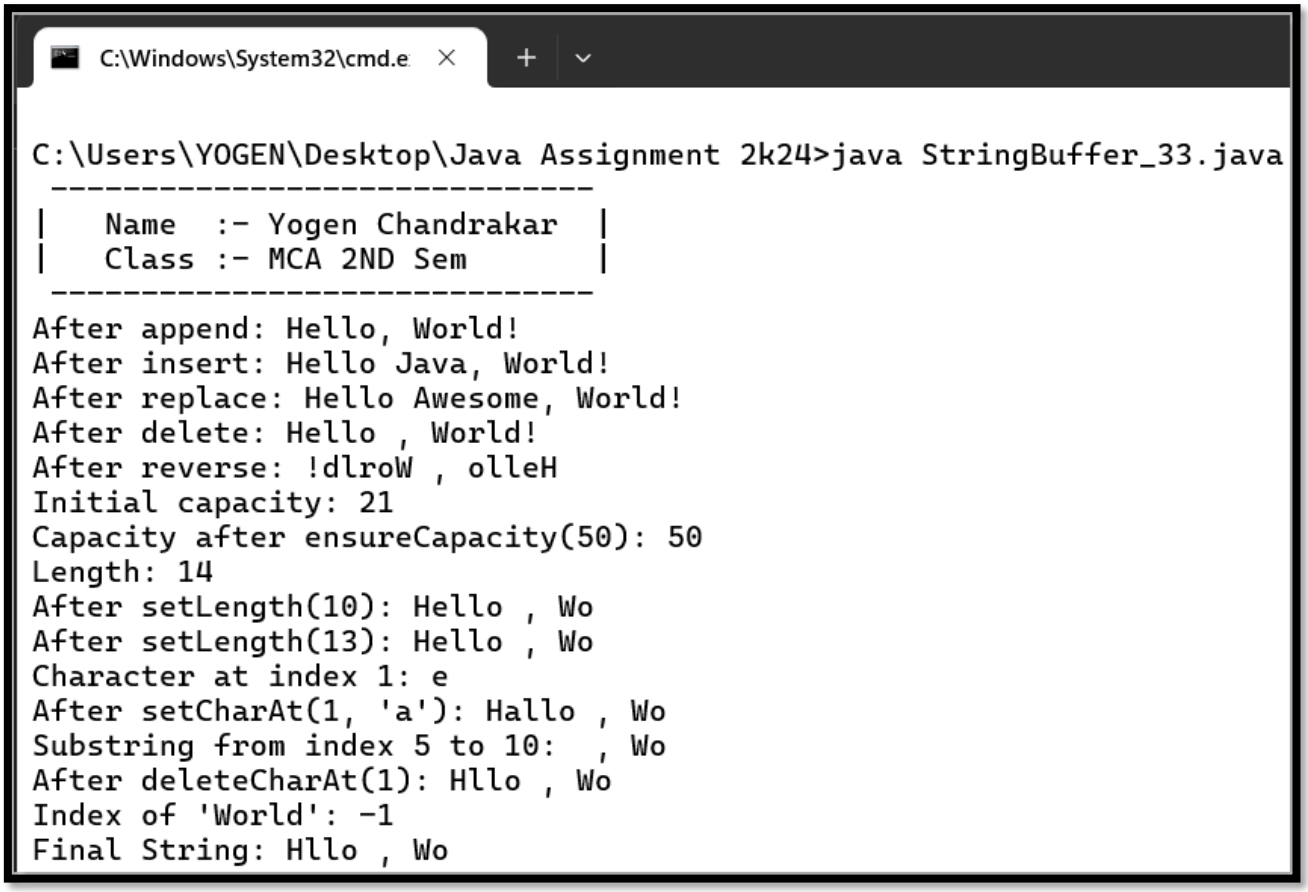
Write a Java program to demonstrate StringBuffer Class & its methods.

Code:-

```
import myinfo.Myinfo;
public class StringBuffer_33{
   public static void main(String[] args) {
      Myinfo m = new Myinfo();
      m.display();
      // Creating StringBuffer instances
      StringBuffer sb1 = new StringBuffer("Hello");
      StringBuffer sb2 = new StringBuffer("World");

      // Append
      sb1.append(", ");
      sb1.append(sb2);
      sb1.append("!");
      System.out.println("After append: " + sb1);

      // Insert
      sb1.insert(5, " Java");
      System.out.println("After insert: " + sb1);

      sb1.replace(6, 10, "Awesome");  // Replace
      System.out.println("After replace: " + sb1);

      sb1.delete(6, 13);  // Delete
      System.out.println("After delete: " + sb1);

      sb1.reverse();  // Reverse
      System.out.println("After reverse: " + sb1);
      sb1.reverse(); // reversing back to original for further operations

      // Capacity and ensureCapacity
      System.out.println("Initial capacity: " + sb1.capacity());
      sb1.ensureCapacity(50);
      System.out.println("Capacity after ensureCapacity(50): " + sb1.capacity());

      // Length and setLength
      System.out.println("Length: " + sb1.length());
      sb1.setLength(10);
      System.out.println("After setLength(10): " + sb1);
      sb1.setLength(13); // resetting length for further operations
      System.out.println("After setLength(13): " + sb1);
```

```
        // charAt and setCharAt
        System.out.println("Character at index 1: " + sb1.charAt(1));
        sb1.setCharAt(1, 'a');
        System.out.println("After setCharAt(1, 'a'): " + sb1);

        // Substring
        String sub = sb1.substring(5, 10);
        System.out.println("Substring from index 5 to 10: " + sub);

        // DeleteCharAt
        sb1.deleteCharAt(1);
        System.out.println("After deleteCharAt(1): " + sb1);

        // Index of
        int index = sb1.indexOf("World");
        System.out.println("Index of 'World': " + index);

        // Converting StringBuffer to String
        String finalString = sb1.toString();
        System.out.println("Final String: " + finalString);
    }
}
```

Output :-



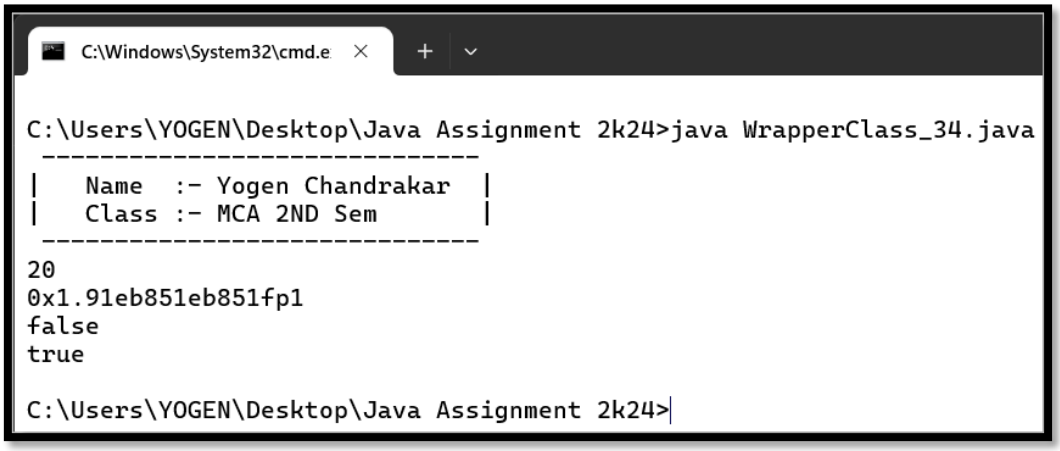
Program No.34

Write a Java program to demonstrate various Wrapper Classes.

Code:-

```
import myinfo.Myinfo;
public class WrapperClass_34 {
 public static void main(String[] args) {
   Myinfo m = new Myinfo();
   m.display();
   // Wrapper classes for primitive types
   Integer i = 10;
   Double d = 3.14;
   Boolean b = true;
   Character c = 'a';
   // Autoboxing
   int j = i;
   double e = d;
   boolean bool = b;
   char ch = c;
   // Wrapper classes for conversions
   Integer intObj = Integer.valueOf("100");
   Double doubleObj = Double.valueOf("3.14");
   Boolean boolObj = Boolean.valueOf("true");
   Character charObj = Character.valueOf('a');
   // Parsing methods
   int a = Integer.parseInt("100");
   double f = Double.parseDouble("3.14");
   boolean bool1 = Boolean.parseBoolean("true");
   char ch1 = "a".charAt(0);
   // Other methods
   System.out.println(Integer.max(10, 20));
   System.out.println(Double.toHexString(3.14));
   System.out.println(Boolean.logicalAnd(true, false));
   System.out.println(Character.isLetter('a'));
   }}
```

Output :-



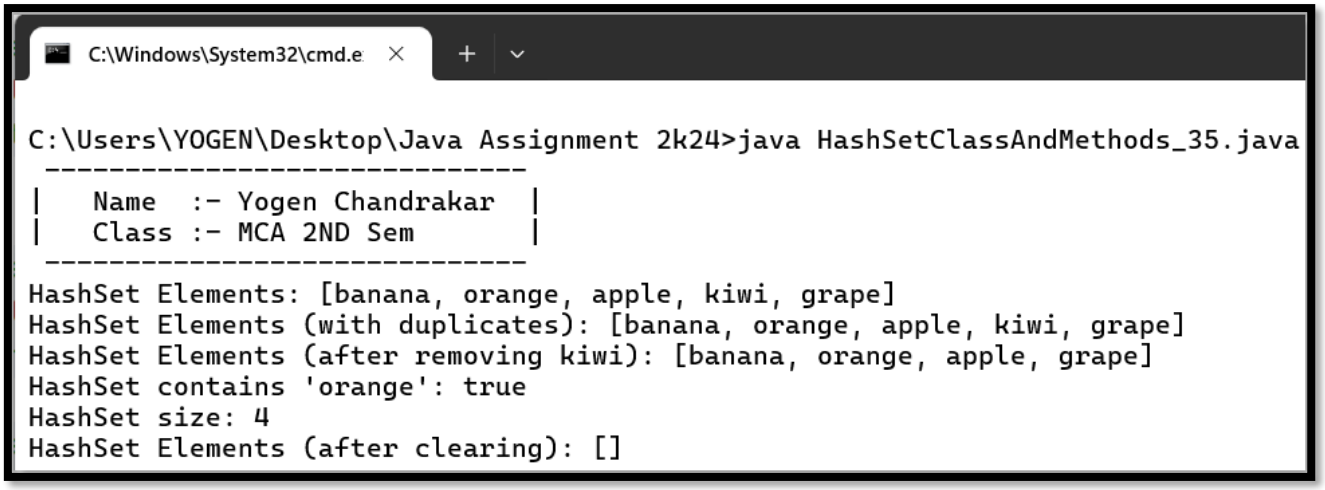
Program No.35

Write a Java program to demonstrate HashSet Class & its methods.

