

# Assignments- Java Programming and DSA

Name – Kiran Ramkisha Padole

Roll No. – 22

Class - FYMAC

## Assignment no. 1

1. Write a program to accept 10 inputs from user through command line and check whether the entered number is even or odd and save even number in even array and odd number in odd array.

```
import java.util.Scanner;

class EvenOddArr

{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        int[] a= new int[10];
        System.out.println("Enter the value of array");
    }
}
```

```
for( int i=0; i<10; i++)
{
    a[i]=s.nextInt();
}

System.out.println("Even array");

for(int i=0; i<10; i++)
{

    if(a[i] % 2 == 0)
    {
        System.out.println("The number is even " +a[i]);
    }
}

System.out.println("Odd array");

for(int i=0; i<10; i++)
{
    if(a[i] % 2 != 0)
    {
        System.out.println("The number is odd " +a[i]);
    }
}
```

```
}
```

```
}
```

```
}
```

```
Command Prompt
```

```
F:\JavaPrograms>javac EvenOddArr.java
```

```
F:\JavaPrograms>java EvenOddArr
Enter the value of array
```

```
5
4
6
8
9
2
3
4
7
1
```

```
Even array
```

```
The number is even 4
The number is even 6
The number is even 8
The number is even 2
The number is even 4
```

```
Odd array
```

```
The number is odd 5
The number is odd 9
The number is odd 3
The number is odd 7
The number is odd 1
```

```
F:\JavaPrograms>
```



```
Type here to search
```



2. Write a program to find all prime numbers between entered range.

```
import java.util.Scanner;
public class primeNo1
```

```
{  
    public static void main(String[] args)  
    {  
  
        Scanner sc=new Scanner(System.in);  
  
        System.out.print("Enter Starting Number : ");  
        int start = sc.nextInt();  
  
        System.out.print("Enter Ending Number : ");  
        int end = sc.nextInt();  
  
        System.out.println("Prime numbers between "+start+" and "+end+"  
are : ");  
  
        int count;  
  
  
        for(int i = start ; i <= end ; i++)  
        {  
  
            count = 0;  
  
            for(int j = 1 ; j <= i ; j++)  
            {  
  
                if(i % j == 0)  
                    count = count+1;  
            }  
        }  
    }  
}
```

```

        }

        if(count == 2)

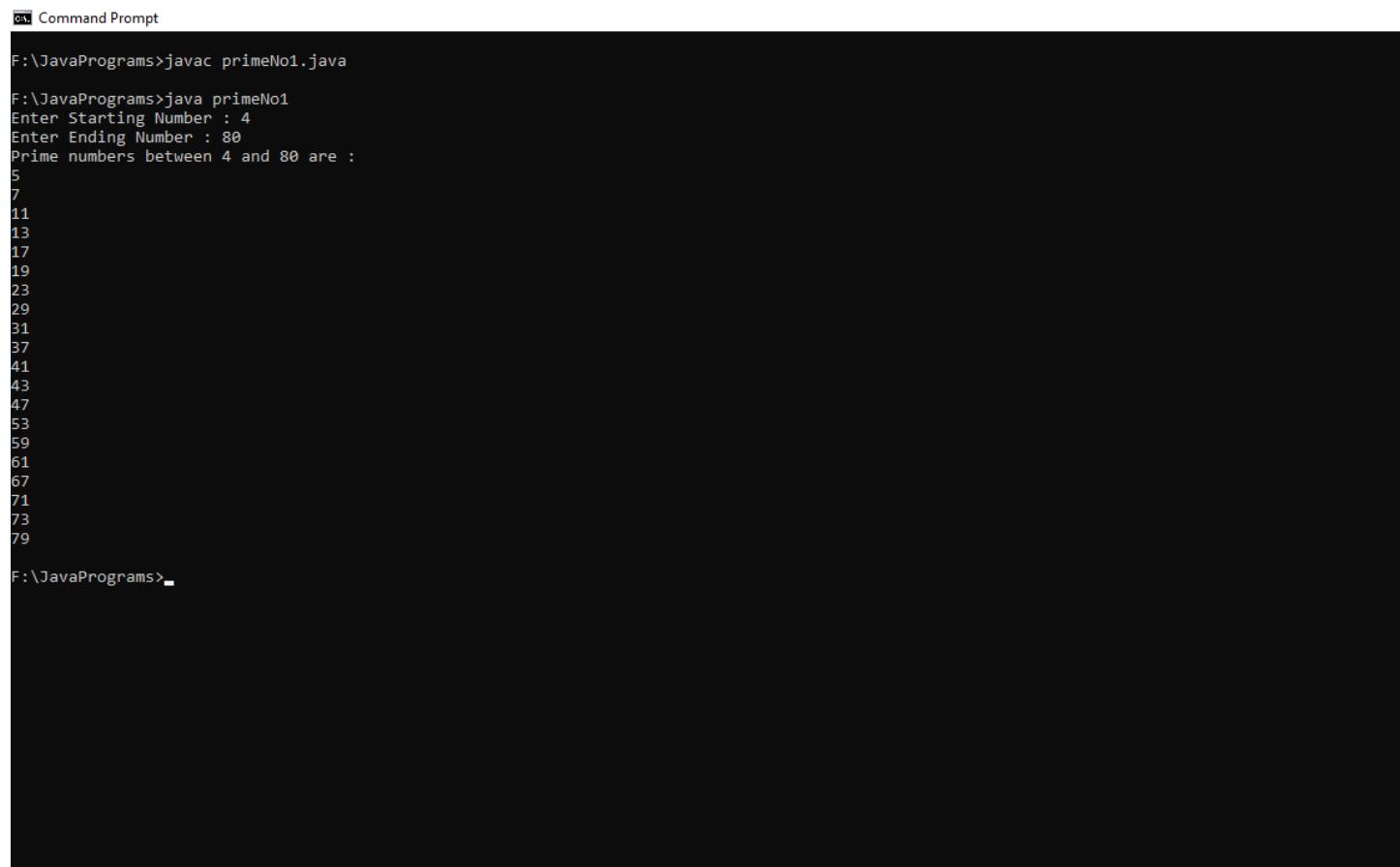
            System.out.println(i);

    }

}

}

