Laboratory Work

Subject: Java Technologies

Branch: B.Tech. (CE)

Semester: IV

Batch: A3

Student Roll No: CE063

Student Name: Bhalodiya Drashti Chandrakantbhai

Department of Computer Engineering,

Faculty of Technology,

Dharmsinh Desai University, Nadiad – 387001

Gujarat, INDIA.

LAB 1

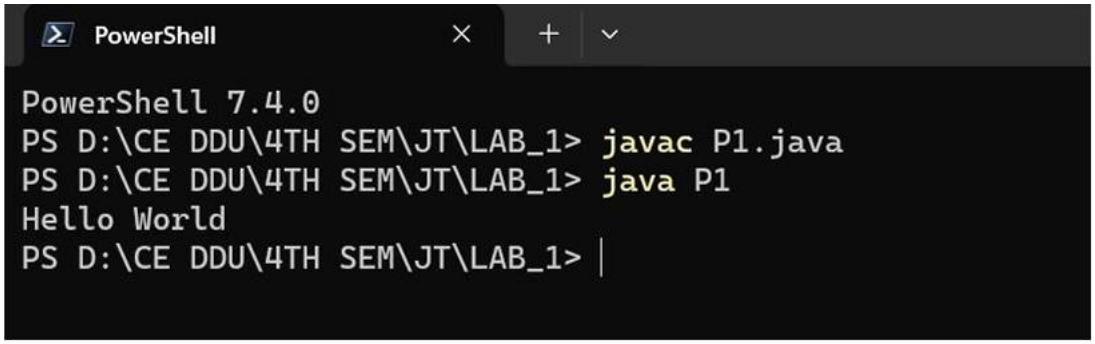
AIM : Topics: print(), println(), Scanner class, 1-D, 2-D array, jagged array

(1) Write a Java program to display “Hello World”.

CODE :

```
public class P1
{
        public static void main(String[] args)
        {
                System.out.println("Hello World");
         }
}
```

OUTPUT :



(2) Write a Java program to print numbers between 1 to n which are divisible by 3, 5 and by both(3 and 5) by taking n as an input from the user.

CODE :

```
import java.util.Scanner;
public class P2
{
        public static void main(String args[] )
        {
                 Scanner scn = new Scanner(System.in);
                System.out.print("enter the numbe n : ");
                int n = scn.nextInt();
                for(int i = 1; i <= n; i++)
                {
                        if(i%3==0 || i%5==0)
                        {
                                System.out.print(i + " ");
                        }
                }
        }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_1> javac P2.java
PS D:\CE DDU\4TH SEM\JT\LAB_1> java P2
enter the number n : 10
3 5 6 9 10
PS D:\CE DDU\4TH SEM\JT\LAB_1>
```

(3) Write a class named Greeter that prompts the user for his or her name, and then prints a personalized greeting. As an example, if the user entered “Era”, the program should respond “Hello Era!”.

CODE :

```
import java.util.Scanner;

class P3
{
   public static void main(String args[])
   {
        Scanner scn = new Scanner(System.in);
        System.out.print("Enter your name : ");
        String name = scn.nextLine();
        System.out.println("Hello " + name + "!");
    }
}
```

```
PS D:\CE DDU\4TH SEM\JT\LAB_1> javac P3.java
PS D:\CE DDU\4TH SEM\JT\LAB_1> java P3
Enter your name : Drashti
Hello Drashti!
PS D:\CE DDU\4TH SEM\JT\LAB_1>
```

(4) Write a Java program that takes Name, Roll No and marks of 5 subjects as input and gives a formatted output as:
Name: ABCD
Roll No. : 1
Average: 84
Also display the grade (e.g. A, B, C…etc) using the average.

CODE :

```
import java.util.Scanner;

class P4
{
        public static void main(String args[])
        {
```

```
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter your name : ");
        String name = scn.nextLine();
        System.out.println("Enter your Roll no. : ");
        int roll = scn.nextInt();

        double sum = 0.0;

        System.out.println("Enter the marks of 5 subjects of your
                4th sem : ");
        for(int i = 0; i < 5; i++){
                int mark = scn.nextInt();
                sum += mark;
        }

        double avg = sum/5;

        System.out.println("Name : " + name);
        System.out.println("Roll No : " + roll);
        System.out.println("Average : " + avg);

        if(avg >= 0 && avg <= 20)
            System.out.println("Grade : E");
        else if(avg > 20 && avg <= 40)
            System.out.println("Grade : D");
        else if(avg > 40 && avg <= 60)
            System.out.println("Grade : C");
        else if(avg > 60 && avg <= 80)
            System.out.println("Grade : B");
        else if(avg > 80 && avg <= 100)
            System.out.println("Grade : A");
    }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_1> javac P4.java
PS D:\CE DDU\4TH SEM\JT\LAB_1> java P4
Enter your name :
Drashti
Enter your Roll no. :
63
Enter the marks of 5 subjects of your 4th sem :
80
70
90
60
90
Name : Drashti
Roll No : 63
Average : 78.0
Grade : B
PS D:\CE DDU\4TH SEM\JT\LAB_1>
```

(5) Calculate and return the sum of all the even numbers present in the numbers array passed to the method calculateSumOfEvenNumbers. Implement the logic inside calculateSumOfEvenNumbers() method. Test the functionalities using the main() method of the Tester class.

CODE :

```
import java.util.Scanner;

class Tester
{
  int calculateSumOfEvenNumbers(int arr[], int n)
  {
      int sum = 0;
      for(int i = 0; i < n; i++)
      {
          if(arr[i]%2 == 0)
```

```
            sum += arr[i];
        }
        return sum;
    }
}

class P5
{
    public static void main(String args[])
    {
         Scanner scn = new Scanner(System.in);
         int n;
         System.out.print("Enter the number for the array : ");
        n = scn.nextInt();

        int[] arr;
        arr = new int[n];
        System.out.print("Enter the data for the array : ");
        for(int i = 0; i < n; i++)
        {
            int num = scn.nextInt();
            arr[i] = num;
         }

         Tester T = new Tester();

         int ans = T.calculateSumOfEvenNumbers(arr,n);
         System.out.print("sum of the all even numbers of given array is : " +
                                ans);
         }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_1> javac P5.java
PS D:\CE DDU\4TH SEM\JT\LAB_1> java P5
Enter the number for the array : 8
Enter the data for the array : 68 79 86 99 23 2 41 100
sum of the all even numbers of given array is : 256
PS D:\CE DDU\4TH SEM\JT\LAB_1> javac P5.java
PS D:\CE DDU\4TH SEM\JT\LAB_1> java P5
Enter the number for the array : 10
Enter the data for the array : 1 2 3 4 5 6 7 8 9 10
sum of the all even numbers of given array is : 30
PS D:\CE DDU\4TH SEM\JT\LAB_1>
```

(6) Write a program to perform matrix addition and matrix multiplication on two Given matrices. Use for-each form of for loop to display the matrices.

CODE :

```
import java.util.Scanner;

class Addition
{
    void addition(int arr1[][], int arr2[][])
    {
        int n1 = arr1.length;
        int m1 = arr1[0].length;
        int n2 = arr2.length;
        int m2 = arr2[0].length;


     if(n1 == n2 && m1 == m2)
     {
        int[][] sum;
        sum = new int[n1][m1];
```

```
        for(int i = 0; i < n1; i++){
            for(int j = 0; j < m1; j++){

              sum[i][j] = arr1[i][j] + arr2[i][j];
           }
        }

       System.err.println("matrix after Addidion : ");
        for(int i = 0; i < n1; i++){
            for(int j = 0; j < m1; j++){

           System.out.print(sum[i][j] + " " );              }
           System.out.print("\n");
        }
     }
    else
    {
        System.out.println("Dimentions are not same so addition can not
be
performed for the given array");
     }

  }
}

class Multiplication
{
   void multiplication(int arr1[][], int arr2[][])
   {
      int n1 = arr1.length;
      int m1 = arr1[0].length;
      int n2 = arr2.length;
      int m2 = arr2[0].length;

      if(m1 == n2)
```

```
        {
            int[][] mul;
            mul = new int[n1][m2];

            for(int i = 0; i < n1; i++)
            {
                for(int j = 0; j < m2; j++)
                {
                    mul[i][j] = 0;
                    for(int k = 0; k < n2; k++)
                    {
                        mul[i][j] += arr1[i][k] * arr2[k][j];
                    }
                }
            }

            System.out.println("matrix after Multiplication : ");
            for(int i = 0; i < n1; i++){
                 for(int j = 0; j < m2; j++){
                      System.out.print(mul[i][j] + " " );

                 }
                System.out.print("\n");
            }
          }
          else
          {
              System.out.println("Multiplication can not be performed for the
given
array");
        }
    }
}

class P6
```

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

      int n1, m1, n2, m2;
      System.out.print("Enter the dimention of the first array : ");
      n1 = scn.nextInt();
      m1 = scn.nextInt();

      int[][] arr1;
      arr1 = new int[n1][m1];

      System.out.print("Enter the data of the first array : ");

      for(int i = 0; i < n1; i++)
      {
          for(int j = 0; j < m1; j++)
         {
             int num = scn.nextInt();
             arr1[i][j] = num;
         }
      }

System.out.print("Enter the dimention of the second array : ");
n2 = scn.nextInt();
m2 = scn.nextInt();
int[][] arr2;

arr2 = new int[n2][m2];
System.out.print("Enter the data of the second array : ");
for(int i = 0; i < n2; i++)
{
    for(int j = 0; j < m2; j++)
    {
```

```
            int num = scn.nextInt();
            arr2[i][j] = num;
        }
    }
        Addition sum = new Addition();
        sum.addition(arr1, arr2);

        Multiplication mul = new Multiplication();
        mul.multiplication(arr1, arr2);
        }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_1> javac P6.java
PS D:\CE DDU\4TH SEM\JT\LAB_1> java P6
Enter the dimention of the first array : 3 3
Enter the data of the first array : 1 2 3 3 4 2 3 2 1
Enter the dimention of the second array : 3 3
Enter the data of the second array : 1 1 1 3 4 2 3 2 1
matrix after Addidion :
2 3 4
6 8 4
6 4 2
matrix after Multiplication :
16 15 8
21 23 13
12 13 8
PS D:\CE DDU\4TH SEM\JT\LAB_1>
```

LAB 2

AIM : String, StringBuffer, StringBuilder, array of objects, this keyword, constructor overloading

(1) Write a program that returns the number of times that the string “hi” appears anywhere in the given string.

CODE :

```
import java.util.Scanner;

class P1{
	public static void main(String[] args)
	{
		Scanner in = new Scanner(System.in);

		System.out.println("eneter the string");
		String str = in.nextLine();
		int ct = 0;

		char[] arr = new char[str.length()];
		for(int i=0;i<str.length();i++)
		{
			arr[i] = str.charAt(i);
		}

		for(int i = 0; i < arr.length; i++)
		{
			if(arr[i] == 'h')
			{
				if(arr[i+1] == 'i')
				{
					ct++;
```

```
                        }
                    }
                }

                System.out.println("the string is : " + str);
                System.out.println("the total number of count of string \"hi\" is
                                    " + ct);
        }
}
```

OUTPUT :

```
PowerShell 7.4.0
PS D:\CE DDU\4TH SEM\JT\LAB_2> javac P1.java
PS D:\CE DDU\4TH SEM\JT\LAB_2> java P1
eneter the string
hi my name is drashti bhalodiya. hi to everyone. hi oncw again
the string is : hi my name is drashti bhalodiya. hi to everyone. hi oncw again
the total number of count of string "hi" is 3
PS D:\CE DDU\4TH SEM\JT\LAB_2>
```

(2) Write a program which checks whether the input string is palindrome or not and then display an appropriate message [e.g. “Refer” is a palindrome string].

CODE :

```
import java.util.Scanner;

class P2{

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

        System.out.println("eneter the string");
        String str = in.nextLine();
```

```
        StringBuffer sb = new StringBuffer();
        sb.append(str);
        sb.reverse();
        String sstr = "";
        sstr = sb.toString();

        if(str.equals(sstr)) {
            System.out.println("given string " + str + " is palindrome");
         }
        else{
            System.out.println("given string " + str + " is not palindrome");
        }
    }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_2> javac P2.java
PS D:\CE DDU\4TH SEM\JT\LAB_2> java P2
eneter the string
madam
given string madam is palindrome
PS D:\CE DDU\4TH SEM\JT\LAB_2> java P2
eneter the string
peacock
given string peacock is not palindrome
PS D:\CE DDU\4TH SEM\JT\LAB_2>
```

(3) Write a program that takes your full name as input and displays the abbreviations of the first and middle names except the last name which is displayed as it is. For example, if your name is Robert Brett Roser, then the output should be R.B.Roser.

CODE :

```
import java.util.Scanner;
import java.util.*;

class P3{

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

        System.out.println("eneter your full name : ");
        String str = in.nextLine();

        String arr[] = str.split(" ");

        char f = arr[0].charAt(0);
        char m = arr[1].charAt(0);

        String ans = " ";
        StringBuffer sb = new StringBuffer();
        ans=sb.append(f).append('.').append(m).append('.').append(arr[2]).
                toString();

        System.out.println("full name : " + str)
        System.out.println(ans);

    }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_2> javac P3.java
PS D:\CE DDU\4TH SEM\JT\LAB_2> java P3
eneter your full name :
Drashti Chandrakantbhai Bhalodiya
full name : Drashti Chandrakantbhai Bhalodiya
D.C.Bhalodiya
PS D:\CE DDU\4TH SEM\JT\LAB_2>
```

(4) Write a method String removeWhiteSpaces(String str) method that removes all the white spaces from the string passed to the method and returns the modified string. Test the functionalities using the main() method of the Tester class.

CODE :

```
import java.util.Scanner;
import java.util.*;

class P4{

   String removeWhiteSpaces(String str)
   {
        String arr[] = str.split(" ");
        StringBuffer sb = new StringBuffer();

        for(int i = 0; i < arr.length; i++)
        {
                sb.append(arr[i]);
        }

         String ans = sb.toString();
        return ans;
   }

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

        System.out.println("enter the string");
        String str = in.nextLine();