Code:-

```
import java.util.HashSet;
import myinfo.Myinfo;
public class HashSetClassAndMethods_35 {
 public static void main(String[] args) {
    Myinfo m = new Myinfo();
    m.display();
    // Creating a HashSet
    HashSet<String> set = new HashSet<>();
    // Adding elements to the HashSet
    set.add("apple");
    set.add("banana");
    set.add("orange");
    set.add("kiwi");
    set.add("grape");
    // Displaying the HashSet elements
    System.out.println("HashSet Elements: " + set);
    // Adding duplicate elements to the HashSet
    set.add("apple");
    set.add("banana");
    // Displaying the HashSet elements after adding duplicates
    System.out.println("HashSet Elements (with duplicates): " + set);
    // Removing an element from the HashSet
    set.remove("kiwi");
    // Displaying the HashSet elements after removing an element
    System.out.println("HashSet Elements (after removing kiwi): " + set);
    // Checking if an element is present in the HashSet
    boolean containsOrange = set.contains("orange");
    System.out.println("HashSet contains 'orange': " + containsOrange);
    // Checking the size of the HashSet
    int size = set.size();
    System.out.println("HashSet size: " + size);
    // Clearing the HashSet
    set.clear();
    System.out.println("HashSet Elements (after clearing): " + set);
    }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java HashSetClassAndMethods_35.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
HashSet Elements: [banana, orange, apple, kiwi, grape]
HashSet Elements (with duplicates): [banana, orange, apple, kiwi, grape]
HashSet Elements (after removing kiwi): [banana, orange, apple, grape]
HashSet contains 'orange': true
HashSet size: 4
HashSet Elements (after clearing): []

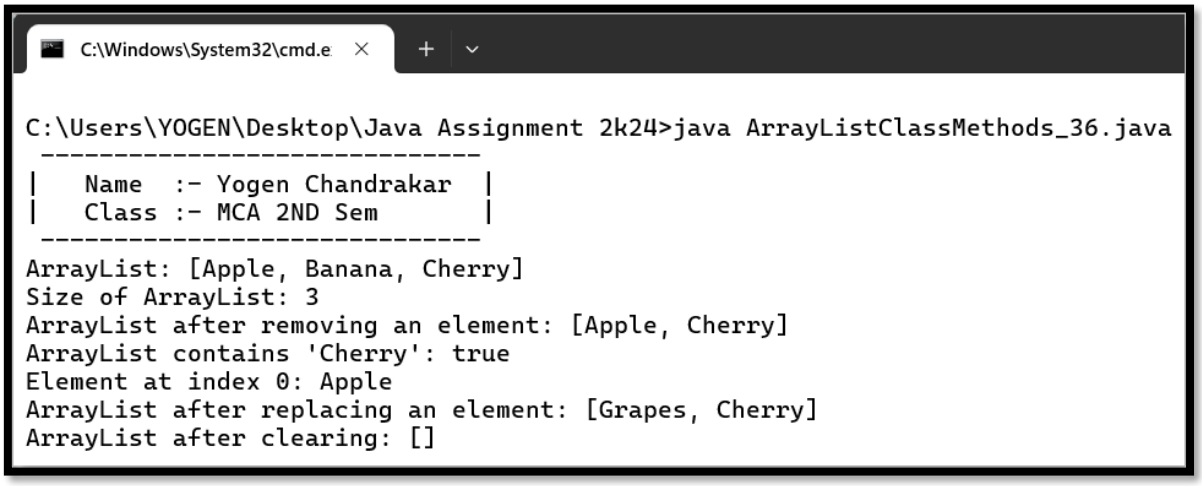
Program No.36

Write a Java program to demonstrate ArrayList Class & its methods.

Code:-

```
import java.util.ArrayList;
import myinfo.Myinfo;
public class ArrayListClassMethods_36 {
 public static void main(String[] args) {
   Myinfo m = new Myinfo();
   m.display();
   // Creating an ArrayList
   ArrayList<String> list = new ArrayList<>();
   // Adding elements to the ArrayList
   list.add("Apple");
   list.add("Banana");
   list.add("Cherry");
   // Displaying the ArrayList elements
   System.out.println("ArrayList: " + list);
   System.out.println("Size of ArrayList: " + list.size());      // Getting the size of the ArrayList
   list.remove("Banana");              // Removing an element from the ArrayList
   System.out.println("ArrayList after removing an element: " + list);
   // Checking if the ArrayList contains an element
   System.out.println("ArrayList contains 'Cherry': " + list.contains("Cherry"));
   // Getting an element from the ArrayList
   System.out.println("Element at index 0: " + list.get(0));
   // Replacing an element in the ArrayList
   list.set(0, "Grapes");
   System.out.println("ArrayList after replacing an element: " + list);
   // Clearing the ArrayList
   list.clear();
   System.out.println("ArrayList after clearing: " + list);
   }
}
```

Output :-



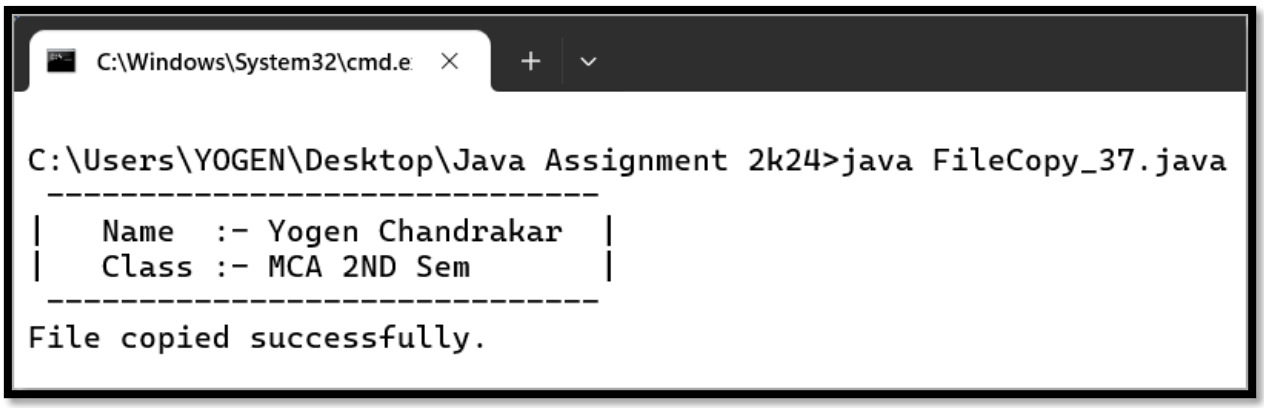
Program No.37

Write a Java program to copy a File.

Code:-

```
import java.io.*;
import myinfo.Myinfo;
public class FileCopy_37 {
 public static void main(String[] args) {
    Myinfo m = new Myinfo();
    m.display();
    try {
       // create file input stream
       FileInputStream in = new FileInputStream("input.txt");
       // create file output stream
       FileOutputStream out = new FileOutputStream("output.txt");
       // read data from input file and write to output file
       int data;
       while ((data = in.read()) != -1) {
       out.write(data);
       }
       // close streams
       in.close();
       out.close();
       System.out.println("File copied successfully.");
    }
    catch (IOException e) {
       System.out.println("An error occurred.");
       e.printStackTrace();
       }
    }
}
```

Output :-



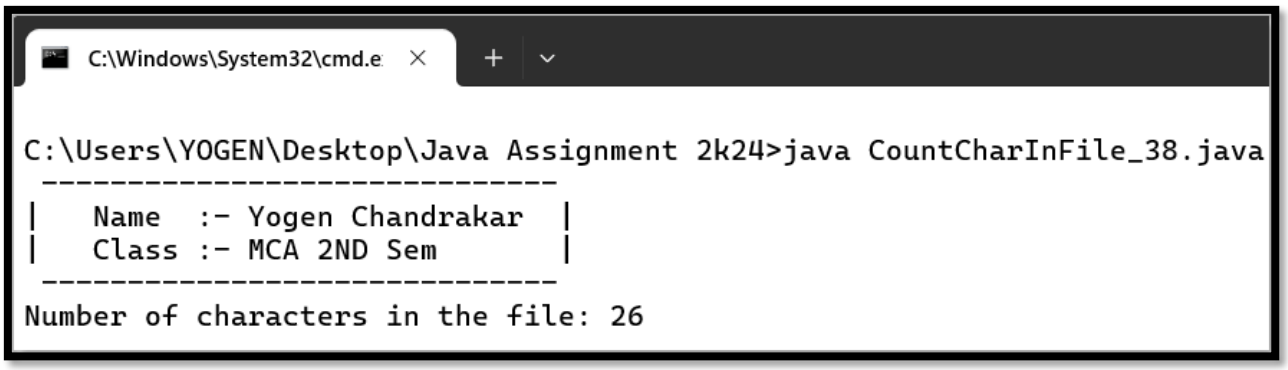
Program No.38

Write a Java program to Count the numbers of Characters in a File.

Code:-

```
import java.io.FileReader;
import java.io.IOException;
import myinfo.Myinfo;
public class CountCharInFile_38 {
 public static void main(String[] args) {
   Myinfo m = new Myinfo();
   m.display();
   try {
      FileReader reader = new FileReader("output.txt");
      int character;
      int count = 0;
      while ((character = reader.read()) != -1) {
      count++;
      }
      System.out.println("Number of characters in the file: " + count);
      reader.close();
   }
   catch (IOException e) {
      e.printStackTrace();
    }
   }
}
```

Output :-



Program No.39

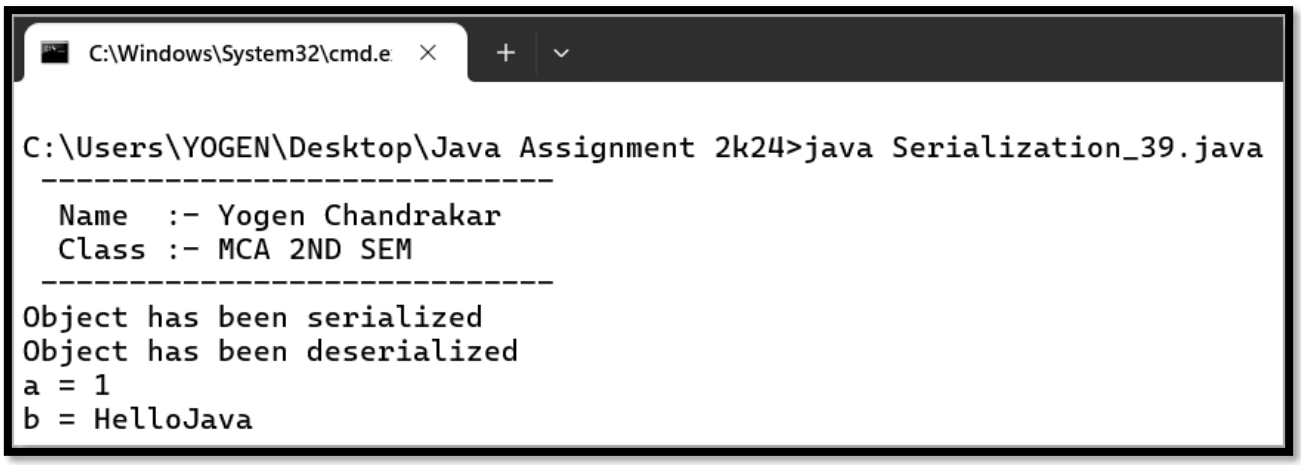
Write a Java program to demonstrate Object Serialization.

Code:-

```
// Java code for serialization of a Java object
import java.io.*;
class Serialization_39{
   public static void main(String[] args){
      System.out.println("  Name  :- Yogen Chandrakar  ");
      System.out.println("  Class :- MCA 2ND SEM       ");
      Demo object = new Demo(1, "HelloJava");
      String filename = "file.ser";

      // Serialization
      try{
         //Saving of object in a file
         FileOutputStream file = new FileOutputStream(filename);
         ObjectOutputStream out = new ObjectOutputStream(file);

         // Method for serialization of object
         out.writeObject(object);

         out.close();
         file.close();
         System.out.println("Object has been serialized");
      }
      catch(IOException ex){
         System.out.println("IOException is caught");
      }
      Demo object1 = null;

      // Deserialization
      try{
         // Reading the object from a file
         FileInputStream file = new FileInputStream(filename);
         ObjectInputStream in = new ObjectInputStream(file);

         // Method for deserialization of object
         object1 = (Demo)in.readObject();

         in.close();
         file.close();

         System.out.println("Object has been deserialized ");
         System.out.println("a = " + object1.a);
```

```
            System.out.println("b = " + object1.b);
         }

         catch(IOException ex){
            System.out.println("IOException is caught");
         }
         catch(ClassNotFoundException ex){
            System.out.println("ClassNotFoundException is caught");
         }
      }
   }
   class Demo implements java.io.Serializable{
      public int a;
      public String b;