```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command `javac primeNo1.java` is run, followed by `java primeNo1`. The user inputs "Enter Starting Number : 4" and "Enter Ending Number : 80". The program outputs a list of prime numbers between 4 and 80, including 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, and 79.

```

F:\JavaPrograms>javac primeNo1.java
F:\JavaPrograms>java primeNo1
Enter Starting Number : 4
Enter Ending Number : 80
Prime numbers between 4 and 80 are :
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
F:\JavaPrograms>_

```

3. Write a program to print factorial of an entered number using recursion function.

```
import java.util.Scanner;

public class FactorialRec

{
    public static void main(String[] args)

    {
        int n, mul;

        Scanner s = new Scanner(System.in);

        System.out.print("Enter any integer:");

        n = s.nextInt();

        FactorialRec obj = new FactorialRec();

        mul = obj.fact(n);

        System.out.println("Factorial of "+n+" :" +mul);

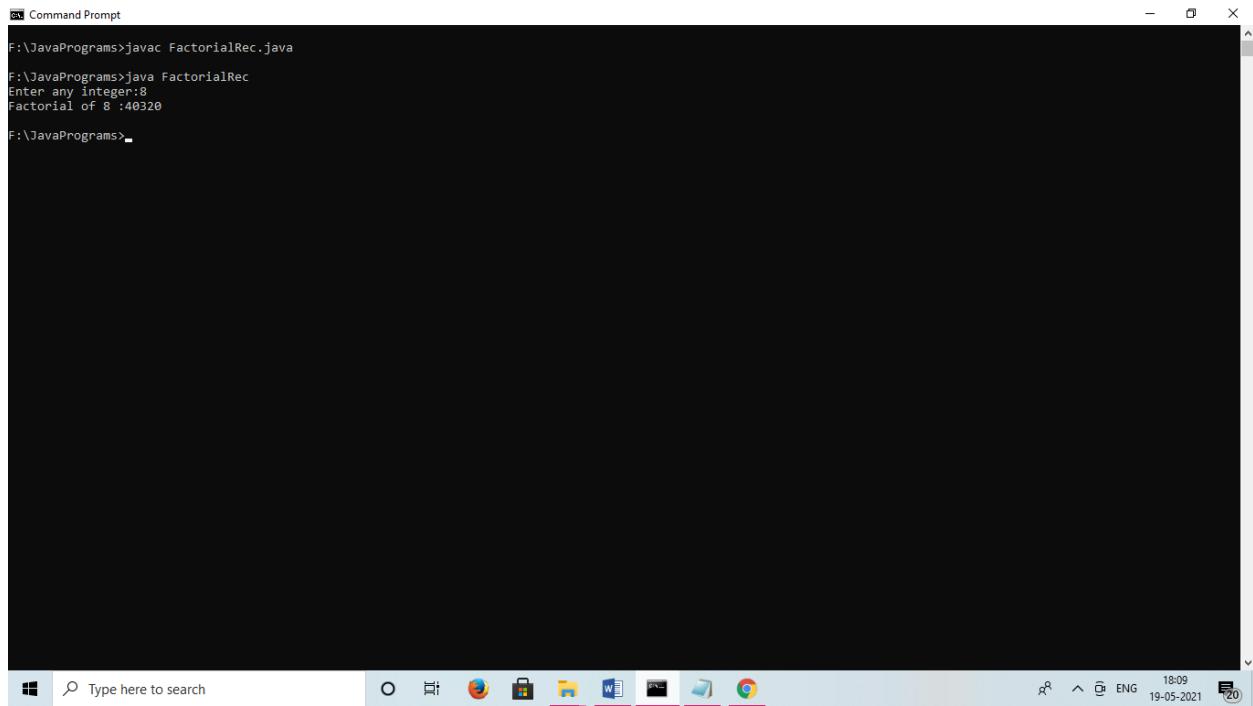
    }

    int fact(int x)

    {
        if(x > 1)

        {
            return(x * fact(x - 1));
        }
    }
}
```

```
    return 1;  
  
}  
  
}
```