        P4 obj = new P4();
        String ans = obj.removeWhiteSpaces(str);
```

```
        System.out.println("Original string is : " + str);
        System.out.println("After removing the white spaces : " + ans);
    }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_2> javac P4.java
PS D:\CE DDU\4TH SEM\JT\LAB_2> java P4
enter the string
My Name Is Drashti Bhalodiya.
Original string is : My Name Is Drashti Bhalodiya.
After removing the white spaces : MyNameIsDrashtiBhalodiya.
PS D:\CE DDU\4TH SEM\JT\LAB_2>
```

(5) Write a class Student with member variables int roll_no, String name and an array to store marks of 5 subjects. Demonstrate constructor overloading and use this keyword. Write a findAverage() method that returns double value. Write a TestStudent class containing main() method to do the following:

a) Store the details of one student by creating one object of Student class and display them.

b) Store the details of 3 students by creating an array of objects of Student class and display the details of the student who has the highest average amongst the three students.

CODE :

```
import java.util.Scanner;
import java.util.*;

class Student{

    int roll_no;
```

```
    String name;
    int marks[];

    Student(){}
    Student(int roll_no,String name, int[] marks)
    {
        this.roll_no = roll_no;

        this.name = new String();
        this.name = name;

        this.marks = new int[5];
         for(int i = 0; i < 5; i++)
        {
                this.marks[i] = marks[i];
        }
    }

    double findAverage()
    {
        int sum = 0;
        for(int i = 0; i < 5; i++)
        {
            sum += marks[i];
         }

         double avg = (double)sum / 5;
         return avg;
    }
}

class TestStudent{

    public static void main(String[] args)
    {
```

```
// (a)

//Scanner in = new Scanner(System.in);
//Scanner sc = new Scanner(System.in);


//int r;

//int arr[];
//arr = new int[5];

//System.out.println("enter your roll no. : ");
//r = in.nextInt();

//System.out.println("enter your name : ");
//String nm = sc.nextLine();

//System.out.println("enter your marks of 5 subjects : ");

//for(int i = 0; i < 5; i++)
//{
//    arr[i] = in.nextInt();
//}

//Student stu1 = new Student(r, nm, arr);

   //System.out.print("Roll no. : " + stu1.roll_no + "\n");
//System.out.print("name : " + stu1.name + "\n");
//System.out.print("marks of 5 subjects : " );
//for(int i = 0; i < 5; i++)
//{
//    System.out.print(stu1.marks[i] + " ");
//}
```

```
// (b)

Student stu[] = new Student[3];

Scanner in = new Scanner(System.in);
Scanner sc = new Scanner(System.in);

for(int i = 0; i < 3; i++)
{
        System.out.println("Enter the details of " + i+1 +" student :
    ");


        int r;
        int arr[];
        arr = new int[5];

        System.out.println("enter roll no. : ");
           r = in.nextInt();

        System.out.println("enter name : ");
        String nm = sc.nextLine();

        System.out.println("enter marks of 5 subjects : ");

        for(int j = 0; j < 5; j++)
        {
            arr[j] = in.nextInt();
        }

        stu[i] = new Student(r, nm, arr);
}

double average[] = new double[3];
for(int i = 0; i < 3; i++)
```

```
        {
            average[i] = stu[i].findAverage();
        }

        int index = 0;
        double max = average[0];
        for(int i = 0; i < 3; i++)
        {
            if(average[i] > max)
            {
                max = average[i];
                index = i;
            }
        }

        System.out.print("Roll no. : " + stu[index].roll_no + "\n");
        System.out.print("name : " + stu[index].name + "\n");
        System.out.print("marks of 5 subjects : " );
        for(int i = 0; i < 5; i++)
        {
            System.out.print(stu[index].marks[i] + " ");
        }
        System.out.println("\n Average is : " + max);

    }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_2> javac TestStudent.java
PS D:\CE DDU\4TH SEM\JT\LAB_2> java TestStudent
for one student :
enter your roll no. :
63
enter your name :
drashti
enter your marks of 5 subjects :
70
60
90
80
65
Roll no. : 63
name : drashti
marks of 5 subjects : 70 60 90 80 65
```

```
for three students :
Enter the details of 1 student :
enter roll no. :
1
enter name :
abc
enter marks of 5 subjects :
56
78
95
68
87
Enter the details of 2 student :
enter roll no. :
2
enter name :
pqr
enter marks of 5 subjects :
45
67
99
79
89
Enter the details of 3 student :
enter roll no. :
3
enter name :
xyz
enter marks of 5 subjects :
66
88
```

```
Enter the details of 3 student :
enter roll no. :
3
enter name :
xyz
enter marks of 5 subjects :
66
88
99
44
55
Roll no. : 1
name : abc
marks of 5 subjects : 56 78 95 68 87
Average is : 76.8
```

LAB 3

AIM : Inheritance, Polymorphism(method overriding), static keyword

1. Write a Java program that checks for prime number using the object oriented approach.
   [Hint: create a class NumberClass with a member value and method isPrimeNumber()]

CODE ;

```
public class NumberClass {

 int value;

 public NumberClass(int value) {
          super();
          this.value = value;
 }

 boolean isPrimeNumber()
 {
          if(value == 1) {
           return false;
         }
          if(value == 2) {
           return true;
          }

          for(int i = 2; i < value; i++) {

                    if(value % i == 0) {
                     return false;
           }
```

```
        }

        return true;
        }
}


import java.util.Scanner;

public class check {

        public static void main(String[] args) {

         Scanner in = new Scanner(System.in);
         int t;
         System.out.println("enter the number for check : ");
         t = in.nextInt();

         NumberClass nc = new NumberClass(t);
         boolean ans = nc.isPrimeNumber();
         if(ans) {
                System.out.println("given number is a prime number");
          }
        else {
                System.out.println("given number is not a prime number");
        }
    }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_3> javac NumberClass.java
PS D:\CE DDU\4TH SEM\JT\LAB_3> javac check.java
PS D:\CE DDU\4TH SEM\JT\LAB_3> java check
enter the number for check :
4
given number is not a prime number
PS D:\CE DDU\4TH SEM\JT\LAB_3> java check
enter the number for check :
93
given number is not a prime number
PS D:\CE DDU\4TH SEM\JT\LAB_3> java check
enter the number for check :
43
given number is a prime number
PS D:\CE DDU\4TH SEM\JT\LAB_3>
```

2. Create two classes:

class Person

Derive a class Student from class Person.

Person - name : String - age : int

+ Person()

+ Person(name : String, age : int)

+ getName() : String

+ getAge() : int

+ setName(name : String) : void

+ setAge(age : int) : void

+ toString() : String

Student - rollno : int - marks : double[]

+ Student()

+ Student(rollno : int)

+ Student(rollno : int, marks : double[])

+ Student(rollno : int, name : String, age : int, marks : double[])

+ getRollno() : int

+ getMarks() : double[]

+ setRollno(rollno: int) : void

+ setMarks(marks : double[]) : void

+ toString() : String
+ displayDetails() : void

Add the following to Student class:

- a static variable count(to count the number of objects)
- a static block to initialize count variable to zero
- a static method String getCount() that returns the number of Student objects created
- Write a TestStudent class containing the main() method.
- Store the details of 3 students by creating an array of objects of Student class and display the student who has highest average amongst the three students as follows using displayDetails() method for that object:

e.g.
RollNo = 100
Name = ABC
Age = 20
Marks=78 86 88 67 92

- Create one more object of the Student class and then call the getCount() to display the number of Student objects created.

CODE :

```
public class person {

        private int age;
        private String name;
       public person() {
         super();
        }

        public person(int age, String name) {
         super();
         this.age = age;
```

```
        this.name = name;
        }

        @Override
        public String toString() {
                return super.toString();
        }

        public int getAge() {
                return age;
        }

        public void setAge(int age) {
                this.age = age;
        }

        public String getName() {
                return name;
        }

        public void setName(String name) {
                this.name = name;
        }
}

public class student extends person {
        private int rollno;
        private double[] marks;
        static int count;

        static {
        count = 0;
        }

        static String getCount() {
```

```
 String ct = "" + count;
 return ct;
}


public student() {
 super();
 count++;
}

public student(int rollno) {
 super();
 this.rollno = rollno;
 count++;
}


public student(int rollno, double[] marks) {
        super();
        this.rollno = rollno;
        this.marks = marks;
        count++;
}

public student(int rollno, String name, int age, double[] marks) {
        super(age, name);
        this.rollno = rollno;
        this.marks = marks;
        count++;
}

public int getRollno() {
        return rollno;
}
```

```
        public void setRollno(int rollno) {
         this.rollno = rollno;
        }

        public double[] getMarks() {
         return marks;
        }

        public void setMarks(double[] marks) {
         this.marks = marks;
        }

        @Override
        public String toString() {
         //TODO Auto-generated method stub
         return super.toString();
        }

        void displayDetails() {

         System.out.println("RollNo = " + rollno);
         System.out.println("Name = " + getName());
         System.out.println("Age = " + getAge());
         System.out.print("Marks = " );
         for(int i = 0; i < marks.length; i++) {
          System.out.print(marks[i] + " ");
         }
         System.out.println();
        }
}

import java.util.Scanner;
public class TestStudent {

 public static void main(String[] args) {
```

```
  Scanner in = new Scanner(System.in);
  Scanner sc = new Scanner(System.in);

  student stu[] = new student[3];
  double sum[] = new double[3];

for(int i = 0; i < 3; i++) {

            int rollno, age;
            String name;
            double marks[] = new double[5];
            sum[i] = 0;

            System.out.print("Enter your name : ");
            name = sc.nextLine();
            //System.out.print("\n");

            System.out.print("Enter your roll no. : ");
            rollno = in.nextInt();
            //System.out.print("\n");

            System.out.print("Enter your age : ");
            age = in.nextInt();
            //System.out.print("\n");

            System.out.print("Enter your 5 subject's marks : ");
            for(int j = 0; j < 5; j++) {
             double m = in.nextDouble();
             marks[j] = m;
             sum[i] += marks[j];
}
System.out.print("\n");

 stu[i] = new student(rollno, name, age, marks);
```

```
        }

        double avg[] = new double[3];
        for(int i = 0; i < 3; i++) {
         avg[i] = sum[i] / 5;
  }

        if(avg[0] > avg[1] && avg[0] > avg[2]) {
         stu[0].displayDetails();
        }
        else if(avg[1] > avg[2]) {
         stu[1].displayDetails();
        }
        else {
         stu[2].displayDetails();
        }


        student S = new student();
        String ct = student.getCount();
        System.out.println("\n total no. of objects of students are : " + ct);
        }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_3> javac person.java
PS D:\CE DDU\4TH SEM\JT\LAB_3> javac student.java
PS D:\CE DDU\4TH SEM\JT\LAB_3> javac TestStudent.java
PS D:\CE DDU\4TH SEM\JT\LAB_3> java TestStudent
Enter your name : abc
Enter your roll no. : 11
Enter your age : 19
Enter your 5 subject's marks : 45 60 70 30 40

Enter your name : pqr
Enter your roll no. : 12
Enter your age : 19
Enter your 5 subject's marks : 60 70 80 50 90

Enter your name : xyz
Enter your roll no. : 13
Enter your age : 19
Enter your 5 subject's marks : 33 67 45 78 89

RollNo = 12
Name = pqr
Age = 19
Marks = 60.0 70.0 80.0 50.0 90.0

 total no. of objects of students are : 4
PS D:\CE DDU\4TH SEM\JT\LAB_3>
```

LAB 4

AIM : Interface, Exception Handling

(1) Write a program that catches the divide-by-zero exception using the try-catch mechanism. Take a numeric value and perform division by zero. Catch the ArithmeticException.

CODE :

```
package first;
import java.util.Scanner;
import java.io.IOException;
public class arithmetic {
	public static void main(String args[])
	{
		Scanner in = new Scanner(System.in);
		System.out.println("enter the number : ");
		int a = in.nextInt();
		int d = 0;
		double ans = 0;
		try {
			ans = a/d;
		}
		catch(ArithmeticException e)
		{
			System.out.println("Exception occurs " + e);
		}
	}
}
```

OUTPUT :

```
PowerShell 7.4.0
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001> javac arithmetic.java
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001> java arithmetic
enter the number :
45
Exception occurs java.lang.ArithmeticException: / by zero
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001>
```

(2) Write a java program using multiple catch blocks. Create a class CatchExercise, inside the try block declare an array a[] with size of 5 elements and initialize with value a[5] =30/5 . Using Multiple catch blocks handle ArithmeticException and ArrayIndexOutOfBoundsException.

CODE :

```
package first;
import java.io.IOException;

public class CatchExercise {

        public static void main(String args[])
        {
                try {
                        int arr[] = new int[5];
                        arr[5] = 30/5;
                }
                catch(ArithmeticException e)
                {
                        System.out.println("Arithmetic Exceptoin occurs");
                        System.out.println("Error is : " + e);
                }
                catch(ArrayIndexOutOfBoundsException e)
                {
                        System.out.println("Out of bounds exception occurs");
                        System.out.println("Error is : " + e);
```

```
            }
            catch(Exception e)
            {
                System.out.println("Error occurs : " + e);
            }
        }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001> javac CatchExercise.java
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001> java CatchExercise
Out of bounds exception occurs
Error is : java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001>
```

(3) Write a program that demonstrates use of finally block. Observe the output of your Program for different cases as mentioned below.

- Case A: exception does not occur. Perform 25/5 mathematical operation. Catch the NullPointerException.
- Case B: exception occurs but not handled. Perform 25/0 mathematical operation. Catch NullPointerException.
- Case C: exception occurs and handled. Perform 25/0 mathematical operation. Catch ArithmeticException

CODE :

```
package first;

public class finallyBlocks {

        static void FcaseA()
        {
            try {
                int a = 25/5;
```

```
                System.out.println("For case A : without error");
            }
        catch(NullPointerException e)
        {
                System.out.println("null pointer Exception occurs " + e);
        }
            finally {
                System.out.println("case A's finally block");
            }
    }

    static void FcaseB()
    {
            try {
                System.out.println("For case B : with arithmetic
                                exception");
                int a = 25/0;
            }
            catch(NullPointerException e)
            {
                System.out.println("null pointer exception occurs " + e);
            }
            finally {
                System.out.println("case B's finally block");
            }
    }

    static void FcaseC()
    {
        try {
        System.out.println("For case C : with arithmetic exception and
                 it also handles by catch block");
        }
        catch(ArithmeticException e)
        {
```

```
                System.out.println("arithmetic exception occurs " + e);
            }
            finally {
                System.out.println("case C's finally block");
            }
        }
        public static void main(String []args)
        {
            FcaseA();
            FcaseB();
            FcaseC();
        }
}
```

OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001> javac finallyBlocks.java
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001> java finallyBlocks
For case A : without error
case A's finally block
For case B : with arithmetic exception
case B's finally block
For case C : with arithmetic exception and it also handles by catch block
case C's finally block
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001>
```

(4) Create an interface Account with two methods: deposit and withdraw. Create class SavingsAccount which implements the interface. Write a custom Exception handler class CustomException for SavingsAccount to handle the scenarios when the withdrawn amount is larger than the balance in the account.