      // Default constructor
      public Demo(int a, String b){
         this.a = a;
         this.b = b;
      }
   }
```

Output :-



Program No.40.

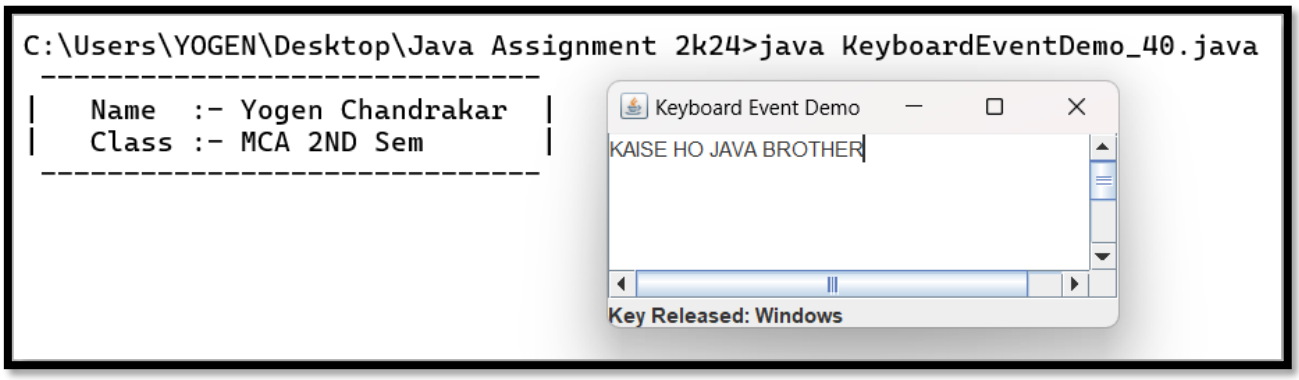
Write a Java program to demonstrate Keyboard Event.

Code:-

```
import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingUtilities;

public class KeyboardEventDemo_41 extends JFrame implements KeyListener {

    private JTextArea textArea;
    private JLabel infoLabel;

    public KeyboardEventDemo_40() {
        // Create a JFrame with a title
        super("Keyboard Event Demo");

        // Create a JTextArea and add a KeyListener to it
        textArea = new JTextArea(10, 30);
        textArea.addKeyListener(this);

        // Create a JLabel to display information
        infoLabel = new JLabel("Type in the text area to see keyboard
                                events.");

        // Add the text area and label to the frame
        add(new JScrollPane(textArea), "Center");
        add(infoLabel, "South");

        // Set the frame's default close operation and size
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 300);
        setVisible(true);
        }

       // Implementing the KeyListener methods
       public void keyTyped(KeyEvent e) {
        infoLabel.setText("Key Typed: " + e.getKeyChar());
        }
```

```
            public void keyPressed(KeyEvent e) {
                infoLabel.setText("Key Pressed: " + KeyEvent.getKeyText(e.getKeyCode()));
            }

            @Override
            public void keyReleased(KeyEvent e) {
                infoLabel.setText("Key Released: " + KeyEvent.getKeyText(e.getKeyCode()));
            }

        public static void main(String[] args) {
            System.out.println(" ------------------------------");
            System.out.println("|   Name  :- Yogen Chandrakar  |" );
            System.out.println("|   Class :- MCA 2ND Sem       |" );
            System.out.println(" ------------------------------");
            // Run the program
            SwingUtilities.invokeLater(() -> new KeyboardEventDemo_40());
        }
    }
```

Output :-



Program No.41

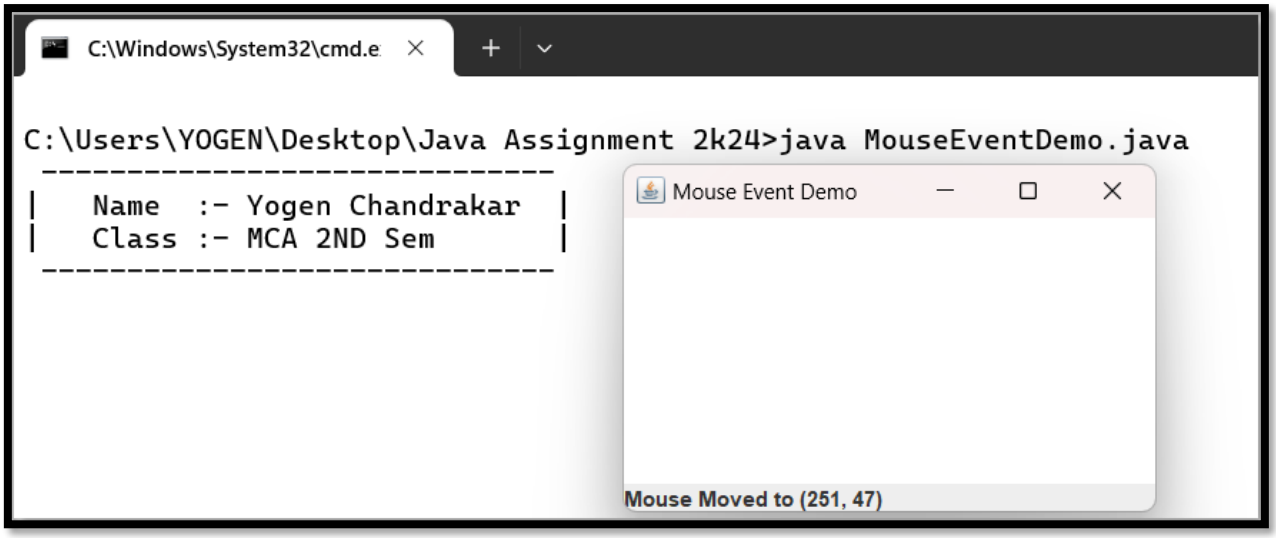
Write a Java program to demonstrate Mouse Event.

Code:-

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

public class MouseEventDemo extends JFrame implements MouseListener, MouseMotionListener {

    private JLabel infoLabel;
    private JPanel panel;

    public MouseEventDemo() {
        // Create a JFrame with a title
        super("Mouse Event Demo");

        // Create a JPanel and add MouseListener and MouseMotionListener to it
        panel = new JPanel();
        panel.setBackground(Color.WHITE);
        panel.addMouseListener(this);
        panel.addMouseMotionListener(this);

        // Create a JLabel to display information
        infoLabel = new JLabel("Interact with the panel using the mouse.");

        // Add the panel and label to the frame
        add(panel, BorderLayout.CENTER);
        add(infoLabel, BorderLayout.SOUTH);

        // Set the frame's default close operation and size
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 300);
        setVisible(true);
    }

    // Implementing the MouseListener methods
    @Override
```

```
    public void mouseClicked(MouseEvent e) {
        infoLabel.setText("Mouse Clicked at (" + e.getX() + ", " + e.getY() + ")");
    }

    @Override
    public void mousePressed(MouseEvent e) {
        infoLabel.setText("Mouse Pressed at (" + e.getX() + ", " + e.getY() + ")");
    }

    @Override
    public void mouseReleased(MouseEvent e) {
        infoLabel.setText("Mouse Released at (" + e.getX() + ", " + e.getY() + ")");
    }

    @Override
    public void mouseEntered(MouseEvent e) {
        infoLabel.setText("Mouse Entered the panel");
    }

    @Override
    public void mouseExited(MouseEvent e) {
        infoLabel.setText("Mouse Exited the panel");
    }

    // Implementing the MouseMotionListener methods
    @Override
    public void mouseDragged(MouseEvent e) {
        infoLabel.setText("Mouse Dragged to (" + e.getX() + ", " + e.getY() + ")");
    }

    @Override
    public void mouseMoved(MouseEvent e) {
        infoLabel.setText("Mouse Moved to (" + e.getX() + ", " + e.getY() + ")");
    }

    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        // Run the program
        SwingUtilities.invokeLater(() -> new MouseEventDemo());
    }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java MouseEventDemo.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
Mouse Event Demo
Mouse Moved to (251, 47)

Program No.42

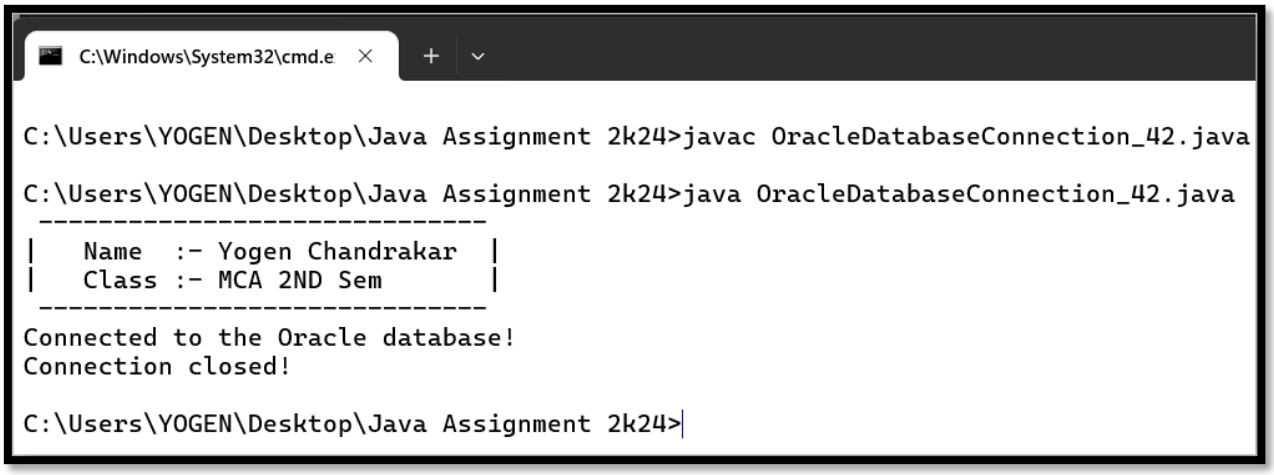
Write a Java program to establish connection to the database.

Code:-

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class OracleDatabaseConnection_42{
   // JDBC URL, username and password of Oracle server
   private static final String URL = "jdbc:oracle:thin:@localhost:1521:xe";
   private static final String USER = "YOGEN03";
   private static final String PASSWORD = "7410";

   // JDBC variables for opening and managing connection
   private static Connection connection;
   public static void main(String[] args) {
      System.out.println("|   Name  :- Yogen Chandrakar  |" );
      System.out.println("|   Class :- MCA 2ND Sem       |" );

      try {
         // Initialize the connection
         connection = DriverManager.getConnection(URL, USER, PASSWORD);

         if (connection != null) {
            System.out.println("Connected to the Oracle database!");
         } else {
            System.out.println("Failed to make connection!");
         }
      } catch (SQLException e) {
         // Print SQL exception information
         System.out.println("SQL State: " + e.getSQLState());
         System.out.println("Error Code: " + e.getErrorCode());
         System.out.println("Message: " + e.getMessage());
         e.printStackTrace();
      } finally {
         // Close the connection if it was established
         try {
            if (connection != null) {
               connection.close();
               System.out.println("Connection closed!");
            }
         } catch (SQLException ex) {
            ex.printStackTrace();
         }
```

```
        }
      }
    }
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>javac OracleDatabaseConnection_42.java
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java OracleDatabaseConnection_42.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
Connected to the Oracle database!
Connection closed!
C:\Users\YOGEN\Desktop\Java Assignment 2k24>

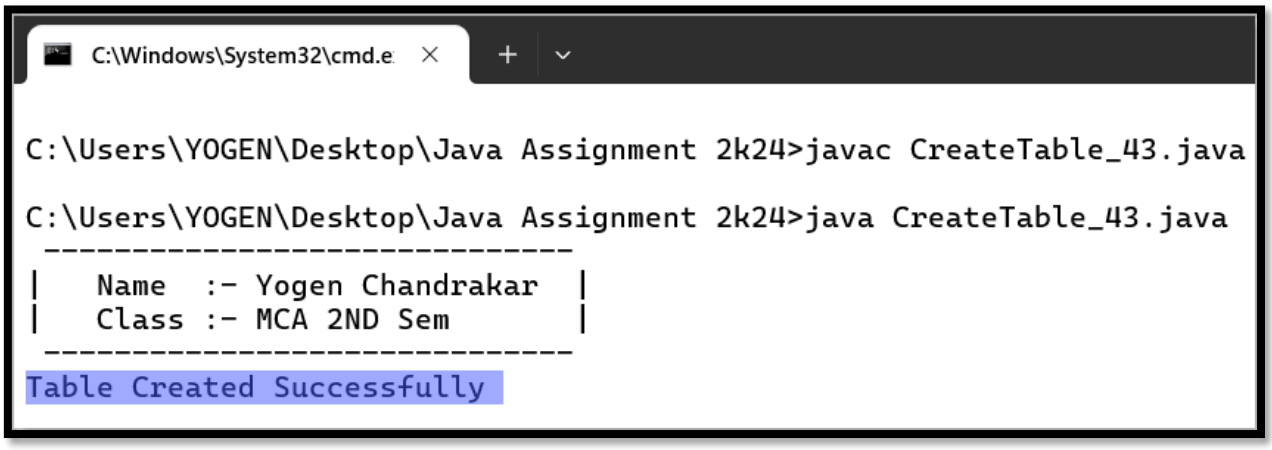
Program No.43

Write a Java program to create a table named employee with fields as emp_id, emp_name, age , dept.

Code:-