The screenshot shows a Windows Command Prompt window titled 'Command Prompt'. The command 'javac FactorialRec.java' is run, followed by 'java FactorialRec'. The user is prompted to 'Enter any integer:8', and the output shows 'Factorial of 8 :40320'. The window has standard window controls (minimize, maximize, close) and is positioned over a desktop background with icons for File Explorer, Task View, Edge, and others.

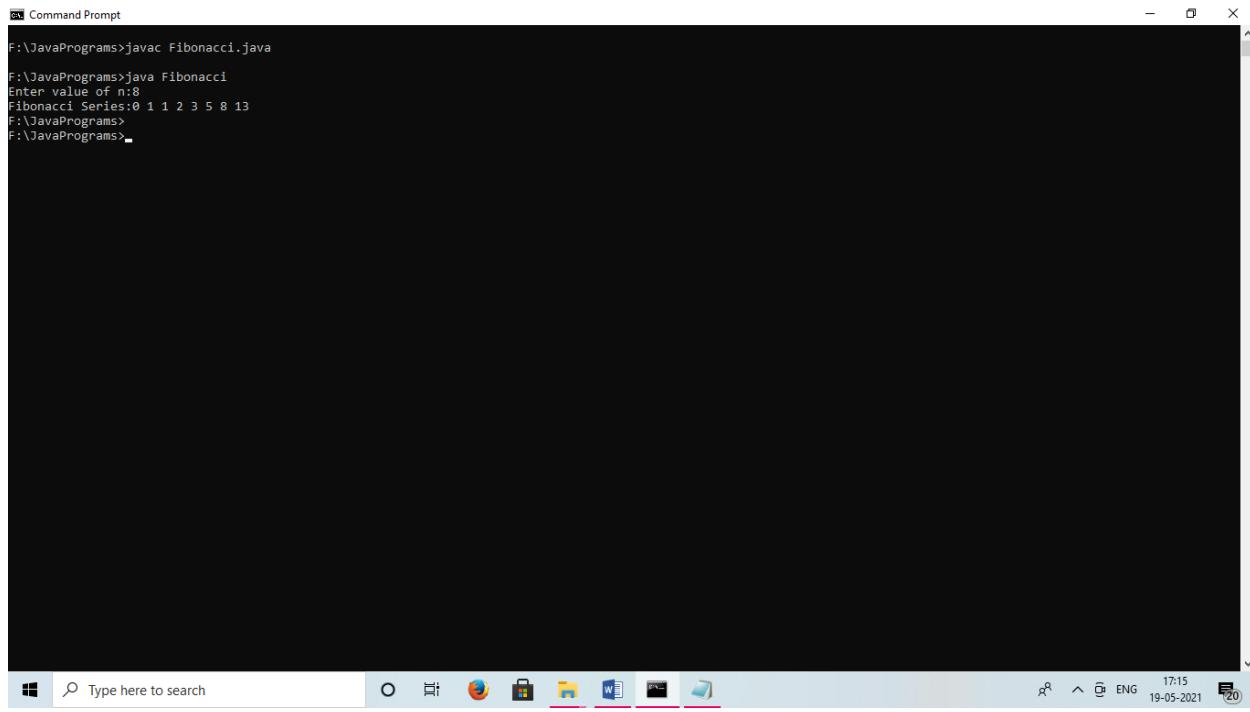
```
F:\JavaPrograms>javac FactorialRec.java  
F:\JavaPrograms>java FactorialRec  
Enter any integer:8  
Factorial of 8 :40320  
F:\JavaPrograms>
```