CODE :

```
package first;
public interface Account {
        public void deposite(int amount);
        public void withdrawn(int amount) throws CustomException;
```

```
}

package first;
import java.util.Scanner;
import java.io.IOException;
class CustomException extends Exception
{
	int x;
	public CustomException(int x) {
	super();
	this.x = x;
}
	public String toString() {
		return "Custom Exception[" + x + "] is larger than your current
				balance";
	}
}


public class SavingsAccount implements Account{
	int balance;
	public SavingsAccount(int balance) {
		super();
		this.balance = balance;
	}
	 public void deposite(int amount)
	 {
		  balance += amount;
		  System.out.println("current balance = " + balance);
	 }

	 public void withdrawn(int amount) throws CustomException
	 {
		if(amount > balance)
		{
```

```
                throw new CustomException(amount);
            }
            else
            {
                balance -= amount;
                System.out.println("current balance = " + balance);
            }
        }

        public static void main(String []args)
        {
            Scanner in = new Scanner(System.in);
            System.out.println("Enter your bank balance : ");

            int bal = in.nextInt();
            SavingsAccount sa = new SavingsAccount(bal);

            System.out.println("enter the amount you want deposite : ");
            int debal = in.nextInt();
            sa.deposite(debal);

            System.out.println("enter the amount you want withdraw : ");
            int wibal = in.nextInt();
            try {
                sa.withdrawn(wibal);
            }
            catch(CustomException e)
            {
                System.out.println("Error occurs : " + e);
            }

        }
    }
```

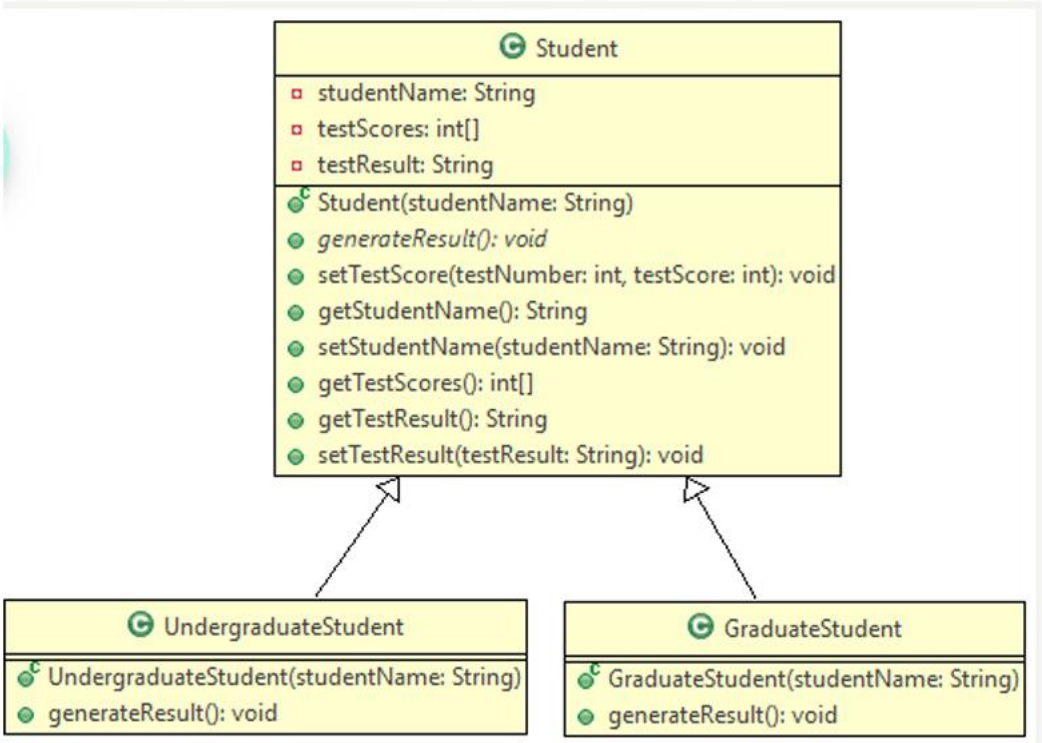
OUTPUT :

```
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001> javac SavingsAccount.java
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001> java SavingsAccount
Enter your bank balance :
30000
enter the amount you want deposite :
4500
current balance = 34500
enter the amount you want withdraw :
50000
Error occurs : Custom Exception[50000] is larger than your current balance
PS D:\CE DDU\4TH SEM\JT\LAB_4\drive-download-20240103T174253Z-001>
```

LAB 5

AIM : Abstract class , Interface , Multithreading

(1) Anchor College offers both UnderGraduate and PostGraduate programs. The college stores the names of the students, their test scores and the final result for each student. Each student has to take 4 tests in total. You need to create an application for the college by implementing the classes based on the class diagram and description given below.



Implement the getter and setter methods appropriately.

1. Student(Class)

   I. Student(String studentName)

   - Initialize the instance variable studentName with the value passed to the constructor and other instance variables to the default values.

II. setTestScore(int testNumber, int testScore)

- Set the value of the testScore in the appropriate position of testScores array based on the testNumber.

2. UndergraduateStudent(Class)

I. UndergraduateStudent(String studentName)

- Initialize the instance variable studentName with the value passed to the constructor and other instance variables to the default values.

II. generateResult()

- Implement the abstract method of Student class by setting the value of testResult based on the below details.

Average Score	Result
>=60	Pass
<60	Fail

Sample Input and Output For UndergraduateStudent

Input:-

Instance Variable	Values
name	**Jerry**
testScores	**{70,69,71,55}**

Output:-

Student Name : Jerry

Result : Pass

3. PostGraduateStudent(Class)

I. PostgraduateStudent(String studentName)

- Initialize the instance variable studentName with the value passed to the constructor and other instance variables to the default values.

II. generateResult()

- Implement the abstract method of Student class by setting the value of testResult based on the below details.

Average Score	Result
>=75	Pass
<75	Fail

Sample Input and Output For PostUndergraduateStudent

Input:-

Instance Variable	Values
name	Tom
testScores	{70,75,80,85}

Output:- **Student Name :** Tom

Result : Pass

CODE :

```
public abstract class Student {


        private String studentName;
        private int testScores[];
        private String testResult;

        public Student(String studentName)
        {
                super();
```

```
        this.studentName = studentName;
        testScores = new int[4];
}

public abstract void generateResult();

public void setTestScore(int testNumber, int testScore)
{
        this.testScores[testNumber] = testScore;
}

public String getStudentName()
{
        return this.studentName;
}

public void setStudentName(String studentName)
{
        this.studentName = studentName;
}

        public int[] getTestScores()
{
 return this.testScores;
}

public String getTestResult()
{
        return this.testResult;
}

public void setTestResult(String testResult)
{
        this.testResult = testResult;
}
```

```
}
```

➢ UnderGraduate Student

```
import java.util.*;

public class UndergraduateStudent extends Student{

        public UndergraduateStudent(String studentName)
        {
                super(studentName);
        }

        public void generateResult()
        {
                int sum = 0;
                int scores[] = super.getTestScores();
                for(int i = 0; i < 4; i++)
                 {
                        sum += scores[i];
                }
                  double avg = (double)sum / 4;

                  if(avg >= 60.0)
                  {
                        super.setTestResult("Pass");
                  }
                  else
                  {
                        super.setTestResult("Fail");
                  }
        }

        public static void main(String[] args)
        {
```

```
        Scanner in = new Scanner(System.in);

        System.out.println("Enter your name : ");
        String name = in.nextLine();

        UndergraduateStudent UGS = new
       UndergraduateStudent(name);

        System.out.println("Enter your marks of 4 subjects : ");

        for(int i = 0; i < 4; i++)
        {
                int x = in.nextInt();
                UGS.setTestScore(i, x);
        }

        UGS.generateResult();

        System.out.print("Student Name : " + UGS.getStudentName()
                + '\n');
        System.out.println("Result : " + UGS.getTestResult());

    }

}
```

OUTPUT :

```
PS C:\Users\Drashti Bhalodiya\Downloads\ce063 (1)-20240113T034953Z-001\ce063 (1)\LAB_5\src> javac UndergraduateStudent.java
PS C:\Users\Drashti Bhalodiya\Downloads\ce063 (1)-20240113T034953Z-001\ce063 (1)\LAB_5\src> java UndergraduateStudent
Enter your name :
Drashti Bhalodiya
Enter your marks of 4 subjects :
70 80 90 80
Student Name : Drashti Bhalodiya
Result : Pass
```

➢ PostGraduate Student

```
import java.util.Scanner;

public class GraduateStudent extends Student{

        public GraduateStudent(String studentName) {
                super(studentName);
         }

         public void generateResult()
         {
                int sum = 0;
                int scores[] = super.getTestScores();
                for(int i = 0; i < 4; i++)
                {
                      sum += scores[i];
                }
                double avg = (double)sum / 4;

                if(avg >= 75.0)
                {
                      super.setTestResult("Pass");
                }
                else
                {
                      super.setTestResult("Fail");
                }
        }

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

                System.out.println("Enter your name : ");
                String name = in.nextLine();
```

```
            GraduateStudent GS = new GraduateStudent(name);

          System.out.println("Enter your marks of 4 subjects : ");

            for(int i = 0; i < 4; i++)
            {
                    int x = in.nextInt();
                    GS.setTestScore(i, x);
            }

        GS.generateResult();

        System.out.print("Student Name : " + GS.getStudentName() + '\n');
        System.out.println("Result : " + GS.getTestResult());

      }
}
```

OUTPUT :

```
PS C:\Users\Drashti Bhalodiya\Downloads\ce063 (1)-20240113T034953Z-001\ce063 (1)\LAB_5\src> javac GraduateStudent.java
PS C:\Users\Drashti Bhalodiya\Downloads\ce063 (1)-20240113T034953Z-001\ce063 (1)\LAB_5\src> java GraduateStudent
Enter your name :
ABCD
Enter your marks of 4 subjects :
50 40 60 70
Student Name : ABCD
Result : Fail
```

(2) Write a Java program as per the given description to demonstrate use of interface.

I. Define an interface RelationInterface.
   Write three abstract methods: isGreater, isLess and isEqual.
   All methods have a return type of boolean and take an argument of type Line with which the caller object will be compared.

II. Define the Line class implements the RelationInterface interface.
   - It has 4 double variables for the x and y coordinates of the line.
   - Define a constructor in Line class that initializes these 4 variables.

- Define a method getLength() that computes length of the line.
  [double length = Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1))].
- Implement the methods of interface in Line class

III. In class CompareLines.Java, create two objects of Line class, call the three methods to compare the lengths of the lines.

CODE:

```
public interface RelationInterface
{

        public boolean isGreater(Line obj);
        public boolean isLess(Line obj);
        public boolean isEqual(Line obj);
}


public class Line implements RelationInterface
{

        private double x1, x2, y1, y2;

        public Line(double x1, double x2, double y1, double y2)
        {
                super();
                this.x1 = x1;
                this.x2 = x2;
                this.y1 = y1;
                this.y2 = y2;
         }

         public double getLength()
         {
          double length = Math.sqrt((x2 - x1)*(x2 - x1) + (y2 - y1)*(y2-y1));
          return length;
```

```
  }

  public boolean isGreater(Line obj)
{
        double len1 = this.getLength();
        double len2 = obj.getLength();

        if(len1 > len2)
        {
            return true;
        }
        else
        {
            return false;
        }
}

  public boolean isLess(Line obj)
  {
   double len1 = this.getLength();
   double len2 = obj.getLength();

   if(len1 < len2)
   {
    return true;
   }
   else {
    return false;
   }
  }

  public boolean isEqual(Line obj)
  {
        double len1 = this.getLength();
        double len2 = obj.getLength();
```

```
        if(len1 == len2)
        {
                return true;
            }
            else {
                return false;
            }
    }
}
```

OUTPUT :

```
PS C:\Users\Drashti Bhalodiya\Downloads\ce063 (1)-20240113T034953Z-001\ce063 (1)\LAB_5\src> javac CompareLines.java
PS C:\Users\Drashti Bhalodiya\Downloads\ce063 (1)-20240113T034953Z-001\ce063 (1)\LAB_5\src> java CompareLines
Enter dimention for line 1 : (x1, y1)(x2, y2)
0 2 0 6
Enter dimention for line 2 : (x1, y1)(x2, y2)
0 0 0 8
Line 1 is smaller than Line 2
```

(3) In the producer–consumer problem, the producer and the consumer share a common, fixed-size buffer used as a queue. (Take buffer size as 1). The producer's job is to generate data, put it into the buffer. At the same time, the consumer is consuming the data (i.e. removing it from the buffer).The problem is to make sure that the producer won't try to add data into the buffer if it's full and that the c consumer won't try to remove data from an empty buffer. Write a Java application consisting of all necessary classes to achieve this.