```
//package javaapplication3;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class CreateTable_43 {
   public static void main(String[] args) throws SQLException {
          System.out.println("|   Name  :- Yogen Chandrakar  |" );
          System.out.println("|   Class :- MCA 2ND Sem       |" );

      Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"YOGEN03", "7410");
      try {
         // Establish connection to the Oracle databas
         // Create a statement
         Statement stmt = conn.createStatement();
         String sql = "CREATE TABLE employee (" + "id INT PRIMARY KEY," + "name
VARCHAR(50)," + "age INT," + "dept VARCHAR(50))";
         stmt.executeUpdate(sql);
         System.out.println("Table Created Successfully ");

         stmt.close();
         conn.close();
      } catch (SQLException e) {
         System.out.println("Table retrieve failed. Error: " + e.getMessage());
          }
   }
}
```

Output :-



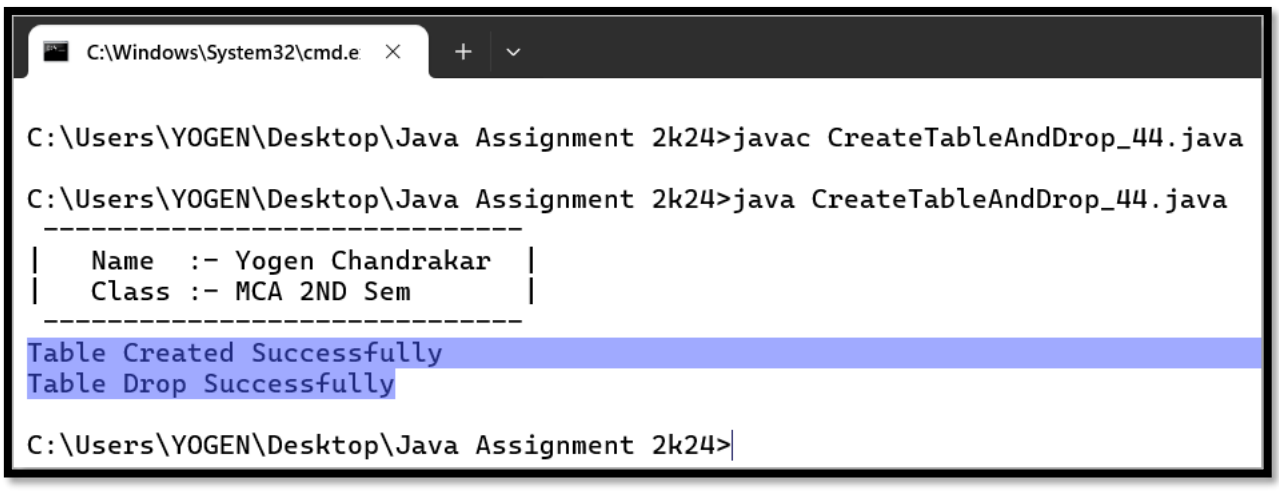
Program No.44

Write a Java program to create a table and drop it.

Code:-

```
//package javaapplication3;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
public class CreateTableAndDrop_44 {
   public static void main(String[] args) throws SQLException {

      Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"YOGEN03", "7410");
      try {
         // Establish connection to the Oracle databas
         // Create a statement
         Statement stmt = conn.createStatement();
         String sql = "CREATE TABLE demo (" + "id INT PRIMARY KEY," + "name
VARCHAR(50)," + "age INT," + "dept VARCHAR(50))";
         stmt.executeUpdate(sql);
         System.out.println("Table Created Successfully ");

         String sql4="drop table demo";   // drop table

        stmt.executeUpdate(sql4);
         stmt.close();
         conn.close();
         System.out.println("Table Drop Successfully ");
      } catch (SQLException e) {
         System.out.println("Table retrieve failed. Error: " + e.getMessage());
        }
   }}
```

Output :-



Program No.45

Write a Java program to insert multiple rows in a table using prepared statement.

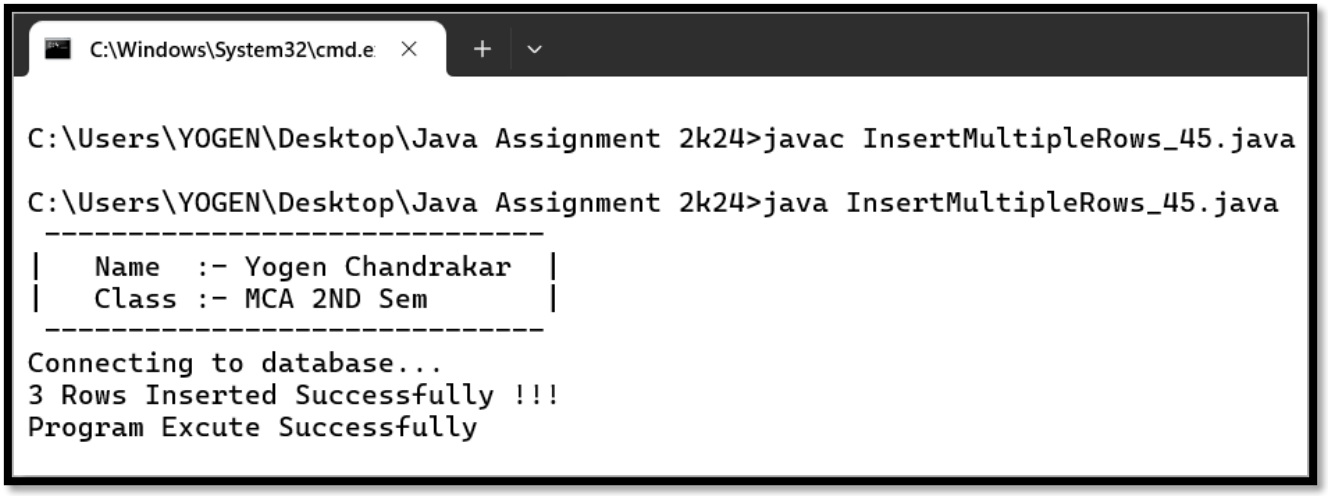
Code:-

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class InsertMultipleRows_45 {

    // JDBC URL, username, and password of the Oracle database server
    static final String JDBC_URL = "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER = "YOGEN03";
    static final String PASSWORD = "7410";

    // JDBC variables for opening and managing connection
    static Connection connection = null;
    static PreparedStatement preparedStatement = null;

    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        try {
            // Step 1: Register Oracle JDBC driver
            Class.forName("oracle.jdbc.driver.OracleDriver");

            // Step 2: Open a connection
            System.out.println("Connecting to database...");
            connection = DriverManager.getConnection(JDBC_URL, USER, PASSWORD);

            // Step 3: Create a SQL insert query
            String insertSQL = "INSERT INTO employee (id, name, age, dept) VALUES (?, ?, ?, ?)";

            // Step 4: Create PreparedStatement object
            preparedStatement = connection.prepareStatement(insertSQL);

            // Step 5: Set the parameters and add to batch
            connection.setAutoCommit(false);  // Disable auto-commit for batch processing

            // Insert first row
            preparedStatement.setInt(1, 101);
            preparedStatement.setString(2, "Yogen");
```

```
            preparedStatement.setInt(3, 21);
            preparedStatement.setString(4, "CS-IT");
            preparedStatement.addBatch();

            // Insert second row
            preparedStatement.setInt(1, 201);
            preparedStatement.setString(2, "RAM");
            preparedStatement.setInt(3, 22);
            preparedStatement.setString(4, "IT");
            preparedStatement.addBatch();

            // Insert third row
            preparedStatement.setInt(1, 301);
            preparedStatement.setString(2, "Monika");
            preparedStatement.setInt(3, 22);
            preparedStatement.setString(4, "Chemistry");
            preparedStatement.addBatch();

            // Execute batch insert
            int[] updateCounts = preparedStatement.executeBatch();
            connection.commit();  // Commit the transaction

            System.out.println(updateCounts.length +" Rows Inserted Successfully !!! " );

        } catch (SQLException se) {
            // Handle errors for JDBC
            se.printStackTrace();
            if (connection != null) {
                try {
                    connection.rollback();  // Rollback transaction in case of error
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        } catch (Exception e) {
            // Handle errors for Class.forName
            e.printStackTrace();
        } finally {
            // finally block used to close resources
            try {
                if (preparedStatement != null) preparedStatement.close();
            } catch (SQLException se2) {
                // nothing we can do
            }
            try {
```

```
            if (connection != null) connection.close();
          } catch (SQLException se) {
            se.printStackTrace();
          }
        }
        System.out.println("Program Excute Successfully");
    }
}
```

Output :-



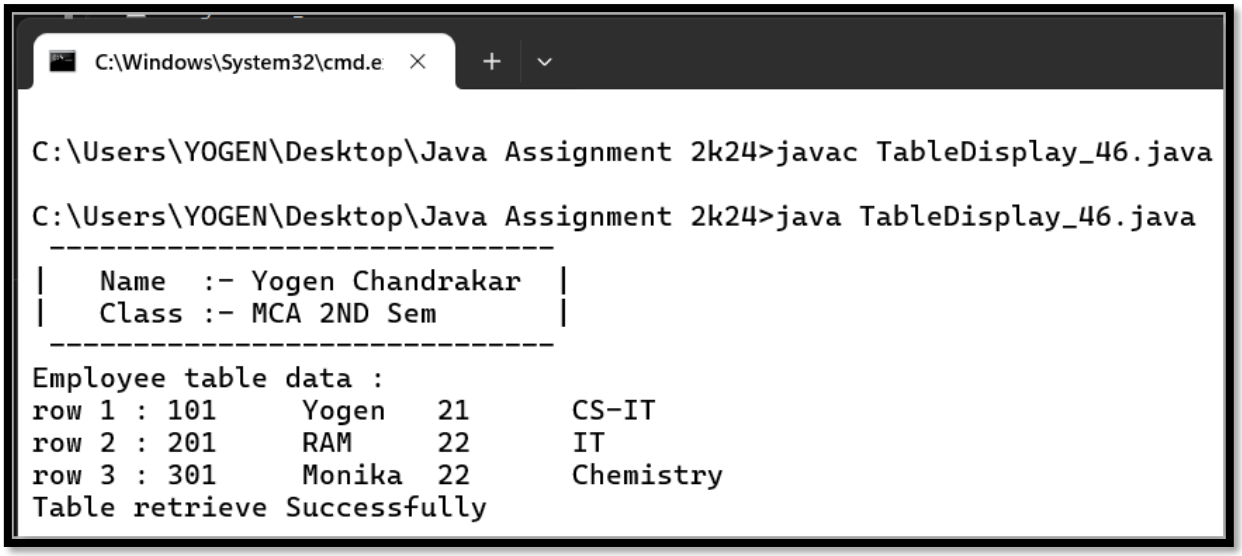
Program No.46

Write a Java program to display contents of a table on the console.