4. Write a program to generate Fibonacci series upto 'n' terms. Accept n from user.

Code:

```
import java.util.Scanner;  
  
public class Fibonacci  
{  
    public static void main(String[] args)
```

```
{  
    int n, a = 0, b = 0, c = 1;  
  
    Scanner s = new Scanner(System.in);  
  
    System.out.print("Enter value of n:");  
  
    n = s.nextInt();  
  
    System.out.print("Fibonacci Series:");  
  
    for(int i = 1; i <= n; i++)  
    {  
        a = b;  
        b = c;  
        c = a + b;  
        System.out.print(a+" ");  
    }  
}  
}
```



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text:

```
P:\JavaPrograms>javac Fibonacci.java
P:\JavaPrograms>java Fibonacci
Enter value of n:8
Fibonacci Series:0 1 1 2 3 5 8 13
P:\JavaPrograms>
P:\JavaPrograms>
```

The window has a dark background and a light gray border. At the bottom, there is a taskbar with the Windows logo, a search bar containing "Type here to search", and several pinned icons. On the right side of the taskbar, there are system status icons and the date/time "17:15 19-05-2021".

5. Write a program to calculate volume of cube, cylinder and cuboid using method overloading(use final variable for pi's value).

```
class Overload {

    double area(int l, float w, int h) {

        return l * w * h;

    }
```

```
    double area(float l) {

        return l * l * l;
```

```
}
```

```
double area(float r, float h) {
```

```
    final double n=3.14;
```

```
    return n * r * r * h;
```

```
}
```

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

```
{
```

```
    Overload ov = new Overload();
```

```
    double rectangleBox = ov.area(7, 8, 9);
```

```
    System.out.println("Area of cuboid is " + rectangleBox);
```

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

```
    double cube = ov.area(6);
```

```
    System.out.println("Area of cube is " + cube);
```

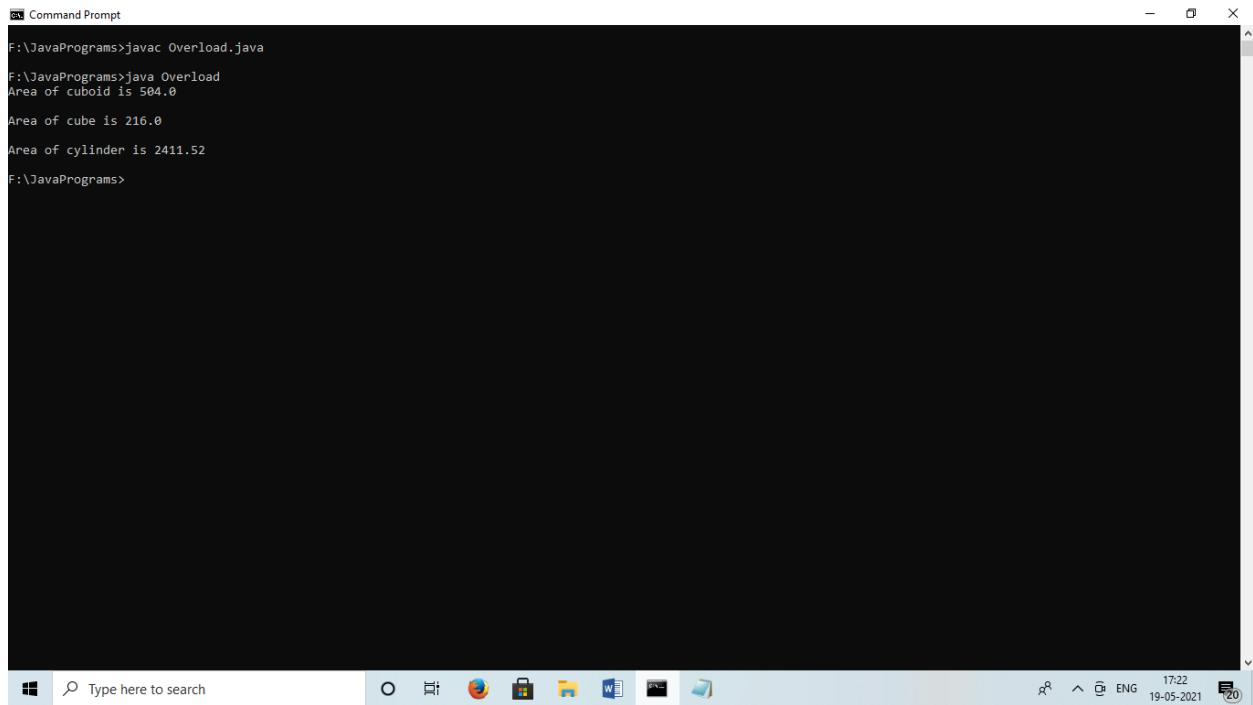
```
    System.out.println("");
```

```
    double cylinder = ov.area(8, 12);
```

```
    System.out.println("Area of cylinder is " + cylinder);
```

```
}
```

```
}
```