CODE:

```
package threads.interthreadcommunication;
// A correct implementation of a producer and consumer.
class Q1 {
    int n;
    boolean valueSet = false;
```

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

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

class Producer1 implements Runnable {
        Q1 q;
        Producer1(Q1 q) {
        this.q = q;
        new Thread(this, "Producer").start();
}
```

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

class Consumer1 implements Runnable {
        Q1 q;
        Consumer1(Q1 q) {
                this.q = q;
                new Thread(this, "Consumer").start();
        }
        public void run() {
                while (true) {
                        q.get();
                }
        }
}

public class ProducerConsumer {
        public static void main(String args[]) {
                Q1 q = new Q1();
                new Producer1(q);
                new Consumer1(q);
                System.out.println("Terminate the process to stop.");
        }
}


import java.util.*;

public class CompareLines
{
```

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

        System .out.println("Enter dimention for line 1 : (x1, y1)(x2,
                    y2)");
        double x1 = in.nextDouble();
        double y1 = in.nextDouble();
        double x2 = in.nextDouble();
        double y2 = in.nextDouble();
        Line obj1 = new Line(x1, x2, y1, y2);

        System.out.println("Enter dimention for line 2 : (x1, y1)(x2,
      y2)");
        x1 = in.nextDouble();
        y1 = in.nextDouble();
        x2 = in.nextDouble();
        y2 = in.nextDouble();

        Line obj2 = new Line(x1, x2, y1, y2);

        boolean greater = obj1.isGreater(obj2);
        boolean less = obj1.isLess(obj2);
        boolean equal = obj1.isEqual(obj2);

        if(greater)
        {
            System.out.println("Line 1 is greater than Line 2");
        }
        else if(less)
        {
            System.out.println("Line 1 is smaller than Line 2");
        }
        else if(equal)
```

```
            {
                System.out.println("Line 1 and Line 2 are equal");
            }
        }
}
```

OUTPUT :

```
Got: 151263
Put: 151264
Got: 151264
Put: 151265
Got: 151265
Put: 151266
Got: 151266
Put: 151267
Got: 151267
Put: 151268
Got: 151268
Put: 151269
Got: 151269
Put: 151270
Got: 151270
Put: 151271
Got: 151271
Put: 151272
Got: 151272
Put: 151273
Got: 151273
Put: 151274
Got: 151274
Put: 151275
Got: 151275
Put: 151276
Got: 151276
Put: 151277
Got: 151277
Put: 151278
```

(4) Write a multithreaded Java application to produce a deadlock condition.

CODE:

```
package threads.interthreadcommunication;

class A {

        synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
              Thread.sleep(1000);
        } catch (InterruptedException e) {
              System.out.println("A Interrupted");
        }
                 System.out.println(name + " trying to call B.last()");
                 b.last();
        }

       synchronized void last() {
              System.out.println("Inside A.last");
      }
}

class B {

       synchronized void bar(A a) {
       String name = Thread.currentThread().getName();
       System.out.println(name + " entered B.bar");
       try {
              Thread.sleep(1000);
        } catch (InterruptedException e) {
             System.out.println("B Interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
  }
```

```
        synchronized void last() {
            System.out.println("Inside B.last");
        }
    }

    public class Deadlock implements Runnable {

        A a = new A();
        B b = new B();

        Deadlock() {
            Thread.currentThread().setName("MainThread");
            Thread t = new Thread(this, "RacingThread");
            t.start();
        a.foo(b); // get lock on a in this thread.
        System.out.println("Back in main thread");
        }
        @Override
        public void run() {
            b.bar(a); // get lock on b in other thread.
            System.out.println("Back in other thread");
        }
            public static void main(String args[]) {
            new Deadlock();
        }
}
```

OUTPUT:

```
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
```

LAB 6

AIM: JDBC, Generics

(1) Write a Java application to perform operations for student information like (id[Primary key, Auto increment],firstName, lastName, branch, username and password) from a database using JDBC.

- Insert two records for student
- Practice the use of the following methods of the ResultSet interface: absolute(), afterLast(), beforeFirst(), first(), isFirst(), isLast(), last(), previous(), next(), relative().

CODE :

```
package lab;
import java.sql.*;


public class Student
{
        public static void main(String args[])
        {

                try (Connection con =
                        DriverManager.getConnection("jdbc:mysql://lo
                                                calhost:3306/lab_db","root", ""))
                {
                  Statement s;

                  int resultsetType = ResultSet.TYPE_SCROLL_SENSITIVE;
                  int resultsetConcurrency = ResultSet.CONCUR_READ_ONLY;
```

```
        s = con.createStatement(resultsetType,
      resultsetConcurrency);


 s.execute(createTable);

 String insertQuery = "INSERT INTO `Student_info`(`FirstName`, `Last
                Name`, `Branch`, `UserName`, `Password`) VALUES(
               'Drashti', 'Bhalodiya', 'CE' , '22CEUOS133',
                '27/09/2004')";
s.executeUpdate(insertQuery);
 insertQuery = "INSERT INTO `Student_info`(`FirstName`,
      `LastName`, `Branch`, `UserName`, `Password`) VALUES(
      'Deep',  'Bhalodiya', 'BCA' , '22CEUOS101', '27/09/2004')";

 s.executeUpdate(insertQuery);

insertQuery = "INSERT INTO `Student_info`(`FirstName`,
      `LastName`,  `Branch`, `UserName`, `Password`) VALUES(
      'Yash', 'Bhalodiya', 'Diploma' , '22CEUOS111', '28/10/2003')";

 s.executeUpdate(insertQuery);
 insertQuery = "INSERT INTO `Student_info`(`FirstName`,
      `LastName`,  `Branch`, `UserName`, `Password`) VALUES(
      'Vrunda','Bhalodiya', '9th' , '22CEUOS121', '05/07/2011')";

 s.executeUpdate(insertQuery);


 String selectQuery = "select * from `Student_info`";
 ResultSet rs;
 rs = s.executeQuery(selectQuery);
 while (rs.next())
 {
      System.out.println(".......................");
```

```
            System.out.println("ID: " + rs.getInt(1));
            System.out.println("FirstName: " + rs.getString(2));
            System.out.println("LastName: " + rs.getString(3));
            System.out.println("Branch: " + rs.getString(4));
            System.out.println("UserName: " + rs.getString(5));
            System.out.println("Password: " + rs.getString(6));
}

          //Practice the use of the following methods of the ResultSet
        interface:  absolute(), afterLast(), beforeFirst(), first(),
          //isFirst(), isLast(), last(), previous(), next(), relative().

          // absolute()
          System.out.println();
          rs.absolute(3);
          System.out.println(".......................");
          System.out.println("at index : "+ 3);
          System.out.println("ID: " + rs.getInt(1));
          System.out.println("FirstName: " + rs.getString(2));
          System.out.println("LastName: " + rs.getString(3));
          System.out.println("Branch: " + rs.getString(4));
          System.out.println("UserName: " + rs.getString(5));
          System.out.println("Password: " + rs.getString(6));

          // afterLast() previous()
          rs.afterLast();
          System.out.println();
          System.out.println(".......................");
          System.out.println("after Last ");
          while(rs.previous())
          {
            System.out.println(".......................");
            System.out.println("ID: " + rs.getInt(1));
            System.out.println("FirstName: " + rs.getString(2));
            System.out.println("LastName: " + rs.getString(3));
```

```
 System.out.println("Branch: " + rs.getString(4));
 System.out.println("UserName: " + rs.getString(5));
 System.out.println("Password: " + rs.getString(6));
}

//beforeFirst() next()
rs.beforeFirst();
System.out.println();
System.out.println(".......................");
System.out.println("before First ");
while(rs.next())
{
 System.out.println(".......................");
 System.out.println("ID: " + rs.getInt(1));
 System.out.println("FirstName: " + rs.getString(2));
 System.out.println("LastName: " + rs.getString(3));
 System.out.println("Branch: " + rs.getString(4));
 System.out.println("UserName: " + rs.getString(5));
 System.out.println("Password: " + rs.getString(6));
}

// first()
rs.first();
System.out.println();
System.out.println(".......................");
System.out.println("first");
System.out.println(".......................");
System.out.println("ID: " + rs.getInt(1));
System.out.println("FirstName: " + rs.getString(2));
System.out.println("LastName: " + rs.getString(3));
System.out.println("Branch: " + rs.getString(4));
System.out.println("UserName: " + rs.getString(5));
System.out.println("Password: " + rs.getString(6));

//rs.last();
```

```
//isFirst()
System.out.println();
System.out.println("......................");
System.out.println("is first");
System.out.println("......................");
boolean first = rs.isFirst();
if(first)
{
 System.out.println("pointer is at first position");
}
else
{
 System.out.println("pointer is not at first position");
}

//rs.first();
//isLast()
System.out.println();
System.out.println("......................");
System.out.println("is last");
System.out.println("......................");
boolean last = rs.isLast();
if(last)
{
 System.out.println("pointer is at last position");
}
else
{
 System.out.println("pointer is not at last position");
}

//last()
rs.last();
System.out.println();
System.out.println("......................");
```

```
System.out.println("last");
System.out.println(".....");
System.out.println("ID: " + rs.getInt(1));
System.out.println("FirstName: " + rs.getString(2));
System.out.println("LastName: " + rs.getString(3));
System.out.println("Branch: " + rs.getString(4));
System.out.println("UserName: " + rs.getString(5));
System.out.println("Password: " + rs.getString(6));

//absolute()
rs.absolute(-3);
System.out.println();
System.out.println(".....");
System.out.println("absolute(-3)");
System.out.println(".....");
System.out.println("ID: " + rs.getInt(1));
System.out.println("FirstName: " + rs.getString(2));
System.out.println("LastName: " + rs.getString(3));
System.out.println("Branch: " + rs.getString(4));
System.out.println("UserName: " + rs.getString(5));
System.out.println("Password: " + rs.getString(6));


//relative()
rs.relative(0);
System.out.println();
System.out.println(".....");
System.out.println("relative");
System.out.println(".....");
System.out.println("ID: " + rs.getInt(1));
System.out.println("FirstName: " + rs.getString(2));
System.out.println("LastName: " + rs.getString(3));
System.out.println("Branch: " + rs.getString(4));
System.out.println("UserName: " + rs.getString(5));
System.out.println("Password: " + rs.getString(6));
```

```
            }
            catch (SQLException e)
            {
             System.out.println(e);
            }

        }
}
```

OUTPUT :

```
.....
ID: 1
FirstName: Drashti
LastName: Bhalodiya
Branch: CE
UserName: 22CEUOS133
Password: 27/09/2004
.....
ID: 2
FirstName: Deep
LastName: Bhalodiya
Branch: BCA
UserName: 22CEUOS101
Password: 27/09/2004
.....
ID: 3
FirstName: Yash
LastName: Bhalodiya
Branch: Diploma
UserName: 22CEUOS111
Password: 28/10/2003
.....
ID: 4
```

FirstName: Vrunda
LastName: Bhalodiya
Branch: 9th
UserName: 22CEUOS121
Password: 05/07/2011

.....
at index : 3
ID: 3
FirstName: Yash
LastName: Bhalodiya
Branch: Diploma
UserName: 22CEUOS111
Password: 28/10/2003
.....
after Last
.....
ID: 4
FirstName: Vrunda
LastName: Bhalodiya
Branch: 9th
UserName: 22CEUOS121
Password: 05/07/2011
.....
ID: 3
FirstName: Yash
LastName: Bhalodiya
Branch: Diploma
UserName: 22CEUOS111
Password: 28/10/2003
.....
ID: 2
FirstName: Deep
LastName: Bhalodiya
Branch: BCA

UserName: 22CEUOS101
Password: 27/09/2004
.....
ID: 1
FirstName: Drashti
LastName: Bhalodiya
Branch: CE
UserName: 22CEUOS133
Password: 27/09/2004
.....
before First
.....
ID: 1
FirstName: Drashti
LastName: Bhalodiya
Branch: CE
UserName: 22CEUOS133
Password: 27/09/2004
.....
ID: 2
FirstName: Deep
LastName: Bhalodiya
Branch: BCA
UserName: 22CEUOS101
Password: 27/09/2004
.....
ID: 3
FirstName: Yash
LastName: Bhalodiya
Branch: Diploma
UserName: 22CEUOS111
Password: 28/10/2003
.....
ID: 4
FirstName: Vrunda

LastName: Bhalodiya
Branch: 9th
UserName: 22CEUOS121
Password: 05/07/2011

.....
first
.....
ID: 1
FirstName: Drashti
LastName: Bhalodiya
Branch: CE
UserName: 22CEUOS133
Password: 27/09/2004

.....
is first
.....
pointer is at first position

.....
is last
.....
pointer is not at last position

.....
Last
.....
ID: 4
FirstName: Vrunda
LastName: Bhalodiya
Branch: 9th
UserName: 22CEUOS121
Password: 05/07/2011

.....
absolute(-3)
.....
ID: 2
FirstName: Deep
LastName: Bhalodiya
Branch: BCA
UserName: 22CEUOS101
Password: 27/09/2004
.....
Relative
.....
ID: 2
FirstName: Deep
LastName: Bhalodiya
Branch: BCA
UserName: 22CEUOS101
Password: 27/09/2004

(2) Using JDBC API and MySql database perform the following operations.

- create a table MOVIES with following columns in the database:
  Id of type INTEGER AUTO INCREMENT,
  Title of type VARCHAR (50),
  Genre of type VARCHAR (50),
  YearOfRelease of type INTEGER.
- Define Movie class with following data members
  private Integer id;
  private String title;
  private String genre;
  private Integer yearOfRelease;

Create getters and setters for the mentioned data members.

- Define following methods in a class, test the methods according to user input

➢ createMovie(Movie m)- it will insert a new record for a movie.

➢ deleteMovie(int MovieID)- it will delete a movie with given MovieID

➢ updateMovieTitle(String title, int id)- it will update the title of a movie with given id.

➢ findMovieById(int MovieId)- it will display all details of a movie with a given MovieId

➢ findAllMovie()- it will display all details of all movies

CODE:

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

public class checkMovie {

 static void createMovie(Movie m)
 {
        try(Connection conn = DriverManager.getConnection("jdbc:mysql://lo
                                                    calhost:3306/lab_db","root",""))
         {
           String query = "INSERT INTO `movies` VALUES (?,?,?,?)";
           PreparedStatement s = conn.prepareStatement(query);
           s.setInt(1, m.getId());
           s.setString(2, m.getTitle());
           s.setString(3, m.getGenre());
           s.setInt(4,  m.getYearOfRelease());
           s.executeUpdate();
           System.out.println("----------------------------------");
           System.out.println("movie inserted succesfully");
           System.out.println("----------------------------------");
          }
          catch(SQLException e)
          {
```

```
                System.out.println(e);
        }
}

static void deleteMovie(int MovieID)
{
        try(Connection conn = DriverManager.getConnection("jdbc:mysql://lo
                calhost:3306/lab_db","root",""))
        {
          String query = "DELETE FROM `movies` WHERE `Id` = " + MovieID ;
          Statement s = conn.createStatement();
          s.executeUpdate(query);
          System.out.println("----------------------------------");
          System.out.println("movie deleted succesfully");
          System.out.println("----------------------------------");
         }
         catch(SQLException e)
         {
                System.out.println(e);
         }
}

static void updateMovieTitle(String title, int id)
{
        try(Connection conn = DriverManager.getConnection("jdbc:mysql://lo
                                        calhost:3306/lab_db","root",""))
        {
          String query = "UPDATE `movies` SET `Title` = ' "+ title + " ' WHERE
                                        `Id` =  " + id ;
          Statement s = conn.createStatement();
          s.executeUpdate(query);
          System.out.println("----------------------------------");
          System.out.println("movie updates succesfully");
          System.out.println("----------------------------------");
        }
```

```
        catch(SQLException e)
        {
            System.out.println(e);
        }
       }

static void findMovieById(int MovieId)
{
    try(Connection conn = DriverManager.getConnection("jdbc:mysql://lo
                                        calhost:3306/lab_db","root",""))
      {
        String query = "SELECT * FROM `movies`";
        Statement s;
        int resultsetType = ResultSet.TYPE_SCROLL_SENSITIVE;
        int resultsetConcurrency = ResultSet.CONCUR_READ_ONLY;
        s = conn.createStatement(resultsetType, resultsetConcurrency);
        ResultSet rs = s.executeQuery(query);
        rs.absolute(MovieId);
        System.out.println("movie details of id: " + MovieId);
        System.out.println("----------------------------------");
        System.out.println("Id : " + rs.getInt(1));
        System.out.println("Title : " + rs.getString(2));
        System.out.println("Genre : " + rs.getString(3));
        System.out.println("Year : " + rs.getInt(4));
        System.out.println("----------------------------------");
        }
        catch(SQLException e)
        {
            System.out.println(e);
        }
}

static void findAllMovie()
{
     try(Connection conn = DriverManager.getConnection("jdbc:mysql://lo
```

```
                                        calhost:3306/lab_db","root",""))
        {
                String query = "SELECT * FROM `movies`";
                Statement s = conn.createStatement();
                ResultSet rs = s.executeQuery(query);
                System.out.println("----------------------------------");
                System.out.println("details of all movies : ");

                while(rs.next())
                {
                 System.out.println("----------------------------------");
                 System.out.println("Id : " + rs.getInt(1));
                 System.out.println("Title : " + rs.getString(2));
                 System.out.println("Genre : " + rs.getString(3));
                 System.out.println("Year : " + rs.getInt(4));
                 System.out.println("----------------------------------");
                }
        }
         catch(SQLException e)
         {
                System.out.println(e);
         }
}



public static void main(String[] args)
{
        try(Connection con = DriverManager.getConnection("jdbc:mysql://lo
                                        calhost:3306/lab_db","root", ""))
        {
                Statement s;
                s = con.createStatement();
                  String createQuery = "CREATE TABLE movies("
                   + "Id int AUTO_INCREMENT PRIMARY KEY,"
```

```
  + "Title VARCHAR(50),"
  + "Genre VARCHAR(50),"
  + "YearOfRelease int );";


s.execute(createQuery);

Scanner in = new Scanner(System.in);
int id;
String title;
String genre;
int year;
int ct = 0;
String act;
boolean flag = true;
System.out.println("I : insert movie" + '\n' +
          "D : delete movie" + '\n' +
          "U : update movie" + '\n' +
          "F : find perticular movie" + '\n' +
           "A : all movies" + '\n' +
          "E : exit");

while(flag)
{
         System.out.println("enter your choice : ");
         act = in.nextLine();

          switch(act)
          {
                  case "I":
                      ct++;
                      System.out.println("enter title : ");
                      title = in.nextLine();
                      System.out.println("enter genre : ");
                      genre = in.nextLine();
```

```
                System.out.println("enter year : ");
                year = in.nextInt();
                in.nextLine();
                Movie m = new Movie(ct, title, genre, year);
                checkMovie.createMovie(m);
                break;
            case "D":
                System.out.println("enter id : ");
                id = in.nextInt();
                in.nextLine();
                checkMovie.deleteMovie(id);
                break;
             case "U":
                System.out.println("enter title : ");
                title = in.nextLine();
                System.out.println("enter id : ");
                id = in.nextInt();
                in.nextLine();
                checkMovie.updateMovieTitle(title, id);
                break;
            case "F":
                System.out.println("enter id : ");
                id = in.nextInt();
                in.nextLine();
                checkMovie.findMovieById(id);
                break;
            case "A":
                checkMovie.findAllMovie();
                break;
            case "E":
                flag = false;
                break;
            default :
        System.out.println("Enter valid action I, D, U, F, A,
                                                                E");
```

```
                                break;
                            }
                        }

                }
                catch(SQLException e)
                {
                        System.out.println(e);
                }
        }
}
```