Code:-

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;


public class TableDisplay_46 {
   public static void main(String[] args) throws SQLException {
      System.out.println(" ------------------------------");
      System.out.println("|   Name  :- Yogen Chandrakar  |" );
      System.out.println("|   Class :- MCA 2ND Sem       |" );
      System.out.println(" ------------------------------");

      Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"YOGEN03", "7410");
      try {
         ResultSet rs = stmt.executeQuery("SELECT * FROM employee");
         int i = 1;
         System.out.println("Employee table data : ");
         while (rs.next()) {
            int emp_id = rs.getInt(1);
            String name = rs.getString(2);
            int age = rs.getInt(3);
            String dept = rs.getString(4);
            System.out.printf("row %d : %d\t%s\t%d\t%s\n", i++, emp_id, name, age, dept);
         }
          System.out.println("Table retrieve Successfully ");
         // Close the resources
         rs.close();
         stmt.close();
         conn.close();
      } catch (SQLException e) {
         System.out.println("Table retrieve failed. Error: " + e.getMessage());

      }
   }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>javac TableDisplay_46.java
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java TableDisplay_46.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
Employee table data :
row 1 : 101 Yogen 21 CS-IT
row 2 : 201 RAM 22 IT
row 3 : 301 Monika 22 Chemistry
Table retrieve Successfully

Program No.47

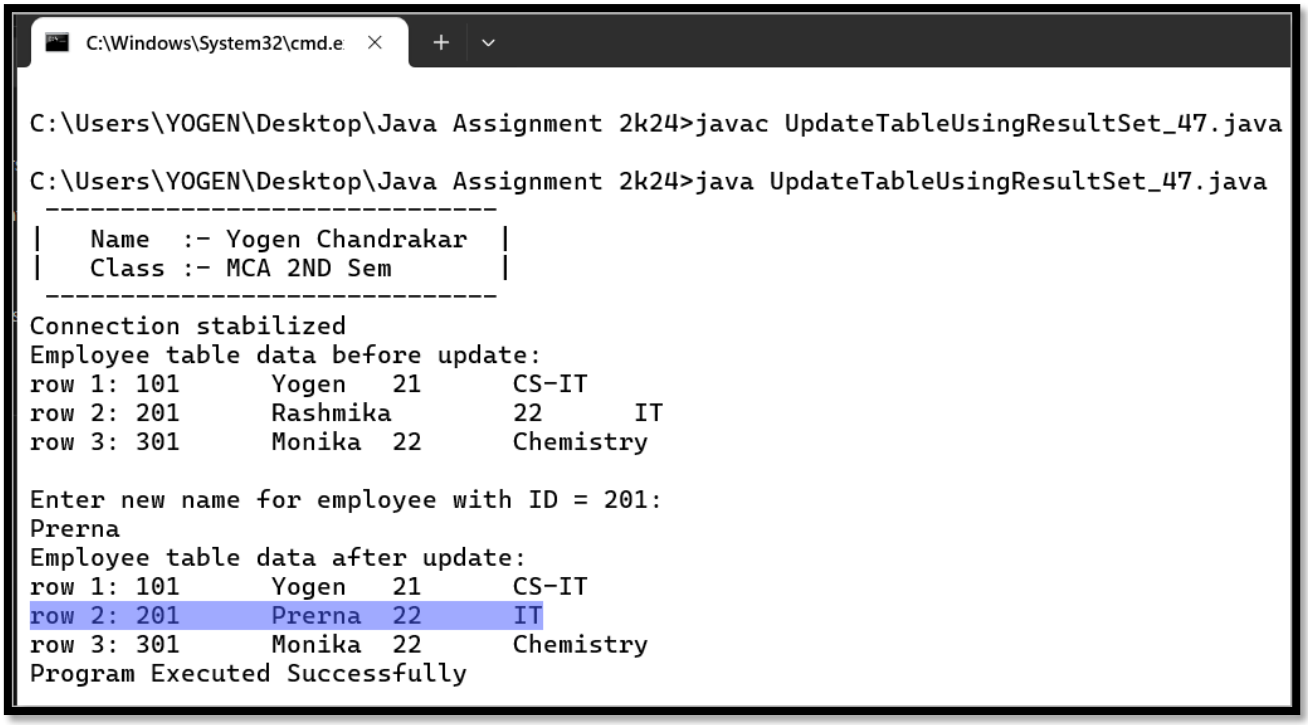
Write a Java program to update rows using result set.

Code:-

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class UpdateTableUsingResultSet_47 {

    // JDBC URL, username, and password of the Oracle database server
    static final String JDBC_URL = "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER = "YOGEN03";
    static final String PASSWORD = "7410";

    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        Statement stmt = null;
        ResultSet rset = null;

        try (Connection con = DriverManager.getConnection(JDBC_URL, USER, PASSWORD)) {
            System.out.println("Connection stabilized");

            // Create a Statement object for updatable ResultSet
            stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE);
            String sql = "SELECT id, name, age, dept FROM employee";
            rset = stmt.executeQuery(sql);

            // Display initial data
            System.out.println("Employee table data before update:");
            displayData(rset);

            // Update starts from here
            Scanner sc = new Scanner(System.in);
            System.out.println("\nEnter new name for employee with ID = 201:");
            String newName = sc.nextLine();

            // Re-initialize rset so it points to the start of the result set
```

```
            rset.beforeFirst();
            while (rset.next()) {
                int id = rset.getInt("id");
                if (id == 201) {
                    rset.updateString("name", newName);
                    rset.updateRow();  // Commit the update to the database
                }
            }

            // Display updated data
            rset.beforeFirst();
            System.out.println("Employee table data after update:");
            displayData(rset);
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            // Close resources
            closeResources(rset, stmt);
        }
        System.out.println("Program Executed Successfully");
    }

    // Utility method to display data
    private static void displayData(ResultSet rset) throws SQLException {
        int i = 1;
        while (rset.next()) {
            int emp_id = rset.getInt("id");
            String name = rset.getString("name");
            int age = rset.getInt("age");
            String dept = rset.getString("dept");
            System.out.printf("row %d: %d\t%s\t%d\t%s\n", i++, emp_id, name, age, dept);
        }
    }

    // Utility method to close resources
    private static void closeResources(ResultSet rset, Statement stmt) {
        try {
            if (rset != null) rset.close();
            if (stmt != null) stmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>javac UpdateTableUsingResultSet_47.java
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java UpdateTableUsingResultSet_47.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
Connection stabilized
Employee table data before update:
row 1: 101 Yogen 21 CS-IT
row 2: 201 Rashmika 22 IT
row 3: 301 Monika 22 Chemistry
Enter new name for employee with ID = 201:
Prerna
Employee table data after update:
row 1: 101 Yogen 21 CS-IT
row 2: 201 Prerna 22 IT
row 3: 301 Monika 22 Chemistry
Program Executed Successfully

Program No.48

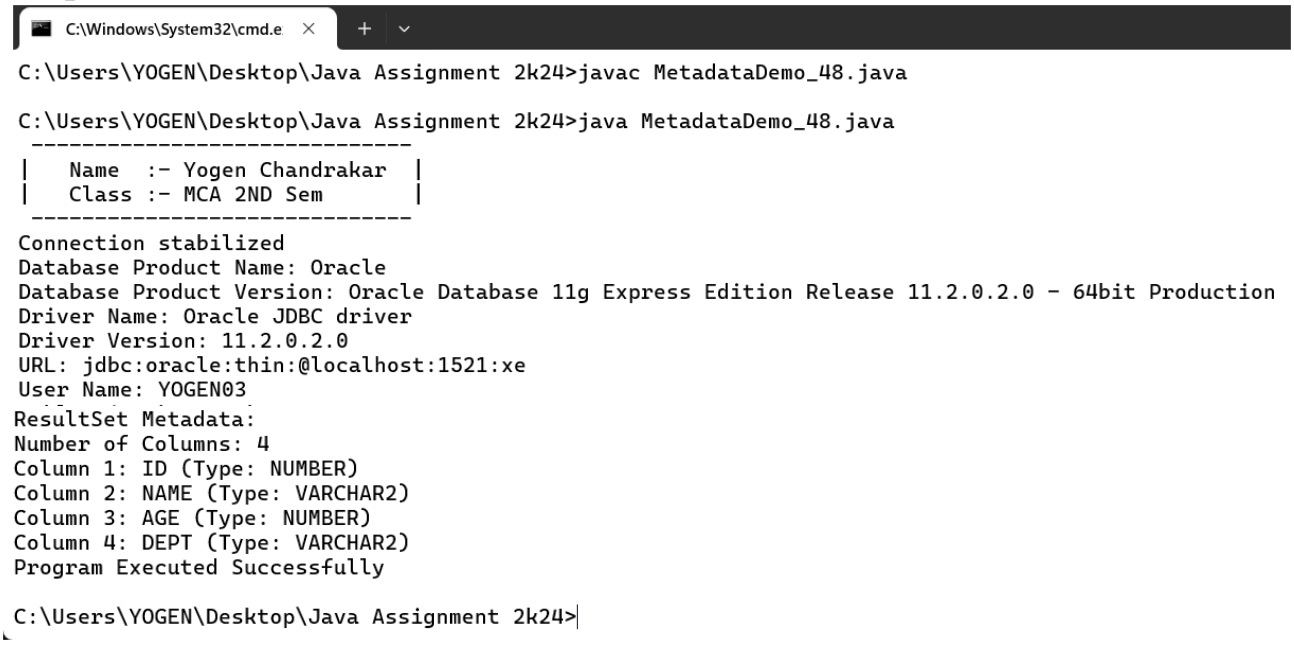
Write a Java program to describe the functions of metadata objects.(resultset & database).

Code:-

```
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;

public class MetadataDemo_48 {

   // JDBC URL, username, and password of the Oracle database server
   static final String JDBC_URL = "jdbc:oracle:thin:@localhost:1521:xe";
   static final String USER = "YOGEN03";
   static final String PASSWORD = "7410";

   public static void main(String[] args) {
      System.out.println(" ------------------------------");
      System.out.println("|   Name  :- Yogen Chandrakar  |" );
      System.out.println("|   Class :- MCA 2ND Sem       |" );
      System.out.println(" ------------------------------");
      Statement stmt = null;
      ResultSet rset = null;

      try (Connection con = DriverManager.getConnection(JDBC_URL, USER, PASSWORD)) {
         System.out.println("Connection stabilized");

         // Database Metadata
         DatabaseMetaData dbMetaData = con.getMetaData();
         System.out.println("Database Product Name: " + dbMetaData.getDatabaseProductName());
         System.out.println("Database Product Version: " + dbMetaData.getDatabaseProductVersion());
         System.out.println("Driver Name: " + dbMetaData.getDriverName());
         System.out.println("Driver Version: " + dbMetaData.getDriverVersion());
         System.out.println("URL: " + dbMetaData.getURL());
         System.out.println("User Name: " + dbMetaData.getUserName());
         System.out.println("Tables in the Database:");
         ResultSet tables = dbMetaData.getTables(null, null, "%", new String[] { "TABLE" });
         while (tables.next()) {
            System.out.println("\t" + tables.getString("TABLE_NAME"));
         }
```

```
            // Creating Statement and Executing Query
            stmt = con.createStatement();
            String sql = "SELECT * FROM employee";
            rset = stmt.executeQuery(sql);

            // ResultSet Metadata
            ResultSetMetaData rsMetaData = rset.getMetaData();
            int columnCount = rsMetaData.getColumnCount();
            System.out.println("\nResultSet Metadata:");
            System.out.println("Number of Columns: " + columnCount);
            for (int i = 1; i <= columnCount; i++) {
                System.out.println("Column " + i + ": " + rsMetaData.getColumnName(i) + " (Type: " +
    rsMetaData.getColumnTypeName(i) + ")");
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            // Close resources
            try {
                if (rset != null) rset.close();
                if (stmt != null) stmt.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        System.out.println("Program Executed Successfully");
    }
}
```