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the output of a Java program named "Overload.java". The program calculates the area of a cuboid, a cube, and a cylinder. The command "javac Overload.java" is run first, followed by "java Overload". The output displays the calculated areas: "Area of cuboid is 504.0", "Area of cube is 216.0", and "Area of cylinder is 2411.52". The prompt "F:\JavaPrograms>" is visible at the bottom.

```
F:\JavaPrograms>javac Overload.java
F:\JavaPrograms>java Overload
Area of cuboid is 504.0
Area of cube is 216.0
Area of cylinder is 2411.52
F:\JavaPrograms>
```

6. Write a program to count number of object created for a class using static variable and static member function.

```
class TestObj {
```

```
    static int noOfObjects = 0;
```

```
{
```

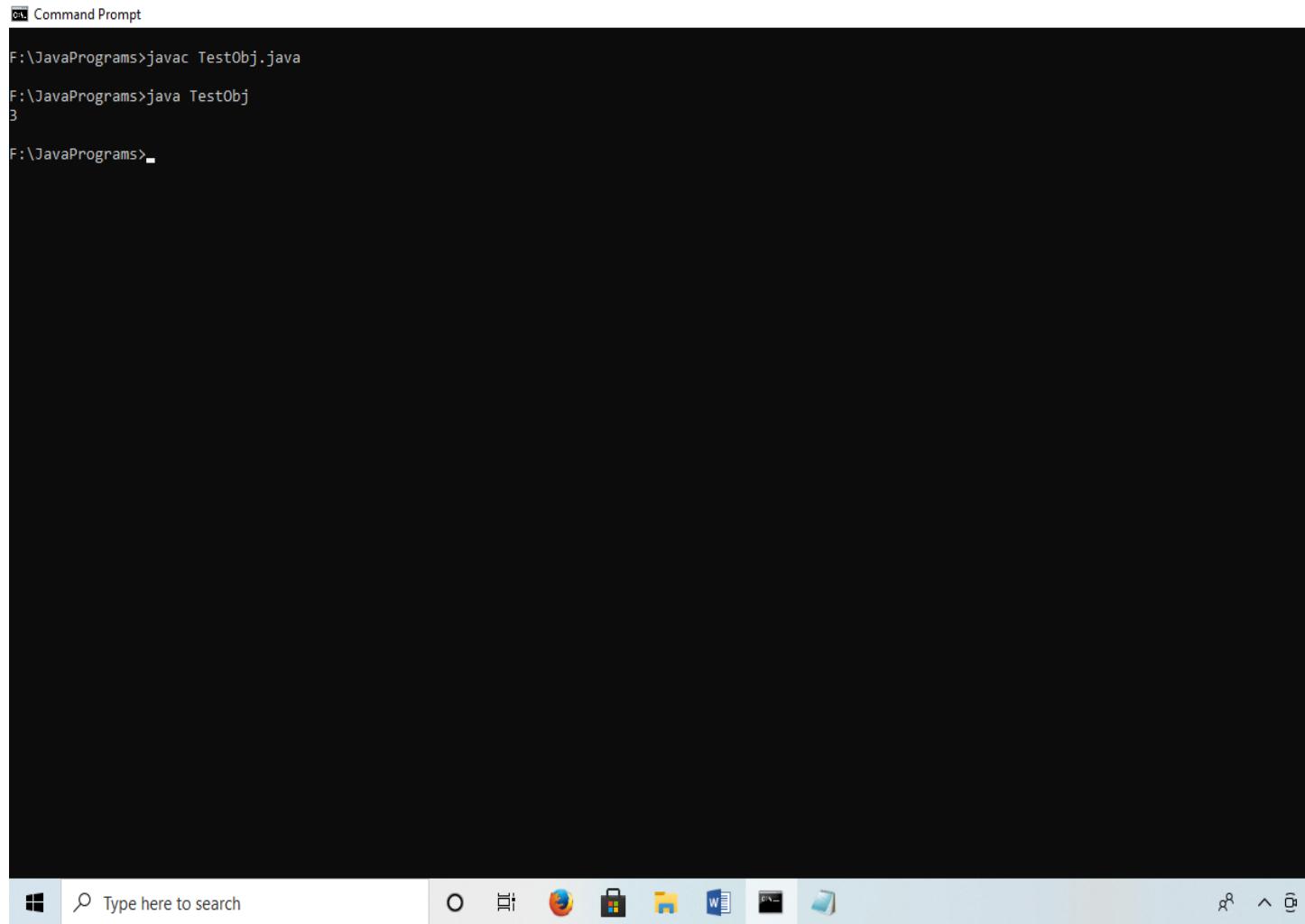
```
    noOfObjects += 1;
```

```
}
```

```
public TestObj()  
{  
}  
  
public TestObj(int n)  
{  
}  
  
public TestObj(String s)  
{  
}  
  
public static void main(String args[])  
{  
    TestObj t1 = new TestObj();  
    TestObj t2 = new TestObj(5);  
    TestObj t3 = new TestObj("GFG");  
  
    System.out.println(TestObj.noOfObjects);
```

```
}
```

```
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command line shows the following sequence of operations:

```
F:\JavaPrograms>javac TestObj.java
F:\JavaPrograms>java TestObj
3
F:\JavaPrograms>
```

The window has a standard Windows title bar and taskbar at the bottom. The taskbar includes icons for File Explorer, Task View, Mozilla Firefox, and other Microsoft applications like OneDrive, Mail, and Photos.

7. Write a program to sort one-dimensional array. Accept size and element of array from user.

```
public class SortAsc {
```

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

```
//Initialize array

int [] arr = new int [] {5, 2, 8, 7, 1};

int temp = 0;

//Displaying elements of original array

System.out.println("Elements of original array: ");

for (int i = 0; i < arr.length; i++) {

    System.out.print(arr[i] + " ");

}

//Sort the array in ascending order

for (int i = 0; i < arr.length; i++) {

    for (int j = i+1; j < arr.length; j++) {

        if(arr[i] > arr[j]) {

            temp = arr[i];

            arr[i] = arr[j];

            arr[j] = temp;

        }

    }

}
```

```

}

System.out.println();

//Displaying elements of array after sorting

System.out.println("Elements of array sorted in ascending order:

");

for (int i = 0; i < arr.length; i++) {

    System.out.print(arr[i] + " ");

}

}

```

```

Command Prompt
F:\JavaPrograms>javac SortAsc.java
F:\JavaPrograms>java SortAsc
Elements of original array:
5 2 8 7 1
Elements of array sorted in ascending order:
1 2 5 7 8
F:\JavaPrograms>

```



8. Write a program for addition, subtraction and multiplication of two 3\*3 matrices using two dimensional array.

```
class Matrix
{
    public static void main(String args[])
    {
        int a[][]={{1,2,3},{2,3,4},{3,4,5}};
        int b[][]={{1,2,3},{2,3,4},{4,5,6}};

        int c[][]=new int[3][3];
        for(int i=0; i<3; i++)
        {
            for(int j=0; j<3;j++)
            {
                c[i][j]=a[i][j]+b[i][j];
                System.out.print(c[i][j]+"\n");
            }
            System.out.println();
        }

        for(int i=0; i<3; i++)
        {
            for(int j=0; j<3;j++)
            {
                c[i][j]=a[i][j]-b[i][j];
                System.out.print(c[i][j]+"\n");
            }
            System.out.println();
        }
    }
}
```

```
}

for(int i=0; i<3; i++)
{
    for(int j=0; j<3;j++)
    {
        c[i][j]=a[i][j]*b[i][j];
        System.out.print(c[i][j]);
    }
    System.out.println();
}
}
```

Command Prompt

```
F:\JavaPrograms>javac Matrix.java
```

```
F:\JavaPrograms>java Matrix
```

```
246
```

```
468
```

```
7911
```

```
000
```

```
000
```

```
-1-1-1
```

```
149
```

```
4916
```

```
122030
```

```
F:\JavaPrograms>
```



Type here to search



ENG

9. Write a program to convert primitive into objects (HINT: int to Integer (Autoboxing)).

Code: -

```
class WrapperEx
{
    public static void main(String args[])
    {
        int a=10;
        Integer i=Integer.valueOf(a);
        Integer J=a;
        System.out.println(a+ " "+i);
    }
}
```

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text:  
F:\JavaPrograms>javac WrapperEx.java  
F:\JavaPrograms>java WrapperEx  
10 10  
F:\JavaPrograms>

The window has a dark background and light-colored text. The title bar is at the top, and the command history is visible at the bottom. The taskbar at the bottom of the screen includes icons for File Explorer, Task View, Edge browser, File Explorer, Word, and Excel.

10. Write a program to convert object into primitives (HINT: Integer to int (Unboxing)).