OUTPUT :

```
table succesfully created
I : insert movie
D : delete movie
U : update movie
F : find perticular movie
A : all movies
E : exit
enter your choice :
I
enter title :
Kesari
enter genre :
Patriotic
enter year :
2019
----------------------------------
movie inserted succesfully
----------------------------------
enter your choice :
I
```

```
enter title :
Animal
enter genre :
Action
enter year :
2023
 ----------------------------------
movie inserted successfully
 ----------------------------------
enter your choice :
I
enter title :
Kantara
enter genre :
Religious
enter year :
2022
----------------------------------
movie inserted successfully
----------------------------------
enter your choice :
I
enter title :
Sholay
enter genre :
Comedy
enter year :
2022
----------------------------------
movie inserted successfully
 ----------------------------------
enter your choice :
I
enter title :
Now You See Me
```

```
enter genre :
Crime
enter year :
2013
 ----------------------------------
movie inserted succesfully
----------------------------------
enter your choice :
I
enter title :
Anupama
enter genre :
Serial
enter year :
2017
----------------------------------
movie inserted succesfully
----------------------------------
enter your choice :
I
enter title :
The Imitation Game
enter genre :
War
enter year :
2014
----------------------------------
movie inserted successfully
 ----------------------------------
enter your choice :
A
-----------------------------------
details of all movies :
 -----------------------------------
Id : 1
```

Title : Kesari
Genre : Patriotic
Year : 2019

Id : 2
Title : Animal
Genre : Action
Year : 2023

Id : 3
Title : Kantara
Genre : Religious
Year : 2022

Id : 4
Title : Sholay
Genre : Comedy
Year : 2022

Id : 5
Title : Now You See Me
Genre : Crime
Year : 2013

Id : 6
Title : Anupama
Genre : Serial
Year : 2017


```
Id : 7
Title : The Imitation Game
Genre : War
Year : 2014
 -----------------------------------
enter your choice :
U
enter title :
Thank God
enter id :
4
-----------------------------------
movie updates succesfully
-----------------------------------
enter your choice :
D
enter id :
6
-----------------------------------
movie deleted successfully
 -----------------------------------
enter your choice :
A
-----------------------------------
details of all movies :
-----------------------------------
Id : 1
Title : Kesari
Genre : Patriotic
Year : 2019
 -----------------------------------
-----------------------------------
Id : 2
Title : Animal
Genre : Action
```

```
Year : 2023
-----------------------------------
-----------------------------------
Id : 3
Title : Kantara
Genre : Religious
Year : 2022
-----------------------------------
-----------------------------------
Id : 4
Title :  Thank God
Genre : Comedy
Year : 2022
-----------------------------------

-----------------------------------
Id : 5
Title : Now You See Me
Genre : Crime
Year : 2013
 -----------------------------------
-----------------------------------
Id : 7
Title : The Imitation Game
Genre : War
Year : 2014
-----------------------------------
enter your choice :
R
Enter valid action I, D, U, F, A, E
enter your choice :
I
enter title :
3 Idiots
enter genre :
```

```
Comedy
enter year :
2009
----------------------------------
movie inserted succesfully
----------------------------------
enter your choice :
A
------------------------------------
details of all movies :
 -------------------------------------
Id : 1
Title : Kesari
Genre : Patriotic
Year : 2019
------------------------------------
------------------------------------
Id : 2
Title : Animal
Genre : Action
Year : 2023
------------------------------------
------------------------------------
Id : 3
Title : Kantara
Genre : Religious
Year : 2022
------------------------------------
------------------------------------
Id : 4
Title :  Thank God
Genre : Comedy
Year : 2022
------------------------------------
------------------------------------
```

```
Id : 5
Title : Now You See Me
Genre : Crime
Year : 2013
-----------------------------------
-----------------------------------
Id : 7
Title : The Imitation Game
Genre : War
Year : 2014
-----------------------------------
-----------------------------------
Id : 8
Title : 3 Idiots
Genre : Comedy
Year : 2009
-----------------------------------
enter your choice :
E
```

(3) Create a Generic class Calculator which can perform addition, subtraction, multiplication and division. Make sure that Calculator class works for Numeric values only. Write an appropriate main method in TestCalculator class.

CODE:

```
package lab;
import java.util.*;

public class GenericMethod<T> {

  public static <T> boolean CheckArray(T arr1[], T arr2[], int n1, int n2)
```

```
{
        if(n1 == n2)
        {
            for(int i = 0; i < n1; i++)
            {
                if(!arr1[i].equals(arr2[i]))
                {
                    return false;
                }
            }
            return true;
        }
        else
        {
            return false;
        }
}

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

        // for array 1
        System.out.println("Enter the length of first array : ");
        int n1 = in.nextInt();
        Integer arr1[] = new Integer[n1];
        System.out.println("Enter the elements of first array(integer) : ");
        for(int i = 0; i < n1; i++)
        {
            arr1[i] = in.nextInt();
        }

        String arr1S[] = new String[n1];
        System.out.println("Enter the elements of first array(String) : ");
        for(int i = 0; i < n1; i++)
```

```
  {
        arr1S[i] = in.next();
  }


  // for array 2
  System.out.println("Enter the length of second array : ");
  int n2 = in.nextInt();
  Integer arr2[] = new Integer[n2];
  for(int i = 0; i < n2; i++)
  {
        arr2[i] = in.nextInt();
  }

  String arr2S[] = new String[n2];
  System.out.println("Enter the elements of first array(String) : ");
  for(int i = 0; i < n2; i++)
  {
        arr2S[i] = in.next();
  }


  boolean ansInt = GenericMethod.CheckArray(arr1, arr2, n1, n2);
  boolean ansStr = GenericMethod.CheckArray(arr1S, arr2S, n1, n2);

  if(ansInt)
  {
        System.out.println("both integer arrays have same elements
                                          and in the same order");
}
else
  {
        System.out.println("both arrays are not same");
  }
```

```
            if(ansStr)
            {
                System.out.println("both String arrays have same elements and
                                                        in the same order");
            }
            else
            {
                System.out.println("both String arrays are not same");
            }

            in.close();
        }
}
```

OUTPUT :

```
Enter the length of first array :
6
Enter the elements of first array(integer) :
2 3 4 1 2 3
Enter the elements of first array(String) :
abc def ghi jkl mno pqr
Enter the length of second array :
6
2 3 4 1 2 3
Enter the elements of first array(String) :
ab cd efg hi jkl mn
both integer arrays have same elements and in the same order
both String arrays are not same
```

(4) Write a Java program to create a generic method that takes two

arrays of T type and checks if they have the same elements in the same order.

CODE:

```
package lab;
import java.util.*;

class Calculator<T extends Number> {

   Double Addition(T num1, T num2) {
      return num1.doubleValue() + num2.doubleValue();
   }

   Double Subtraction(T num1, T num2) {
      return num1.doubleValue() - num2.doubleValue();
   }

   Double Multiplication(T num1, T num2) {
      return num1.doubleValue() * num2.doubleValue();
   }

   Double Division(T num1, T num2) {
      return num1.doubleValue() / num2.doubleValue();
   }
}

public class TestCalculator {

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

      // integer type
      Calculator<Integer> objInt = new Calculator<>();
```

```
Integer a, b;

System.out.println("Enter the value of a and b for Integer type: ");
a = in.nextInt();
b = in.nextInt();

System.out.println("For integer values: ");
System.out.println("Addition = " + objInt.Addition(a, b));
System.out.println("Subtraction = " + objInt.Subtraction(a, b));
System.out.println("Multiplication = " + objInt.Multiplication(a, b));
System.out.println("Division = " + objInt.Division(a, b));

// double type
Calculator<Double> objDouble = new Calculator<>();

Double c, d;

System.out.println("Enter the value of c and d for Double type: ");
c = in.nextDouble();
d = in.nextDouble();

System.out.println("For Double values: ");
System.out.println("Addition = " + objDouble.Addition(c, d));
System.out.println("Subtraction = " + objDouble.Subtraction(c, d));
System.out.println("Multiplication = " + objDouble.Multiplication(c,
                                                        d));
System.out.println("Division = " + objDouble.Division(c, d));

// float type
Calculator<Float> objFloat = new Calculator<>();

Float e, f;

  System.out.println("Enter the value of e and f for Float type: ");
  e = in.nextFloat();
```

```
        f = in.nextFloat();
        System.out.println("For Float values: ");
        System.out.println("Addition = " + objFloat.Addition(e, f));
        System.out.println("Subtraction = " + objFloat.Subtraction(e, f));
        System.out.println("Multiplication = " + objFloat.Multiplication(e, f));
        System.out.println("Division = " + objFloat.Division(e, f));
        in.close();
        }
}
```

OUTPUT :

```
Enter the value of a and b for Integer type:
45 9
For integer values:
Addition = 54.0
Subtraction = 36.0
Multiplication = 405.0
Division = 5.0
Enter the value of c and d for Double type:
56.98
34.21
For Double values:
Addition = 91.19
Subtraction = 22.769999999999996
Multiplication = 1949.2857999999999
Division = 1.665594855305466
Enter the value of e and f for Float type:
4.67
2.11
For Float values:
Addition = 6.7799999713897705
Subtraction = 2.56000018119812
Multiplication = 9.85369967107772
```

Division = 2.2132702883768904

LAB 7

AIM: Collection , I/O

(1) Write a Java program that accepts two filenames. Based on the user's choice to copy or append, copy the first file into the second file or append the content of the first file to the second file.

CODE:

```
package lab7;
import java.io.*;
import java.util.Scanner;

public class IO_operations {
 public static void main(String[] args) throws IOException
 {
        FileInputStream first = null;
        FileOutputStream second = null;
        System.out.println("Enter your choice : ");
        System.out.println("Copy : 1 ");
        System.out.println("append : 2");
        int choice;
        Scanner in = new Scanner(System.in);
        choice = in.nextInt();
        try {

                if(choice == 1)
                {
                 first = new FileInputStream("C:\\Users\\Drashti
                        Bhalodiya\\eclipse-workspace\\LAB_07\\src\\f1.txt");
                 second = new FileOutputStream("C:\\Users\\Drashti
                              Bhalodiya\\eclipse
                               workspace\\LAB_07\\src\\f2.txt");
```

```
                }
            if(choice == 2)
            {
                first = new FileInputStream("C:\\Users\\Drashti
                        Bhalodiya\\eclipse-
                        workspace\\LAB_07\\src\\f1.txt");
                second = new FileOutputStream("C:\\Users\\Drashti
                        Bhalodiya\\eclipse-
                        workspace\\LAB_07\\src\\f2.txt", true);
            }
            int c;
            while((c = first.read()) != -1)
            {
             second.write(c);

            }
        }
        catch(Exception e)
        {
            System.out.println("Error occurs : " + e);
        }
        finally {
            if(first != null)
            {
                first.close();
            }
            if(second != null)
            {
                second.close();
            }
                in.close();
            }
        }
}
```

OUTPUT:

Context of file_1 This is the content of First file...
(1) Java
(2) Pyhton
(3) C
(4) C++
(5) php

Context of file_2 This is the content of second file -> hello world -> my name is drashti bhalodiya

Enter your choice :
Copy : 1
append : 2
2

This is the content of second file -> hello world -> my name is drashti bhalodiyaThis is the content of First file...
(1) Java
(2) Pyhton
(3) C
(4) C++
(5) php

(2) Write a Java program to generate a linked list of some five students (Student objects) and display the list of students in a sorted order as per name of students.