Output :-



Program No.49

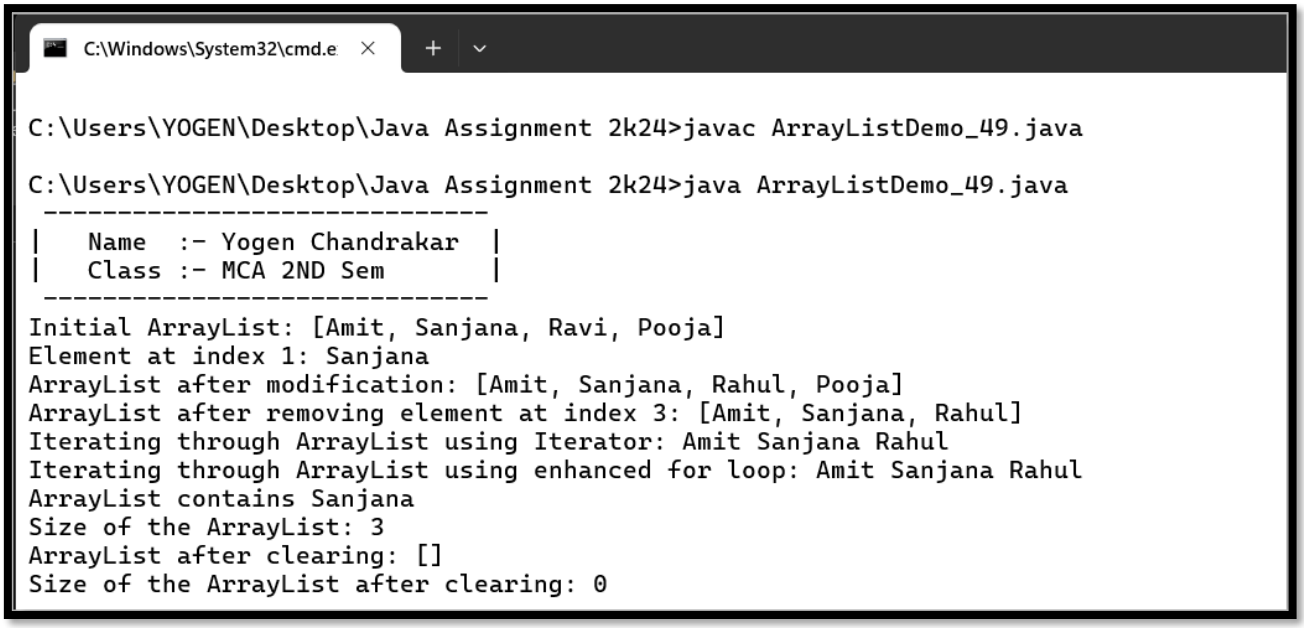
Write a Java program to demonstrate the ArrayList class.

Code:-

```
import java.util.ArrayList;
import java.util.Iterator;

public class ArrayListDemo_49 {
    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        // Create an ArrayList
        ArrayList<String> names = new ArrayList<>();

        // Add elements to the ArrayList
        names.add("Amit");
        names.add("Sanjana");
        names.add("Ravi");
        names.add("Pooja");

        // Display the ArrayList
        System.out.println("Initial ArrayList: " + names);

        // Access elements
        System.out.println("Element at index 1: " + names.get(1));

        // Modify elements
        names.set(2, "Rahul");
        System.out.println("ArrayList after modification: " + names);

        // Remove elements
        names.remove(3);
        System.out.println("ArrayList after removing element at index 3: " + names);

        // Iterate through the ArrayList using Iterator
        System.out.print("Iterating through ArrayList using Iterator: ");
        Iterator<String> iterator = names.iterator();
        while (iterator.hasNext()) {
            System.out.print(iterator.next() + " ");
        }
        System.out.println();

        // Iterate through the ArrayList using enhanced for loop
```

```
        System.out.print("Iterating through ArrayList using enhanced for loop: ");
        for (String name : names) {
            System.out.print(name + " ");
        }
        System.out.println();

        // Check if the ArrayList contains a specific element
        String nameToFind = "Sanjana";
        if (names.contains(nameToFind)) {
            System.out.println("ArrayList contains " + nameToFind);
        } else {
            System.out.println("ArrayList does not contain " + nameToFind);
        }

        // Get the size of the ArrayList
        System.out.println("Size of the ArrayList: " + names.size());

        // Clear the ArrayList
        names.clear();
        System.out.println("ArrayList after clearing: " + names);
        System.out.println("Size of the ArrayList after clearing: " + names.size());
    }
}
```

Output :-



Program No.50

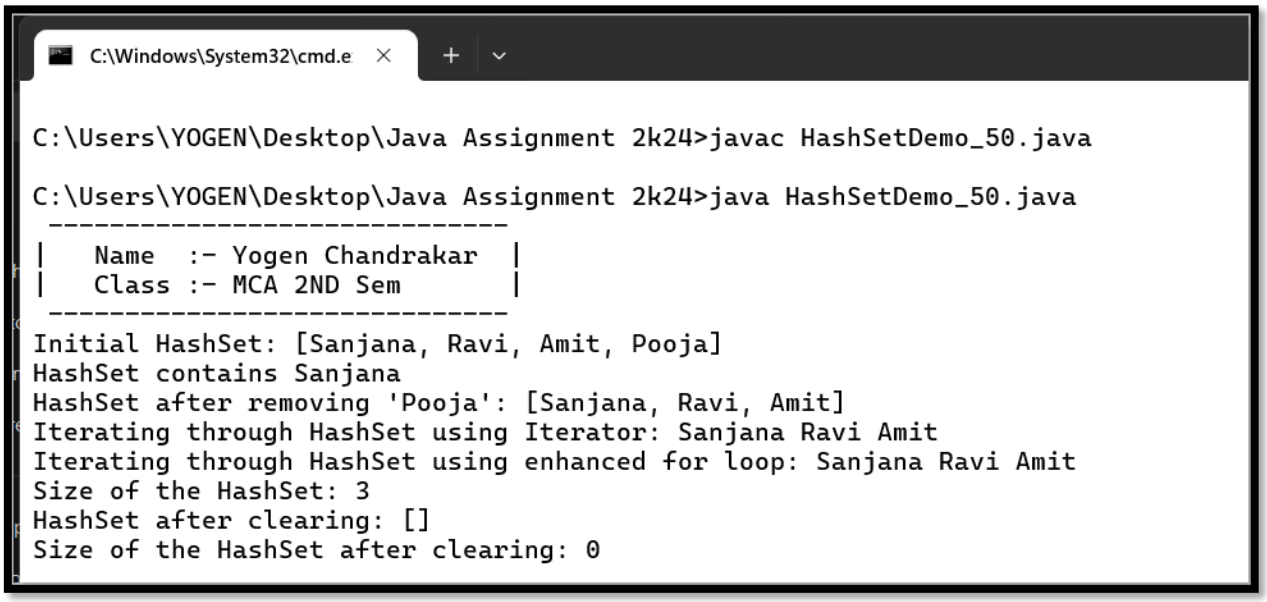
Write a Java program to demonstrate the HashSet class.

Code:-

```
import java.util.HashSet;
import java.util.Iterator;

public class HashSetDemo_50 {
    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        // Create a HashSet
        HashSet<String> names = new HashSet<>();

        // Add elements to the HashSet
        names.add("Amit");
        names.add("Sanjana");
        names.add("Ravi");
        names.add("Pooja");
        names.add("Amit"); // Duplicate element

        // Display the HashSet
        System.out.println("Initial HashSet: " + names);

        // Check if a specific element exists
        String nameToFind = "Sanjana";
        if (names.contains(nameToFind)) {
            System.out.println("HashSet contains " + nameToFind);
        } else {
            System.out.println("HashSet does not contain " + nameToFind);
        }

        // Remove an element
        names.remove("Pooja");
        System.out.println("HashSet after removing 'Pooja': " + names);

        // Iterate through the HashSet using Iterator
        System.out.print("Iterating through HashSet using Iterator: ");
        Iterator<String> iterator = names.iterator();
        while (iterator.hasNext()) {
            System.out.print(iterator.next() + " ");
        }
        System.out.println();
```

```
        // Iterate through the HashSet using enhanced for loop
        System.out.print("Iterating through HashSet using enhanced for loop: ");
        for (String name : names) {
            System.out.print(name + " ");
        }
        System.out.println();

        // Get the size of the HashSet
        System.out.println("Size of the HashSet: " + names.size());

        // Clear the HashSet
        names.clear();
        System.out.println("HashSet after clearing: " + names);
        System.out.println("Size of the HashSet after clearing: " + names.size());
    }
}
```

Output :-



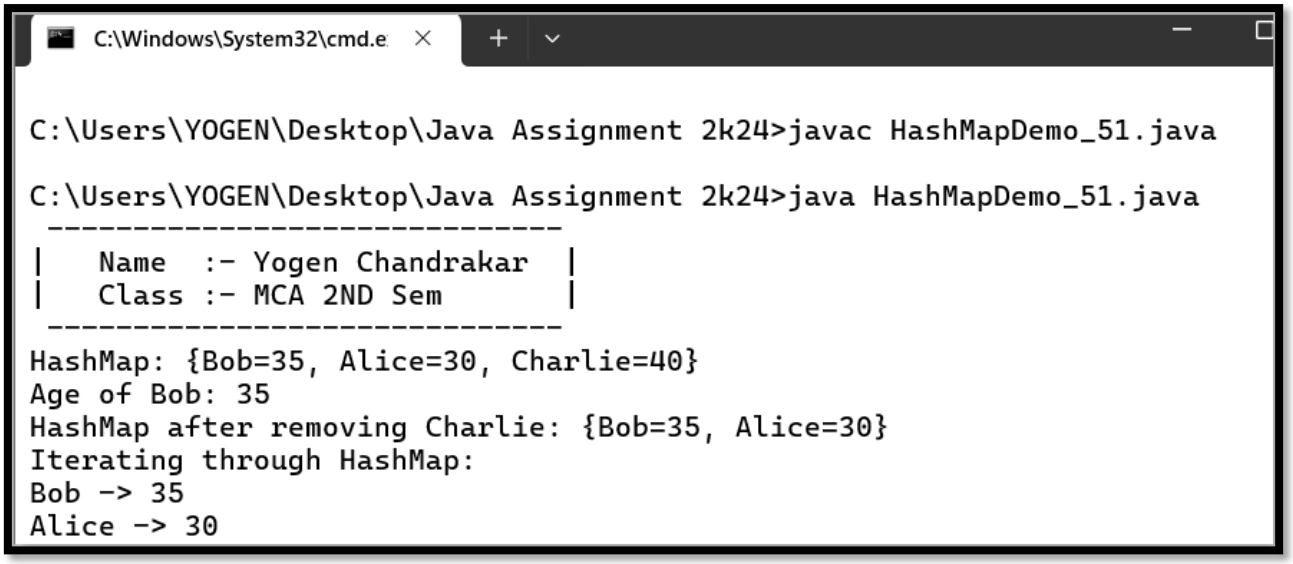
Program No.51

Write a Java program to demonstrate the HashMap class.