```
//Program to convert object primitives(unboxing)
```

```
class Primitives
```

```
{  
public static void main(String args[])  
{  
    Integer a=new Integer(20);  
    Integer b=new Integer(30);  
    int j=a;  
    int k=b;  
    System.out.println("Primitive values");  
    System.out.println("int values" +j);  
    System.out.println("int values"+k);  
}  
}
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window contains the following text:

```
F:\JavaPrograms>javac Primitives.java  
F:\JavaPrograms>java Primitives  
Primitive values  
int values20  
int values30  
F:\JavaPrograms>
```

The window has a standard Windows title bar with minimize, maximize, and close buttons. The taskbar at the bottom shows several pinned icons, including File Explorer, Edge browser, Task View, and others. The system tray in the bottom right corner displays the date and time (19-05-2021, 17:34), battery status (20%), and network connection.

11. write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.*;  
  
abstract class Shape {  
    int length, breadth, radius;  
  
    Scanner input = new Scanner(System.in);  
  
    abstract void printArea();  
  
}  
  
class Rectangle extends Shape {  
    void printArea() {  
        System.out.println("*** Finding the Area of Rectangle ***");  
        System.out.print("Enter length and breadth: ");  
    }  
}
```

```
        length = input.nextInt();

        breadth = input.nextInt();

        System.out.println("The area of Rectangle is: " + length * breadth);

    }

}
```

```
class Triangle extends Shape {

    void printArea() {

        System.out.println("\n*** Finding the Area of Triangle ***");

        System.out.print("Enter Base And Height: ");

        length = input.nextInt();

        breadth = input.nextInt();

        System.out.println("The area of Triangle is: " + (length * breadth) / 2);

    }

}
```

```
class Cricle extends Shape {

    void printArea() {

        System.out.println("\n*** Finding the Area of Cricle ***");

        System.out.print("Enter Radius: ");

        radius = input.nextInt();


```

```
        System.out.println("The area of Cricle is: " + 3.14f * radius * radius);

    }

}

public class AbstractClassExample {

    public static void main(String[] args) {

        Rectangle rec = new Rectangle();

        rec.printArea();

        Triangle tri = new Triangle();

        tri.printArea();

        Cricle cri = new Cricle();

        cri.printArea();

    }

}
```

A screenshot of a Microsoft Windows Command Prompt window titled "Command Prompt". The window shows the following text:

```
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SRP>f:
F:>cd javaprograms
F:\JavaPrograms>javac AbstractClassExample.java
F:\JavaPrograms>java AbstractClassExample
*** Finding the Area of Rectangle ***
Enter length and breadth: 8 9
The area of Rectangle is: 72

*** Finding the Area of Triangle ***
Enter Base And Height: 9
8
The area of Triangle is: 36

*** Finding the Area of Circle ***
Enter Radius: 9
The area of Circle is: 254.34
F:\JavaPrograms>
```

Q12. There is an abstract class called Employee. This class contains one abstract method called calculatePay,. There are three subclasses that inherit behaviour and data attributes from the Employee class. Write a program to calculate earnings of an employee based on category of employee like hourly Employee, salaried Employee on weekly basis or employed on monthly basic.

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

```
abstract class Employee
```



```
{  
void calculatePay()  
{  
int week, weeklyrate=600;  
System.out.println("Salary calculation of SalariedEmployee");  
  
System.out.println("Enter how many months SalariedEmployee has worked");  
week=input.nextInt();  
earnings=week*weeklyrate;  
System.out.println("The salary of SalariedEmployee is"+earnings+ "Rs");  
}  
  
}  
  
class ExemptEmployee extends Employee  
{  
void calculatePay()  
{  
int month, monthlyrate=8000;  
System.out.println("Salary calculation of ExemptEmployee");  
  
System.out.println("Enter how many months ExemptEmployee has worked");
```

```
month=input.nextInt();

earnings=month*monthlyrate;

System.out.println("The salary of ExemptEmployee is"+earnings+ "Rs");

}

}

public class AbstractEmployeeSalary

{

    public static void main(String args[])

    {

        HourlyEmployee H=new HourlyEmployee();

        H.calculatePay();

        SalariedEmployee S=new SalariedEmployee();