CODE:

```
package lab7;
import java.util.*;
```

```
class Student implements Comparable<Student> {
        private String name;
        private int age;

        Student(String name, int age) {
           this.name = name;
           this.age = age;
        }

        @Override
        public int compareTo(Student arg) {
           return this.name.compareTo(arg.name);
        }

        public String getName() {
           return name;
        }

        public void setName(String name) {
           this.name = name;
        }

        public int getAge() {
           return age;
        }

        public void setAge(int age) {
           this.age = age;
        }
}


public class LinkedList_student {
   public static void main(String[] args) {
```

```
        Scanner in = new Scanner(System.in);
        int n;
        System.out.println("Enter the number of students");
        n = in.nextInt();

        Student[] stu = new Student[n];
        for (int i = 0; i < n; i++) {
            System.out.println("for student " + (i+1));
            String name;
            int age;
            System.out.println("Enter the name : ");
            name = in.next();
            in.nextLine();
            System.out.println("Enter the age : ");
            age = in.nextInt();

            stu[i] = new Student(name, age);
        }

        List<Student> list = Arrays.asList(stu);

        LinkedList<Student> llist = new LinkedList<>(list);

        Collections.sort(llist);

        for (Student student : llist) {
            System.out.println(student.getName() + " " + student.getAge());
        }

        in.close();
    }
}
```

OUTPUT:

```
Enter the number of students
6
for student 1
Enter the name :
Smruti
Enter the age :
19
for student 2
Enter the name :
Drashti
Enter the age :
19
for student 3
Enter the name :
Aman
Enter the age :
19
for student 4
Enter the name :
Meet
Enter the age :
20
for student 5
Enter the name :
Chintan
Enter the age :
19
for student 6
Enter the name :
Jaimin
Enter the age :
19
Aman 19
```

Chintan 19
Drashti 19
Jaimin 19
Meet 20
Smruti 19

(3) Write a Java program to map Person objects to string hobby using TreeMap class. This mapping should store Person objects in ascending sorting by their name.

Persons hobby

Name: Bhairavi Age: 22 Singing
Name: Dhara Age:23 Sketching
Name: Anmol Age: 23 Reading
Name: Megh Age 21 Singing
Name: Raag Age:22 Sketching
Find the unique list/set of all the hobbies that are mapped in this collection and display it.

CODE:

```
package lab7;
import java.util.*;


public class TreeMap_example {
   public static void main(String[] args) {
      Scanner in = new Scanner(System.in);
      int n;
      System.out.println("Enter the number of students");
      n = in.nextInt();
```

```
Student[] stu = new Student[n];
TreeMap<Student, String> StuHob = new TreeMap<>();
Set<String> hobbies = new HashSet<>();

for (int i = 0; i < n; i++) {
        System.out.println("for student " + (i+1));
        String name;
        int age;
        String hobby;
        System.out.println("Enter the name : ");
        name = in.next();
        in.nextLine();
        System.out.println("Enter the age : ");
        age = in.nextInt();
        in.nextLine();
        System.out.println("Enter your hobby : ");
        hobby= in.next();


        stu[i] = new Student(name, age);
        StuHob.put(stu[i], hobby);
        hobbies.add(hobby);
}


for (Map.Entry<Student, String> entry : StuHob.entrySet())
{
        Student obj = entry.getKey();
        String hob = entry.getValue();
        System.out.println("---------------------------");
        System.out.println("Name : " + obj.getName() + '\n' + "Age. :  " +
                        obj.getAge() + '\n' + "Hobby : " + hob);
}

System.out.println("--------------------------");
```

```
    System.out.println("unique hobbies : ");
    for(String hob : hobbies)
    {
        System.out.println(hob);
      }

      in.close();
   }
}
```

OUTPUT:

```
Enter the number of students
6
for student 1
Enter the name :
Smruti
Enter the age :
19
Enter your hobby :
Singing
for student 2
Enter the name :
Drashti
Enter the age :
19
Enter your hobby :
Reading
for student 3
Enter the name :
Jaimin
Enter the age :
19
Enter your hobby :
```

```
Singing
for student 4
Enter the name :
Chintan
Enter the age :
19
Enter your hobby :
Sports
for student 5
Enter the name :
Meet
Enter the age :
20
Enter your hobby :
writing
for student 6
Enter the name :
Aman
Enter the age :
19
Enter your hobby :
Sports
 ---------------------------
Name : Aman
Age. :  19
Hobby : Sports
---------------------------
Name : Chintan
Age. :  19
Hobby : sports
---------------------------
Name : Drashti
Age. :  19
Hobby : Reading
---------------------------
```

```
Name : Jaimin
Age. :  19
Hobby : Singing
---------------------------
Name : Meet
Age. :  20
Hobby : writing
---------------------------
Name : Smruti
Age. :  19
Hobby : Singing
---------------------------
unique hobbies :
Reading
Singing
writing
Sports
```

(4) Write a generic interface MinMax having two methods findMin() and findMax(). Create a class named MyClass which implements the above interface.Write an appropriate demo class to test your classes and interfaces. Create an object of MyClass which stores an array of Book and find books having minimum and maximum price.

CODE:

```
package lab7;
interface MinMax <T extends Comparable<T>> {
        T findMin(T[] array);
        T findMax(T[] array);
}

class MyClass<T extends Comparable<T>> implements MinMax<T> {
  @Override
```

```
    public T findMin(T[] array)
    {
     T min = array[0];
        for (T element : array) {
            if (element.compareTo(min) < 0) {
                min = element;
            }
        }
        return min;
    }

    @Override
    public T findMax(T[] array)
    {
     T max = array[0];
        for (T element : array) {
            if (element.compareTo(max) > 0) {
                max = element;
            }
        }
        return max;
    }
}

class Book implements Comparable<Book> {
    private String title;
    private double price;

    public Book(String title, double price) {
        this.title = title;
        this.price = price;
    }

    public String getTitle() {
        return title;
```

```
    }

    public double getPrice() {
        return price;
    }

    @Override
    public int compareTo(Book other) {
        return Double.compare(this.price, other.price);
    }
}


public class demo {

 public static void main(String[] args) {

   Scanner in = new Scanner(System.in);
   int n;
      System.out.println("Enter the number of books");
      n = in.nextInt();

      Book[] books = new Book[n];

      for(int i = 0; i < n; i++)
      {
         System.out.println("for book " + (i+1));
         String title;
         double price;
         System.out.println("Enter the title : ");
         title = in.next();
         in.nextLine();
         System.out.println("Enter the price : ");
         price = in.nextDouble();
```

```
        books[i] = new Book(title, price);
    }

            MyClass<Book> myClass = new MyClass<>();

            Book minBook = myClass.findMin(books);
            System.out.println("Book with minimum price: " +
                                minBook.getTitle() + '\n' + minBook.getPrice());


            Book maxBook = myClass.findMax(books);
            System.out.println("Book with maximum price: " + maxBook.get
                                    Title() + '\n' + maxBook.getPrice());

           in.close();
     }
}
```

OUTPUT:

```
Enter the number of books
4
for book 1
Enter the title :
Smoke and Ashes
Enter the price :
594
for book 2
Enter the title :
Empire Building
Enter the price :
679
for book 3
Enter the title :
The Last Heroes
```

Enter the price :
389
for book 4
Enter the title :
1947-1957, india
Enter the price :
900
Book with minimum price: The
389.0
Book with maximum price: 1947-1957,
900.0

LAB 8

AIM: Servlet,JSP

(1) Create a Simple JAVA web application to display the welcome message using JSP or servlet.

CODE:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>First</title>
</head>
<body>
        <h1>Welcome to the First jsp file</h1>
        <h1>Hello World!!</h1>
</body>
</html>
```

OUTPUT :

(2) Write a Java web application for a login module which contains the following components:

index.html: for getting input from the user.
ValidateServlet.java: a servlet class for validating the user. If it is a valid User (validate from a database using PreparedStatement),it will forward the request to the WelcomeServlet. If the user is not validated then it displays an Error message along with the response from index.html.
Welcome.jsp: a JSP file for displaying the welcome message.

CODE:

- index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Log in page</title>
</head>
<body>
	<h1>log-in page</h1>
	<form action="validate" method="post">
		Enter User name : <input type="text" name="user">
		 <br><br><br>
		Enter password : <input type="password" name="pass">
		<br><br><br>
		<button type="submit">Login</button>
	</form>
</body>
</html>
```

- ValidateServlet.java

```
package servlets;

import java.io.IOException;
```

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

import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class ValidateServlet extends HttpServlet{

   @Override
   protected void doPost(HttpServletRequest req, HttpServletResponse
                        res) throws ServletException, IOException
   {
            res.setContentType("text/html");
            PrintWriter out = res.getWriter();

            String userName = req.getParameter("user");
            String password = req.getParameter("pass");

            try {
               Class.forName("com.mysql.cj.jdbc.Driver");
                     try(Connection conn =
                              DriverManager.getConnection
                              ("jdbc:mysql://localhost:3306/lab_db","root",""))
                        {

                              PreparedStatement prestmt =
                                       conn.prepareStatement("SELECT
                                       * FROM userlog WHERE userName = ? AND
                                        password = ?");

                 prestmt.setString(1, userName);
                 prestmt.setString(2, password);
                 ResultSet rs = prestmt.executeQuery();

                 if(rs.next()) {
```

```
                    RequestDispatcher rd =
                                req.getRequestDispatcher("Welcome.jsp");
                    rd.forward(req, res);
                }
                 else
                {
                    out.println("<h1>Sorry, username or password is
                                 incorrect!</h1>");
                   RequestDispatcher rd =
                                req.getRequestDispatcher("index.html");
                    rd.include(req, res);
                }


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

          } catch (ClassNotFoundException e) {
                 e.printStackTrace();
      }
  }
}
```

- welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
```

```
<body>
        <h1>Welcome again <%= request.getParameter("user") %></h1>
</body>
</html>
```

OUTPUT :

Enter user name : Narendra Modi
Enter password : namo@2024

Welcome again Narendra Modi

Enter user name : abcde
Enter password : 123abc@

Sorry, username or password is incorrect!

LAB - 9

AIM: Servlet,JSP

(1) Write a Java web application using HttpSession which allows only logged in users to access the other JSPs/Servlets of the application. Write the following components:

1. Login.html allows users to provide username and password and send them as request parameters toLoginVerifierServlet.
2. LoginVerifierServlet takes username and password from login.html and verifies it. If credentials are correct then it creates a session. It displays welcome message along with username and links to first.jsp and second.jsp.
3. first.jsp and second.jsp display some text with username and can be accessed if the user is logged in. (you should delegate to Login.html If the user is not logged in)

CODE:

Login.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Log in page</title>
</head>
<body>
	<h1>log-in page</h1>
	<form action="validate" method="post">
		Enter User name : <input type="text" name="user">
		<br><br><br>
		Enter password (within 15) : <input type="password"
				name="pass">
		<br><br><br>
		<button type="submit">Login</button>
	</form>
```

```
</body></html>
```

LoginVerifierServlet

```
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;

import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;


public class LoginVerifierServlet extends HttpServlet{

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse
                        res) throws ServletException, IOException
    {
                res.setContentType("text/html");
                PrintWriter out = res.getWriter();

                String userName = req.getParameter("user");
                String password = req.getParameter("pass");

                try {
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        try(Connection conn =
                                DriverManager.getConnection("jdbc:mysql://lo
                                calhost:3306/lab_db","root",""))
                        {

                                PreparedStatement prestmt =
                                        conn.prepareStatement("SELECT * FROM
```

```
                    userlog WHERE userName = ? AND password = ?");

            prestmt.setString(1, userName);
            prestmt.setString(2, password);
            ResultSet rs = prestmt.executeQuery();

            if(rs.next()) {

                HttpSession session = req.getSession();
                session.setAttribute("user", userName);
                RequestDispatcher rd =
                        req.getRequestDispatcher("Welcome.jsp");
                 rd.forward(req, res);
            }
            else
            {

              out.println("<h1>Sorry, username or password is
                        incorrect!</h1>");
              RequestDispatcher rd =
                      req.getRequestDispatcher("Login.html");
              rd.include(req, res);
           }


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

      } catch (ClassNotFoundException e) {
            e.printStackTrace();
      }
  }
  private static final long serialVersionUID = 1L;
}
```

- Welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1" session="true" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
 <%
        HttpSession s = request.getSession(false);
        if(s != null){
                String name = (String)s.getAttribute("user");
                if(name != null)
                {
 %>
                        <h1>Welcome again  ${user}</h1>
                        <a href="first.jsp">First</a>
                        |
                        <a href="second.jsp">Second</a>
                        <br><br>
                        <a href="logout.jsp">Logout</a>
 <%
        }
        else
        {
                response.sendRedirect("Login.html");
        }
 }
 else
 {
        response.sendRedirect("Login.html");
 }
 %>
```

```
</body> </html>
```

- First.jsp

```
 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1" session="true"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
        HttpSession s = request.getSession(false);
        if(s != null){

                String name = (String)s.getAttribute("user");
                if(name != null)
                {
 %>
                        <h1>Welcome again in the first jsp file ${user}</h1>
                        <hr>
                        <h2>Servlet Information</h2>
                        <h3 style="color:grey;">Servlet technology is used to
                                create a
                                web application (resides at server side and
                                generates a dynamic web page).
                                Servlet technology is robust and scalable because
                                of java language. Before Servlet, CGI (Common
                                Gateway Interface)  scripting language was
                                common as a server-side programming language.
                                However, there were many disadvantages to this
                                technology. There are many interfaces and classes
                                in the  Servlet API such as Servlet, GenericServlet,
                                HttpServlet,  ServletRequest, ServletResponse,
                                etc.
                        </h3>
```

```
                    <a href="Welcome.jsp">Welcome</a>
                |
                <a href="second.jsp">Second</a>
                <br><br>
                <a href="logout.jsp">Logout</a>
 <%
            }
            else
            {
                    response.sendRedirect("Login.html");
            }
        }
        else
        {
            response.sendRedirect("Login.html");
        }
%>

</body>
</html>
```

Second.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1" session="true"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
        HttpSession s = request.getSession(false);
        if(s != null){
```

```
            String name = (String)s.getAttribute("user");
            if(name != null)
            {
%>
            <h1>Welcome again in the second jsp file ${user}</h1>
            <hr>
            <h2>JSP Information</h2>
            <h3 style="color:grey;">JSP technology is used to create web
                application just like Servlet technology.  It can be
                thought of as an extension to Servlet because it provides
                more functionality than servlet such as expression
                language, JSTL, etc.  A JSP page consists of HTML tags
                and JSP tags. The JSP pages are easier to maintain than
                Servlet because we can separate designing and
                development. It provides some additional features such
                as Expression Language, Custom Tags, etc.
            </h3>

            <a href="Welcome.jsp">Welcome</a>
            |
            <a href="first.jsp">First</a>
            <br><br>
            <a href="logout.jsp">Logout</a>
<%
        }
        else
        {
            response.sendRedirect("Login.html");
        }
 }
 else
 {
        response.sendRedirect("Login.html");
 }
%>
</body>
</html>
```

- Logout.html

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1" session="true" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
        <% session.invalidate();
                response.sendRedirect("Login.html");
        %>

</body>
</html>
```

OUTPUT :

OUTPUT: Enter user name : Deep123
Enter password : 123Hello

Welcome again Deep123

First | Second

Logout

Welcome again in the first jsp file Deep123

Servlet Information

Servlet technology is used to create a web application (resides at server side and generates a dy namic web page). Servlet technology is robust and scalable

because of java lan guage. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

Welcome | Second

Logout

Welcome again in the second jsp file Deep123

JSP Information

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc. A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional fea tures such as Expression Language, Custom Tags, etc.