Code:-

```
import java.util.HashMap;
import java.util.Map;
public class HashMapDemo_51 {
   public static void main(String[] args) {
      System.out.println(" ------------------------------");
      System.out.println("|   Name  :- Yogen Chandrakar  |" );
      System.out.println("|   Class :- MCA 2ND Sem       |" );
      System.out.println(" ------------------------------");
      // Create a HashMap
      HashMap<String, Integer> ages = new HashMap<>();

      // Add elements to the HashMap
      ages.put("Alice", 30);
      ages.put("Bob", 35);
      ages.put("Charlie", 40);

      // Display the HashMap
      System.out.println("HashMap: " + ages);

      // Access elements
      System.out.println("Age of Bob: " + ages.get("Bob"));

      // Remove an element
      ages.remove("Charlie");
      System.out.println("HashMap after removing Charlie: " + ages);

      // Iterate through the HashMap
      System.out.println("Iterating through HashMap:");
      for (Map.Entry<String, Integer> entry : ages.entrySet()) {
         System.out.println(entry.getKey() + " -> " + entry.getValue());
      }
   }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>javac HashMapDemo_51.java
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java HashMapDemo_51.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
HashMap: {Bob=35, Alice=30, Charlie=40}
Age of Bob: 35
HashMap after removing Charlie: {Bob=35, Alice=30}
Iterating through HashMap:
Bob -> 35
Alice -> 30

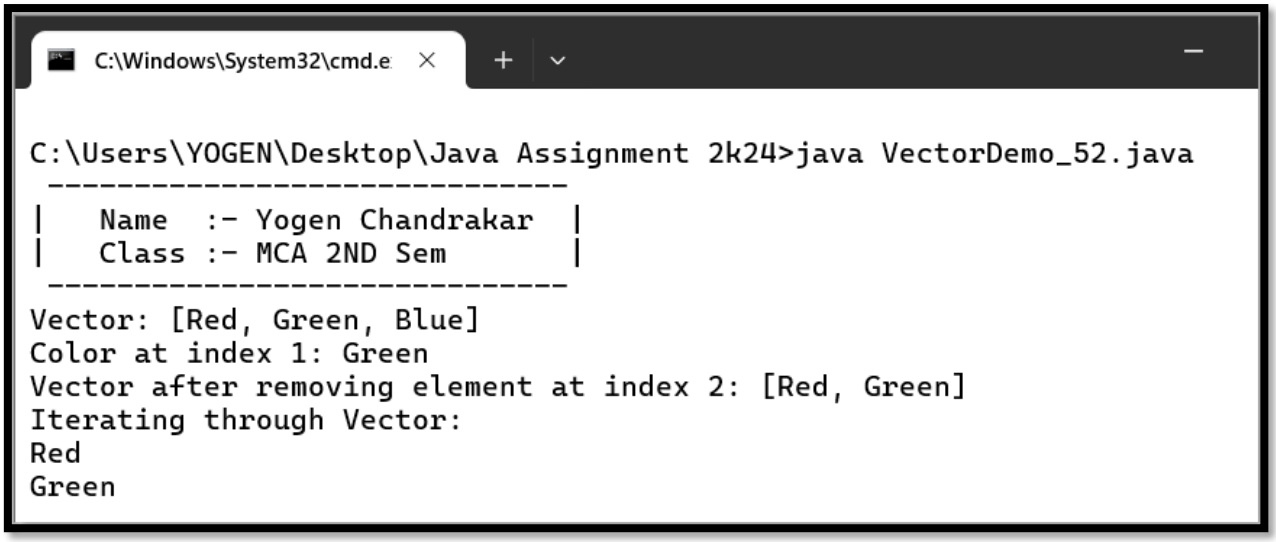
Program No.52

Write a Java program to demonstrate the Vector class.

Code:-

```
import java.util.Vector;
public class VectorDemo_52 {
    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        // Create a Vector
        Vector<String> colors = new Vector<>();

        // Add elements to the Vector
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");

        // Display the Vector
        System.out.println("Vector: " + colors);

        // Access elements
        System.out.println("Color at index 1: " + colors.get(1));

        // Remove an element
        colors.remove(2);
        System.out.println("Vector after removing element at index 2: " + colors);

        // Iterate through the Vector
        System.out.println("Iterating through Vector:");
        for (String color : colors) {
            System.out.println(color);
        }
    }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java VectorDemo_52.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
Vector: [Red, Green, Blue]
Color at index 1: Green
Vector after removing element at index 2: [Red, Green]
Iterating through Vector:
Red
Green

Program No.53

Write a Java program to demonstrate the LinkedList class.

Code:-

```
import java.util.LinkedList;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class LinkedListDemon_53 {
   public static void main(String[] args) {
      System.out.println(" ------------------------------");
      System.out.println("|   Name  :- Yogen Chandrakar  |" );
      System.out.println("|   Class :- MCA 2ND Sem       |" );
      System.out.println(" ------------------------------");
      // Create a LinkedList
      LinkedList<String> names = new LinkedList<>();

      // Add elements to the LinkedList
      names.add("Alice");
      names.add("Bob");
      names.add("Charlie");

      // Display the LinkedList
      System.out.println("LinkedList: " + names);

      // Access elements
      System.out.println("First element: " + names.getFirst());
      System.out.println("Last element: " + names.getLast());

      // Remove an element
      names.removeLast();
   }
}
```

Output :-

```
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java LinkedListDemon_53.java
 ------------------------------
|   Name  :- Yogen Chandrakar  |
|   Class :- MCA 2ND Sem       |
 ------------------------------
LinkedList: [Alice, Bob, Charlie]
First element: Alice
Last element: Charlie
```

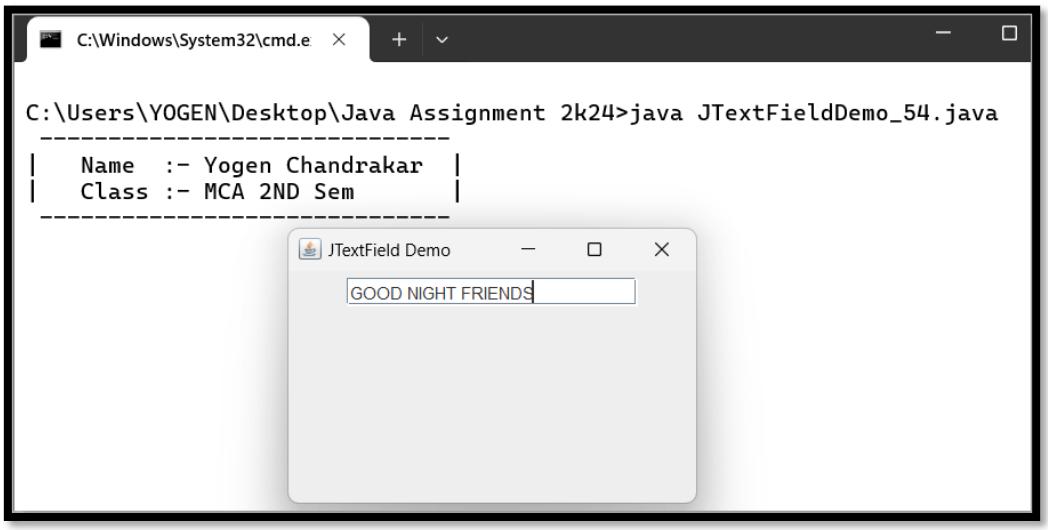
Program No.54

Write a Java program to demonstrate the JTextField class.

Code:-

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class JTextFieldDemo_54 {
   public static void main(String[] args) {
      System.out.println("|   Name  :- Yogen Chandrakar  |" );
      System.out.println("|   Class :- MCA 2ND Sem       |" );
      System.out.println(" ------------------------------");
      // Create a JFrame
      JFrame frame = new JFrame("JTextField Demo");

      // Create a JPanel
      JPanel panel = new JPanel();

      JTextField textField = new JTextField(20);
      textField.setText("Enter text here");

      // Add JTextField to the panel
      panel.add(textField);

      frame.add(panel);
      // Set frame size and make it visible
      frame.setSize(300, 200);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
   }
}
```

Output :-



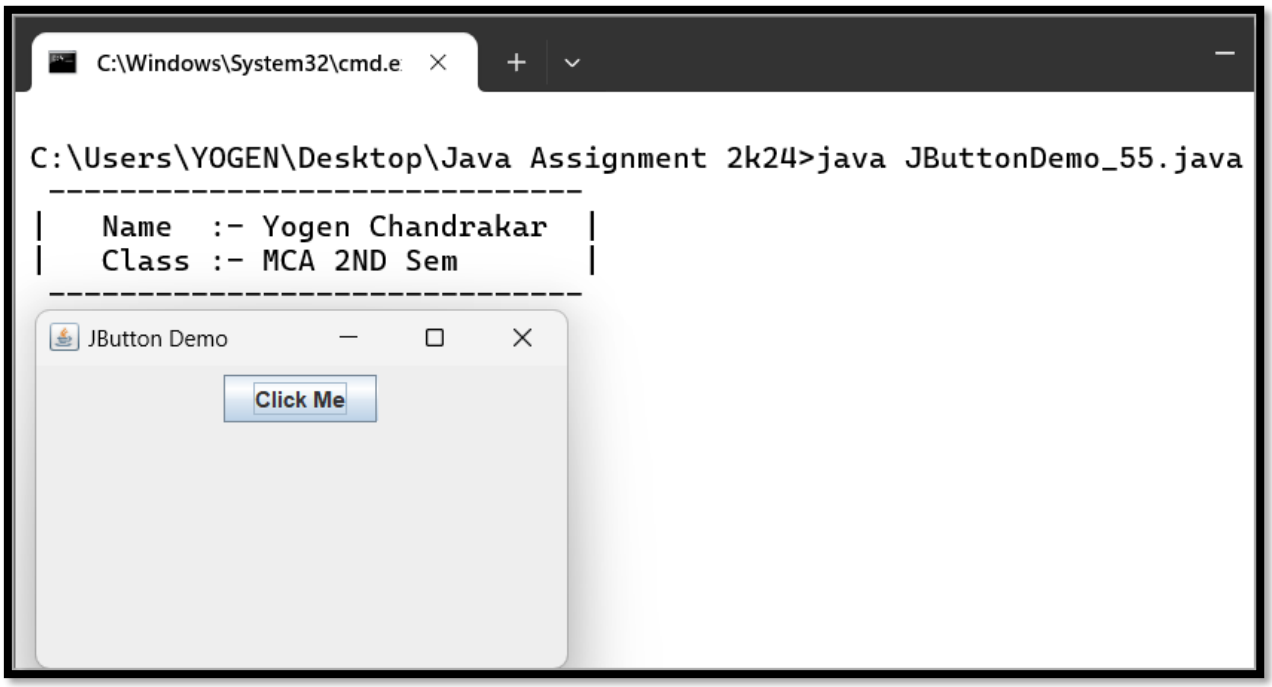
Program No.55

Write a Java program to demonstrate the JButton class.

Code:-

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class JButtonDemo_55 {
    public static void main(String[] args) {
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        // Create a JFrame
        JFrame frame = new JFrame("JButton Demo");
        // Create a JPanel
        JPanel panel = new JPanel();
        // Create a JButton
        JButton button = new JButton("Click Me");
        // Add JButton to the panel
        panel.add(button);
        // Add panel to the frame
        frame.add(panel);
        // Set frame size and make it visible
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Output :-



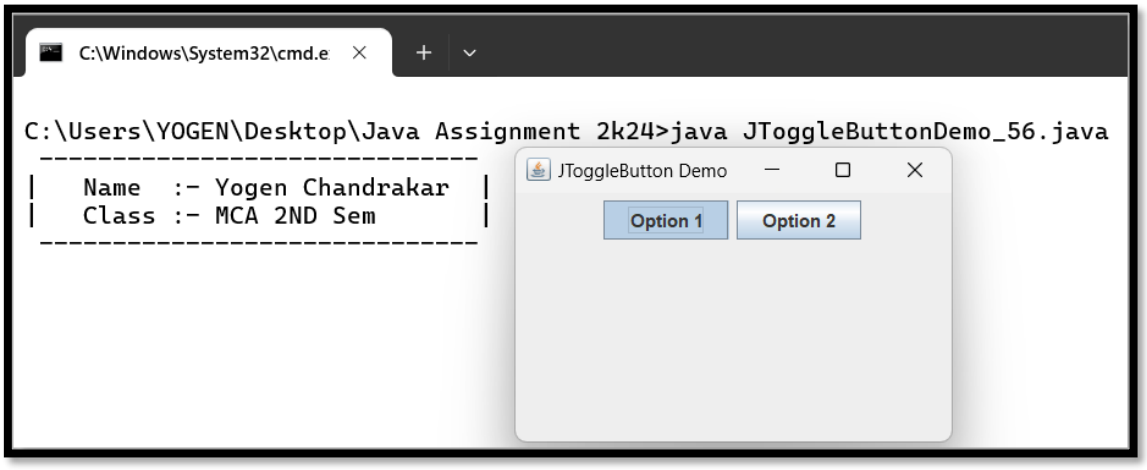
Program No.56

Write a Java program to demonstrate the JToggleButton class.

Code:-

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JToggleButton;

public class JToggleButtonDemo_56 {
    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        // Create a JFrame
        JFrame frame = new JFrame("JToggleButton Demo");
        JPanel panel = new JPanel();

        // Create JToggleButtons
        JToggleButton toggleButton1 = new JToggleButton("Option 1");
        JToggleButton toggleButton2 = new JToggleButton("Option 2");

        // Add JToggleButtons to the panel
        panel.add(toggleButton1);
        panel.add(toggleButton2);
        // Add panel to the frame
        frame.add(panel);
        // Set frame size and make it visible
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Output :-



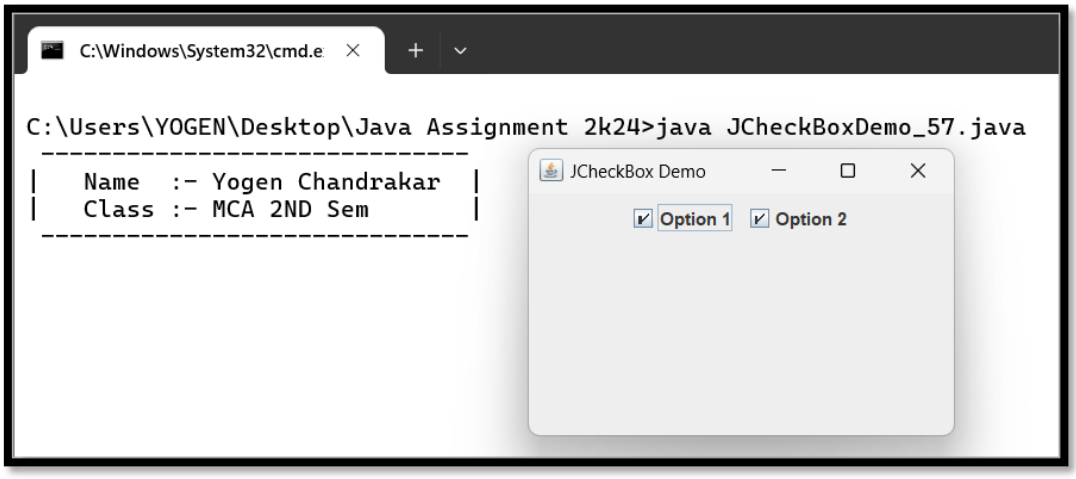
Program No.57

Write a Java program to demonstrate the JCheckbox class.

Code:-