Welcome | First

Logout

log-in page

Enter User name :

Enter password (within 15) :

Login

(2) Write a web based java application containing a JSP which performs the simple arithmetic calculation. Take the necessary operands and operators in textboxes.
Write your JSP code using jsp:useBean action tag.

CODE:

- Calculator.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<form method="post" action="show.jsp">
		Enter num1 : <input type="text" name="num1"><br><br>
		Enter num2 : <input type="text" name="num2"><br><br>
		Enter operator : <input type="text" name="op"><br><br>
		<button type="submit">Calculate</button>
	</form>
</body>
</html>
```

- Show.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
```

```
<body>
        <jsp:useBean class="classes.calc" id="c1" scope="request" />
        <jsp:setProperty name="c1" property="num1" param="num1"/>
        <jsp:setProperty name="c1" property="num2" param="num2"/>
        <jsp:setProperty name="c1" property="operator" param="op"/>

        num1 : <jsp:getProperty property="num1" name="c1"/><br>
        num2 : <jsp:getProperty property="num2" name="c1"/><br>
        operator : <jsp:getProperty property="operator" name="c1"/><br>

        <%
         int num1 = c1.getNum1();
         int num2 = c1.getNum2();
         char op = c1.getOperator();
         double ans;


         if(op == '+')
         {
                ans = (double)num1 + (double)num2;
         }
         else if(op == '-')
         {
                ans = (double)num1 - (double)num2;
         }
         else if(op == '*')
         {
                ans = (double)num1 * (double)num2;
         }
         else if(op == '/')
         {
                ans = (double)num1 / (double)num2;
         }
         else
         {
                ans = (double)num1 % (double)num2;
         }
        %>
```

```
        Ans : <%= ans %>
</body>
</html>
```

OUTPUT :

Enter num1 : 24

Enter num2 : 12

Enter operator : +

Calculate

num1 : 24
num2 : 12
operator : +
Ans : 36.0

Project (Lab 10-11-12)

Project Definition : Organ Donation Management System (ODMS)

Description :

The Organ Donation Management System (ODMS) is a software platform that facilitates organ donation and transplantation processes. It includes users, authorities, doctors, donors, organs, patients, and transplant records. Users register and access the system based on their roles. Authorities regulate policies, while doctors evaluate donors and recipients. Donors consent to donation, and organs are matched with patients based on compatibility. The system schedules and tracks transplant surgeries, ensuring efficient coordination and improved patient outcomes.

Entities :

- Users

  Contains people who use the Organ Donation Management System, including doctors, coordinators, administrators, and other staff members involved in organ donation and transplantation processes.
- Authorities

  Contains user roles which user has which authorities
- Doctor

  Contains details of doctors which are the ones who check if people can donate organs or if they need them. They do surgeries to transplant organs and take care of patients before and after the surgery.
- Donor

  Contains details of donors who want to donate organs

- Organ

  Contains details of organ which are donated by donors
- Patient

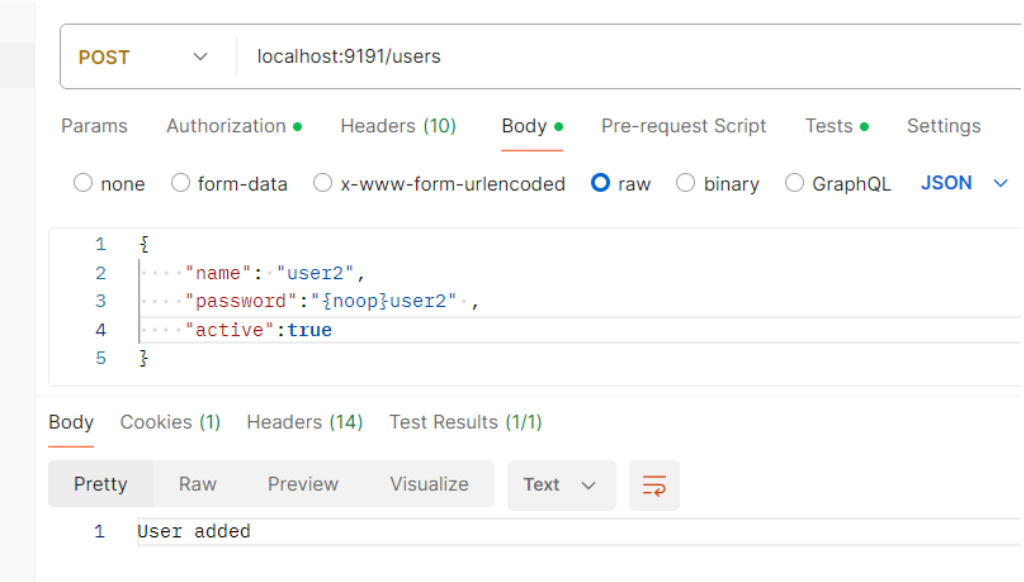
  Contains details of patients who need organs
- Transplant

  Contains details of surgery which donor donate organ and which doctor done surgery and surgery is successful or not etc.

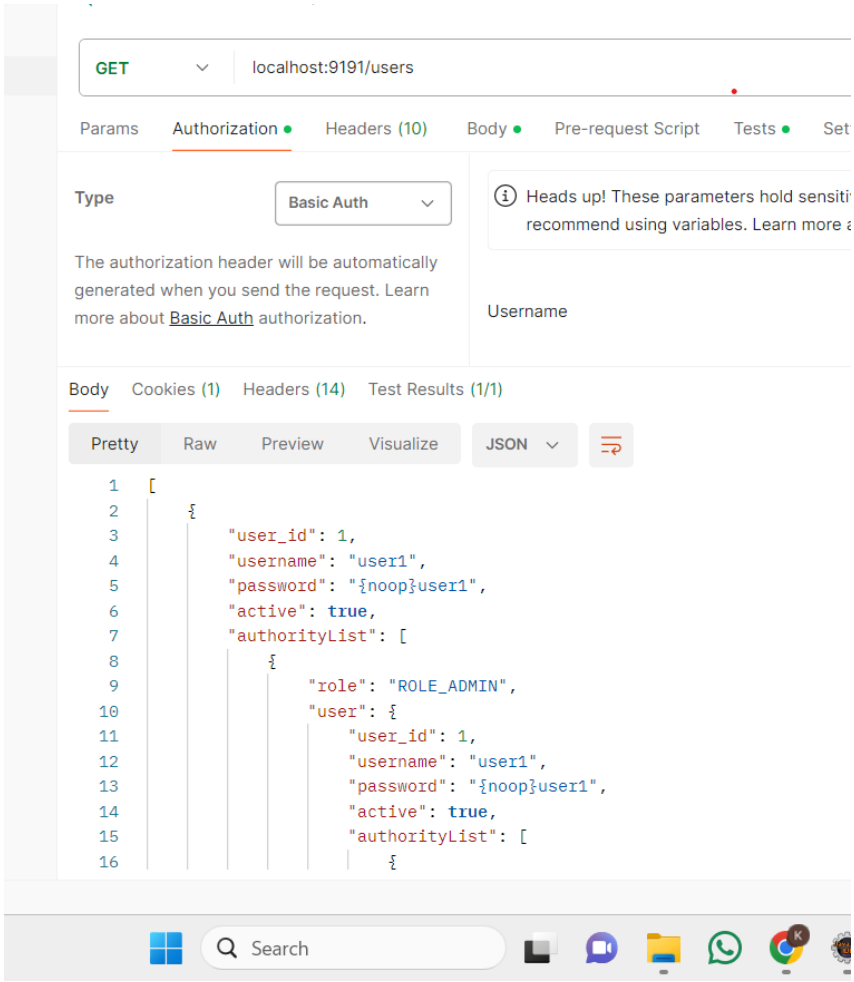
Crud Operations :

Crud on User

- add user



- Get user


GET
localhost:9191/users
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Set
Type
Basic Auth
Heads up! These parameters hold sensiti
recommend using variables. Learn more a
The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.
Username
Body
Cookies (1)
Headers (14)
Test Results (1/1)
Pretty
Raw
Preview
Visualize
JSON
[
{
"user_id": 1,
"username": "user1",
"password": "{noop}user1",
"active": true,
"authorityList": [
{
"role": "ROLE_ADMIN",
"user": {
"user_id": 1,
"username": "user1",
"password": "{noop}user1",
"active": true,
"authorityList": [
{
Search

-update user

users / **Update data**

PUT http://localhost:9191/users/3

Params Authorization Headers (10) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary Grap

```
1  {
2      "user_id":3,
3      "username":"yash",
4      "active":0,
5      "password":"{noop}yash123"
6  }
7
8
```

Body Cookies (1) Headers (14) Test Results (1/1)

Pretty Raw Preview Visualize Text

```
1  User details of 3 is updated.
```

-delete user

users / **Delete data**

DELETE http://localhost:9191/users/3

Params Authorization Headers (8) Body Pre-request S

none form-data x-www-form-urlencoded raw

```
1
```

Body Cookies (1) Headers (14) Test Results (1/1)

Pretty Raw Preview Visualize Text

```
1  User data of id 3 is deleted successfully.
```

Crud on Roles

- add role

POST localhost:8080/roles

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

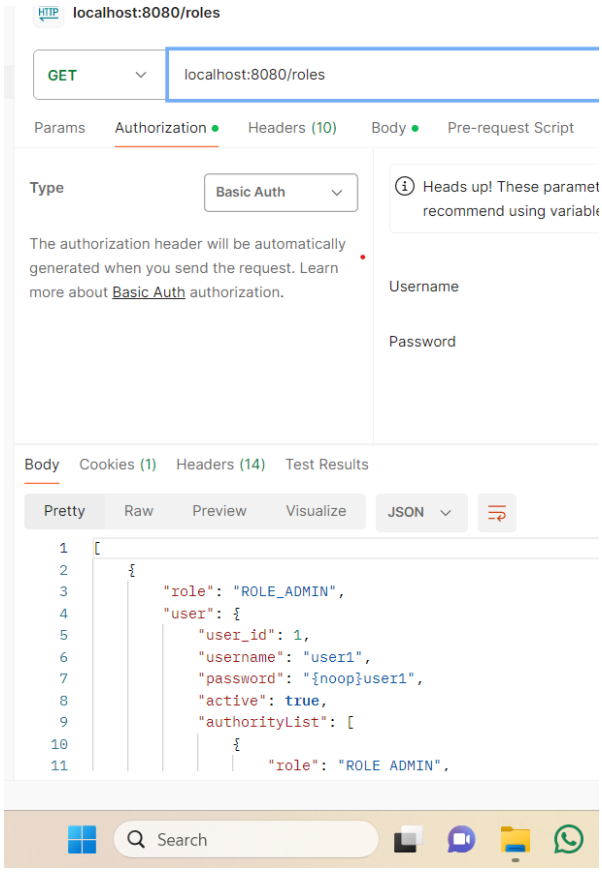
```
1  {
2      "role":"ROLE_ADMIN",
3      "user":
4      {
5          "user_id":2
6      }
7  }
```

Body Cookies (1) Headers (14) Test Results Status: 200 OK Time:

Pretty Raw Preview Visualize Text

```
1  role added
```

- get roles


localhost:8080/roles
GET
localhost:8080/roles
Params
Authorization
Headers (10)
Body
Pre-request Script
Type
Basic Auth
Heads up! These paramet
recommend using variabl
The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.
Username
Password
Body
Cookies (1)
Headers (14)
Test Results
Pretty
Raw
Preview
Visualize
JSON
[
{
"role": "ROLE_ADMIN",
"user": {
"user_id": 1,
"username": "user1",
"password": "{noop}user1",
"active": true,
"authorityList": [
{
"role": "ROLE_ADMIN",
Search

-update role

roles / **Update data**

PUT http://localhost:9191/roles/2

Params Authorization Headers (10) Body Pre-request Script

none form-data x-www-form-urlencoded raw binary

```
1 {
2     "user_id":2,
3     "role":"ROLE_DOCTOR"
4 }
```

Body Cookies (1) Headers (14) Test Results (1/1)

Pretty Raw Preview Visualize Text

```
1 Role details of 2 is updated.
```

-delete role

roles / **Delete data**

DELETE http://localhost:9191/roles/5

Params Authorization Headers (8) Body Pre-request Script Test

none form-data x-www-form-urlencoded raw binary

```
1
```

Body Cookies (1) Headers (14) Test Results (1/1)

Pretty Raw Preview Visualize Text

```
1  User data of id 5 is deleted successfully.
```

Crud on Doctor

- add doctor

doctor / **post**

POST http://localhost:9191/doctors?password=deep123

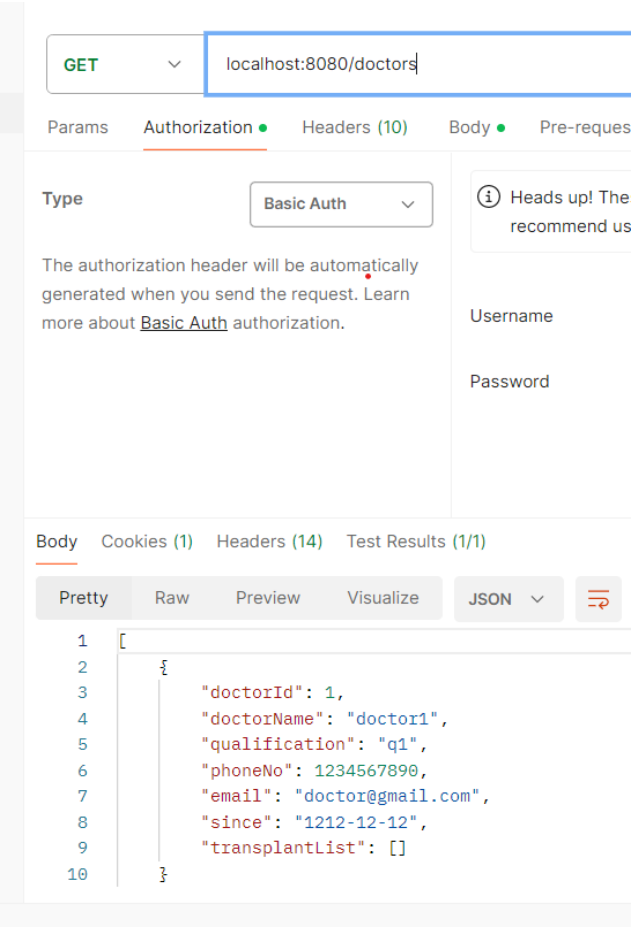
Params Authorization Headers (10) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary GraphQL

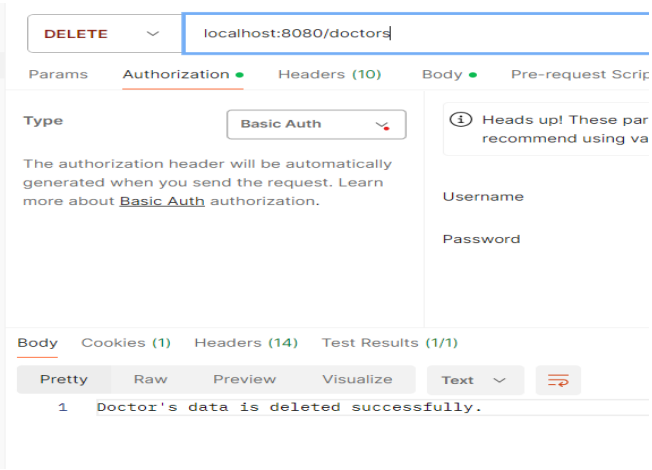
```
16 {
17     "doctorName":"deep",
18     "email" : "deep123@gmail.com",
19     "phoneNo" : 9876543211,
20     "since": "2013-09-09",
21     "qualification":"mbbs"
22 
23 }
```

Body Cookies (1) Headers (14) Test Results

- get doctor



-delete doctor



-delete by id

doctor / **delete**

DELETE http://localhost:9191/doctors/2

Params Authorization Headers (8) Body Pre-request Scrip

none form-data x-www-form-urlencoded raw bin

```
1
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
1 Doctor data of ID 2 has been deleted successfully.
2 Affected patients:
3 jkl
```

-update doctor

doctor / **put**

PUT http://localhost:9191/doctors/2

Params Authorization Headers (10) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary GraphQ

```
1 {
2     "doctorId": 2,
3     "doctorName": "yash1",
4     "qualification": "mbbs",
5     "phoneNo": 9876543278,
6     "email": "yash123@gmail.com",
7     "since": "2013-09-09"
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
1 Doctor details of 2 is updated.
```

Crud on Donor

- add donor

donor / **New Request**

POST http://localhost:9191/donors?password=JD123

Params • Authorization • Headers (10) Body • Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

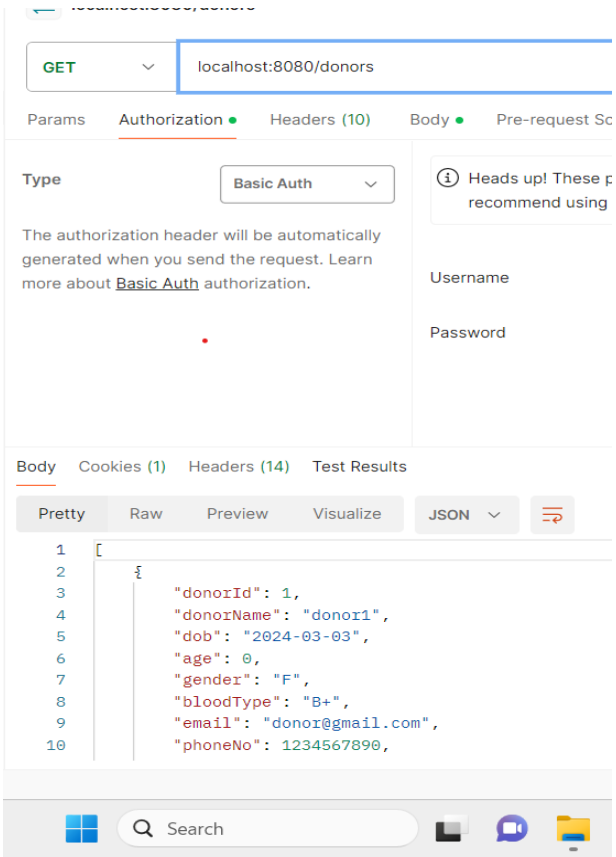
```
{
  "donorName": "John Doe",
  "dob": "1990-05-15",
  "gender": "M",
  "bloodType": "O+",
  "email": "john.doe@example.com",
  "phoneNo": 1234567890,
  "street": "123 Main Street",
  "city": "Anytown",
  "state": "State",
  "organsList": [
    {
      "organName": "heart",
      "available": true,
      "reasonOfDonation": "Donating because of personal reasons",
      "dateOfDonation": "2023-01-10"
```

Body Cookies (1) Headers (14) Test Results

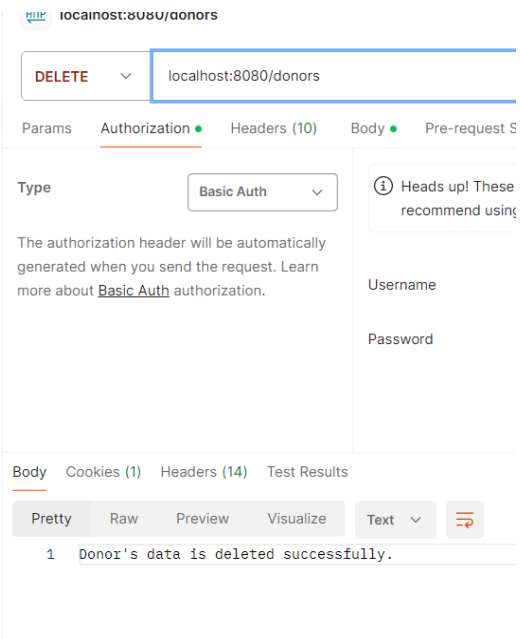
Pretty Raw Preview Visualize Text

```
Donor added
```

- get donor



- delete donor



-delete by id

donor / **delete**

DELETE http://localhost:9191/donors/2

Params | Authorization | Headers (8) | Body | Pre-request S

Type: Basic A...

The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.

Username

Password

Body | Cookies (1) | Headers (14) | Test Results

Pretty | Raw | Preview | Visualize | Text

```
Donor data of id 2 is deleted successfully.
```

-update donor

donor / **put**

PUT http://localhost:9191/donors/1

Params | Authorization | Headers (10) | Body | Pre-request Script | Tests

none | form-data | x-www-form-urlencoded | raw | binary | GraphQl

```
{
    "donorId" : 1,
    "donorName": "John Doe",
    "email": "john.doe@example.com",
    "bloodType": "O+",
    "dob": "2012-03-01",
    "phoneNo": 9807657854,
    "city" : "baroda",
    "street" : "jakanaka",
    "state" : "gujarat",
    "gender" : "M"

}
```

Body | Cookies (1) | Headers (14) | Test Results

Pretty | Raw | Preview | Visualize | Text

```
Donor details of 1 is updated.
```

Crud on Organ

- add organ

donor / **New Request**

POST http://localhost:9191/donors?password=JD123

Params • Authorization • Headers (10) Body • Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```
{
  "donorName": "John Doe",
  "dob": "1990-05-15",
  "gender": "M",
  "bloodType": "O+",
  "email": "john.doe@example.com",
  "phoneNo": 1234567890,
  "street": "123 Main Street",
  "city": "Anytown",
  "state": "State",
  "organsList": [
    {
      "organName": "heart",
      "available": true,
      "reasonOfDonation": "Donating because of personal reasons",
      "dateOfDonation": "2023-01-10"
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
Donor added
```

- get organ

GET localhost:8080/organs

Params Authorization Headers (10) Body Pre-request Script

Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.

Heads up! These paramet recommend using variable

Username

Password

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON


```
[
    {
        "organId": 1,
        "organName": "organ1",
        "available": true,
        "reasonOfDonation": "nothimg",
        "dateOfDonation": "2022-01-12",
        "donor": {
            "donorId": 2,
            "donorName": "donor1",
```


-delete organ

DELETE localhost:8080/organs

Params Authorization Headers (10) Body Pre-request Script

Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.

Heads up! These param recommend using variab

Username

Password

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
organ's data is deleted successfully.
```

-delete by id

organs / **delete**

DELETE http://localhost:9191/organs/1

Params Authorization Headers (8) Body Pre-request Scrip

Type Basic A...

The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.

Username

Password

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
1   organ details of 1 is deleted successfully.
```

-update organ

organs / put

PUT http://localhost:9191/organs/2

Params Authorization Headers (10) Body Pre-request Script Tests Setti

none form-data x-www-form-urlencoded raw binary GraphQL J


```
6    "dateOfDonation": "2013-02-10",
7    "donor" :
8    {
9        "donorId": 2,
10           "donorName": "devi",
11           "dob": "2012-09-01",
12           "age": 0,
13           "gender": "F",
14           "bloodType": "AB-",
15           "email": "x23@gmail.com",
16           "phoneNo": 1245778155,
17           "street": "jakanaka",
18           "city": "baroda",
19           "state": "gujarat"
```


Body Cookies (1) Headers (14) Test Results 20

Pretty Raw Preview Visualize Text

```
1  organ details of 2 is updated.
```

Crud on Patient

- add patient

patient / post

POST http://localhost:9191/patients?password=123jkl

Params • Authorization • Headers (10) Body • Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary GraphQL

```
{
    "patientName" : "jkl",
    "dob": "2019-07-09",
    "gender" : "M",
    "bloodType" : "O-",
    "email" : "123qwe@gmail.com",
    "phoneNo" : 1230965345,
    "street" : "kishan",
    "city" : "nadiad",
    "state" : "gujarat",
    "organRequired" : "heart",
    "reasonOfProcurement" : "heart failed",
    "requestDate" : "2022-03-24"
}
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
patient added
```

- get patient

GET localhost:8080/patients

Params Authorization • Headers (10) Body • Pre-request Scr

Type Basic Auth

Heads up! These pa
recommend using

The authorization header will be automatically

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON


```
[
    {
        "patientId": 1,
        "patientName": "patient1",
        "dob": "1999-12-02",
        "age": 0,
        "gender": "F",
        "bloodType": "B+",
        "email": "patient@gmail,com",
        "phoneNo": 1234567890,
        "street": "street1",
        "city": "city1",
        "state": "state1",
        "organRequired": "yes",
        "reasonOfProcurement": "reson",
        "requestDate": "2020-03-18",
        "transplant": null
    }
]
```

- update patient

patient / put

PUT http://localhost:9191/patients/2

Params Authorization Headers (10) Body Pre-request Script

none form-data x-www-form-urlencoded raw binary

```
{
    "patientId" : 2,
    "patientName" : "rajiv",
    "dob": "2019-07-09",
    "gender" : "M",
    "bloodType" : "O-",
    "email" : "123qwe@gmail.com",
    "phoneNo" : 1230965345,
    "street" : "kishan samosa",
    "city" : "nadiad",
    "state" : "gujarat",
    "organRequired" : "heart",
    "reasonOfProcurement" : "",
    "requestDate" : "2022-03-24"
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
patient details of 2 is updated.
```

- delete patient

DELETE localhost:8080/patients

Params Authorization Headers (10) Body Pre-request Scr

Type Basic Auth

Heads up! These pa
recommend using

The authorization header will be automatically

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
patient's data is deleted successfully.
```

Crud on Transplant

- add transplant

transplants / post

POST http://localhost:9191/transplants

Params Authorization Headers (10) Body Pre-request Script Test

none form-data x-www-form-urlencoded raw binary Gra

```
{
    "dateOfTransplantation" : "2022-04-01",
    "success":false,
    "organ" : {
        "organId" : 1
    },
    "patient" : {
        "patientId" : 1
    },
    "doctor" :{
        "doctorId" : 2
    }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
transplant added
```

- get transplant

localhost:8080/transplants

GET localhost:8080/transplants

Params Authorization Headers (10) Body Pre-request Script Tes

Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.

Heads up! These parameters l
recommend using variables. L

Username

Password

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
[]
```

- delete transplant

DELETE localhost:8080/transplants

Params Authorization Headers (10) Body

Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.

Username

Password

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
1  transplant's data is not found.
```

-delete by id

transplants / delete

DELETE http://localhost:9191/transplants/2

Params Authorization Headers (8) Body Pre-request Script

none form-data x-www-form-urlencoded raw binary

```
1
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize Text

```
1  transplant data of id 2 is deleted successfully.
```

Extra Query

-Available organs

organs / **delete**

GET http://localhost:9191/organs/availableOrgans

Params Authorization Headers (8) Body Pre-request Script

Type Basic A... Username

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON


```
[
    {
        "organName": "heart",
        "quantity": 1
    },
    {
        "organName": "kidney",
        "quantity": 2
    }
]
```


-Donor names by organ names

organs / **delete**

GET http://localhost:9191/organs/donorNames/heart

Params Authorization Headers (8) Body Pre-request Script

Type Basic A... Username

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON


```
[
    "John Doe",
    "devi"
]
```

-patient name by organ name

organs / **delete**

GET http://localhost:9191/organs/PatientName/heart

Params Authorization Headers (8) Body Pre-request Script Te

Type Basic A... Username

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
[
    "jkl",
    "rajiv"
]
```

-patient under doctor

transplants / **New Request**

GET http://localhost:9191/transplants/patientsUnderDoc/yash1

Params Authorization Headers (8) Body Pre-request Script Tests S

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
[
    {
        "patientName": "jkl",
        "transplantSuccess": true
    }
]
```

-success transplants by doctor name