```
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class JCheckBoxDemo_57 {
   public static void main(String[] args) {
      System.out.println(" ------------------------------");
      System.out.println("|   Name  :- Yogen Chandrakar  |" );
      System.out.println("|   Class :- MCA 2ND Sem       |" );
      System.out.println(" ------------------------------");
      // Create a JFrame
      JFrame frame = new JFrame("JCheckBox Demo");

      // Create a JPanel
      JPanel panel = new JPanel();
      // Create JCheckBoxes
      JCheckBox checkBox1 = new JCheckBox("Option 1");
      JCheckBox checkBox2 = new JCheckBox("Option 2");
      // Add JCheckBoxes to the panel
      panel.add(checkBox1);
      panel.add(checkBox2);
      // Add panel to the frame
      frame.add(panel);
      // Set frame size and make it visible
      frame.setSize(300, 200);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
   }
}
```

Output :-



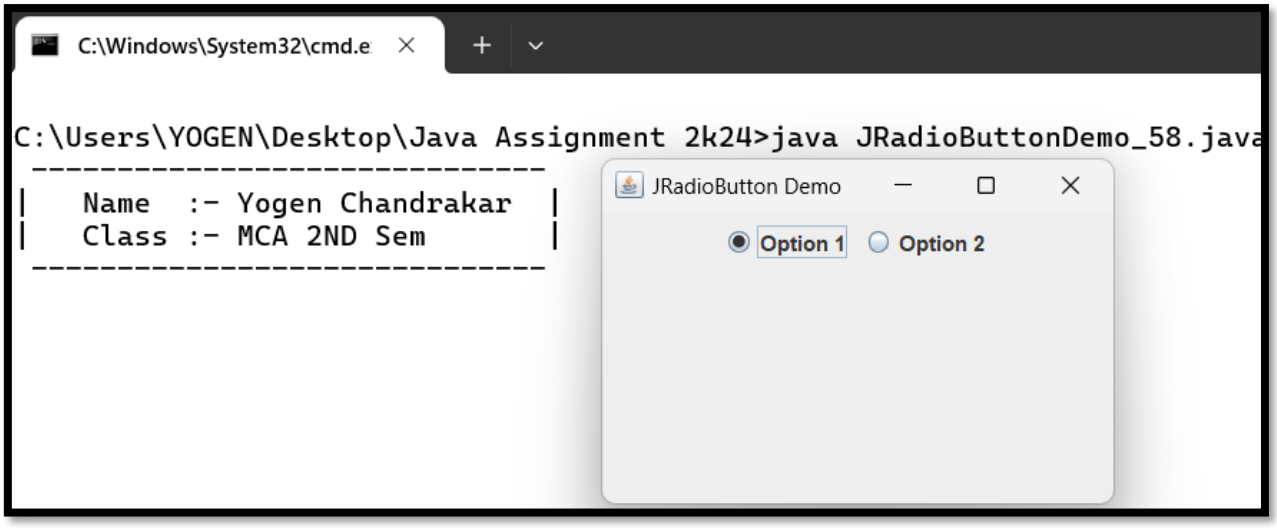
Program No.58

Write a Java program to demonstrate the JRadioButton class.

Code:-

```
import javax.swing.ButtonGroup;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JRadioButton;

public class JRadioButtonDemo_58 {
    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        // Create a JFrame
        JFrame frame = new JFrame("JRadioButton Demo");

        // Create a JPanel
        JPanel panel = new JPanel();

        // Create JRadioButtons
        JRadioButton radioButton1 = new JRadioButton("Option 1");
        JRadioButton radioButton2 = new JRadioButton("Option 2");

        // Create a ButtonGroup and add JRadioButtons to it
        ButtonGroup group = new ButtonGroup();
        group.add(radioButton1);
        group.add(radioButton2);

        // Add JRadioButtons to the panel
        panel.add(radioButton1);
        panel.add(radioButton2);

        // Add panel to the frame
        frame.add(panel);

        // Set frame size and make it visible
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Output :-


C:\Windows\System32\cmd.e
C:\Users\YOGEN\Desktop\Java Assignment 2k24>java JRadioButtonDemo_58.java
| Name :- Yogen Chandrakar |
| Class :- MCA 2ND Sem |
JRadioButton Demo
Option 1
Option 2

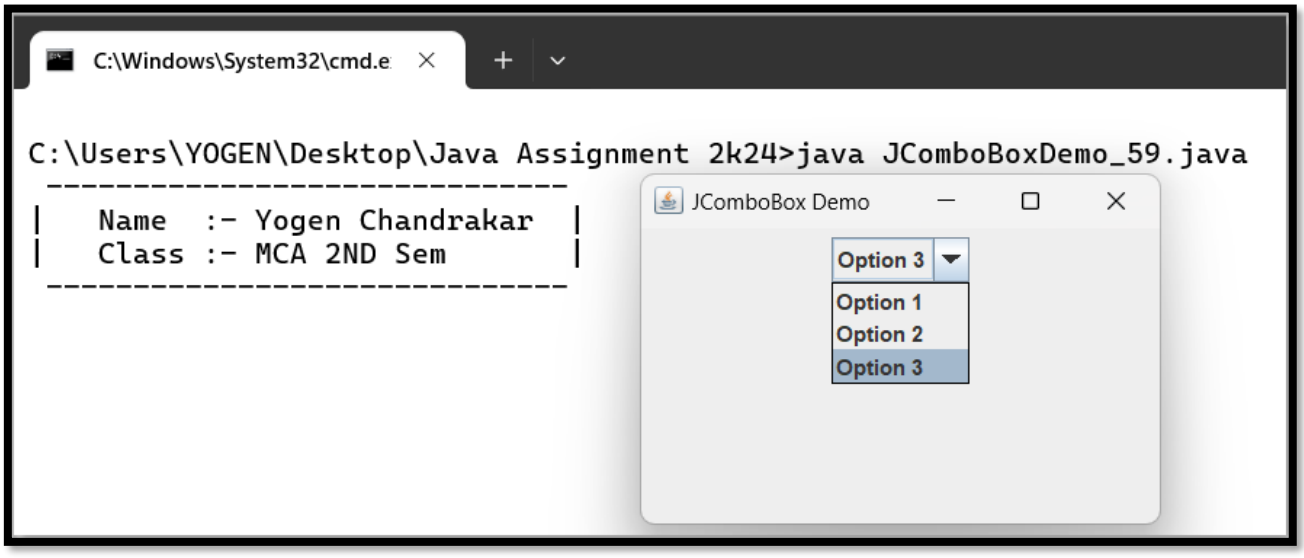
Program No.59

Write a Java program to demonstrate the JComboBox class.

Code:-

```
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class JComboBoxDemo_59 {
   public static void main(String[] args) {
      System.out.println("|   Name  :- Yogen Chandrakar  |" );
      System.out.println("|   Class :- MCA 2ND Sem       |" );

      // Create a JFrame
      JFrame frame = new JFrame("JComboBox Demo");
      // Create a JPanel
      JPanel panel = new JPanel();
      // Create an array of items
      String[] items = { "Option 1", "Option 2", "Option 3" };

      // Create a JComboBox with the array of items
      JComboBox<String> comboBox = new JComboBox<>(items);
      // Add JComboBox to the panel
      panel.add(comboBox);
      // Add panel to the frame
      frame.add(panel);
      // Set frame size and make it visible
      frame.setSize(300, 200);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
   }}
```

Output :-



Program No.60

Write a Java program to demonstrate the JList class.

Code:-

```
import javax.swing.DefaultListModel;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JScrollPane;

public class JListDemo_60 {
    public static void main(String[] args) {
        System.out.println(" ------------------------------");
        System.out.println("|   Name  :- Yogen Chandrakar  |" );
        System.out.println("|   Class :- MCA 2ND Sem       |" );
        System.out.println(" ------------------------------");
        // Create a JFrame
        JFrame frame = new JFrame("JList Demo");

        // Create a JPanel
        JPanel panel = new JPanel();

        // Create a DefaultListModel
        DefaultListModel<String> model = new DefaultListModel<>();

        // Add items to the DefaultListModel
        model.addElement("Item 1");
        model.addElement("Item 2");
        model.addElement("Item 3");

        // Create a JList with the DefaultListModel
        JList<String> list = new JList<>(model);

        // Create a JScrollPane and add JList to it
        JScrollPane scrollPane = new JScrollPane(list);

        // Add JScrollPane to the panel
        panel.add(scrollPane);

        // Add panel to the frame
        frame.add(panel);

        // Set frame size and make it visible
        frame.setSize(300, 200);
```

```
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Output :-