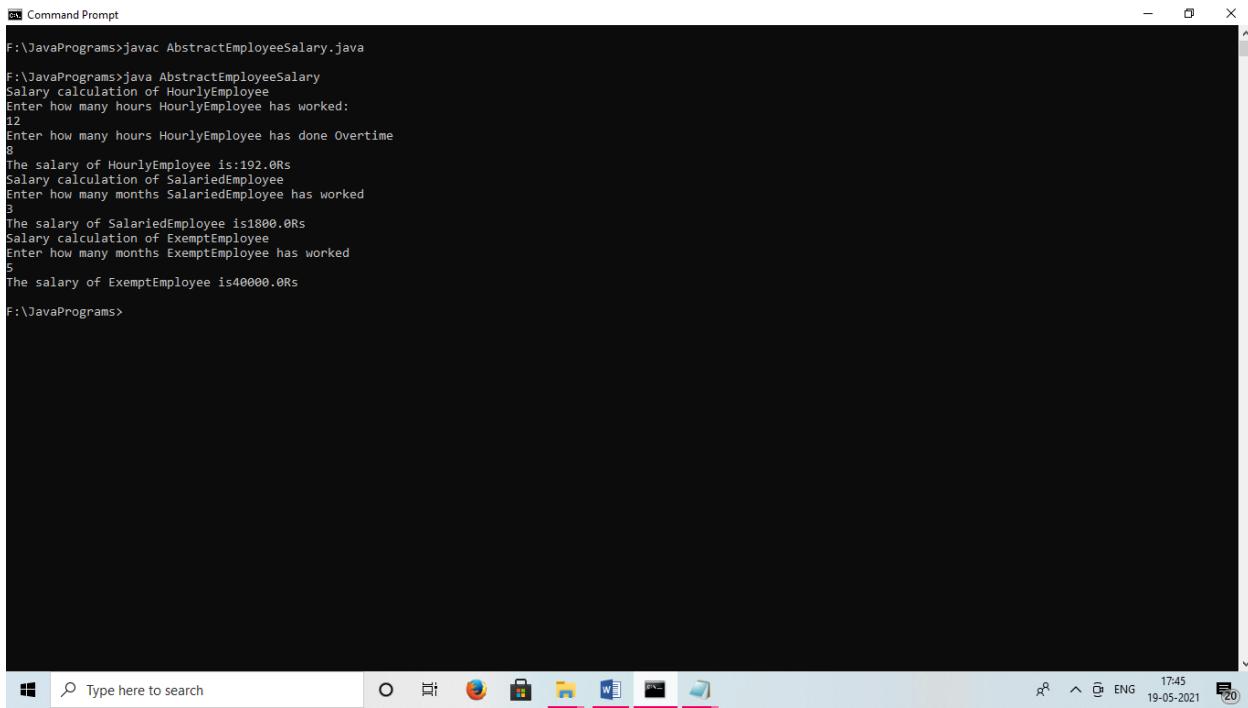
        S.calculatePay();

        ExemptEmployee e=new ExemptEmployee();

        e.calculatePay();

    }

}
```



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window contains the following text output:

```
F:\JavaPrograms>javac AbstractEmployeeSalary.java
F:\JavaPrograms>java AbstractEmployeeSalary
Salary calculation of HourlyEmployee
Enter how many hours HourlyEmployee has worked:
12
Enter how many hours HourlyEmployee has done Overtime
9
The salary of HourlyEmployee is:192.0Rs
Salary calculation of SalariedEmployee
Enter how many months SalariedEmployee has worked
5
The salary of SalariedEmployee is1800.0Rs
Salary calculation of ExemptEmployee
Enter how many months ExemptEmployee has worked
5
The salary of ExemptEmployee is40000.0Rs
F:\JavaPrograms>
```

The taskbar at the bottom shows various pinned icons and the system tray with the date and time (19-05-2021, 17:45).

Q13. Write a java application for Bank, performing following operation:

- i. Login Credentials for Bank Officials
- ii. Opening new account in Bank with atleast 5 details ( Account No, Name, Address, City, Mobile No)
- iii. Deposit
- iv. Withdraw
- v. Mini Statement (Last transactions)
- vi. Close Account
- vii. Exit

```
import java.util.Scanner;

public class BankAccount

{
```

```
{  
  
Scanner in=new Scanner(System.in);  
  
int userChoice;  
  
boolean quit=false;  
  
float balance = 0f;  
  
do{  
  
    System.out.println("1.Login credentials for bank office");  
  
    System.out.println("2. Opening a new account in bank");  
  
    System.out.println("3. deposite");  
  
    System.out.println("4. Withdraw Money");  
  
    System.out.println("5. Transaction");  
  
    System.out.println("your choice, 6 to quit:");  
  
  
    userChoice=in.nextInt();  
  
    switch(userChoice)  
  
    {  
  
        case 1:  
  
            String username,pass;  
  
            System.out.println("welcome to bank");  
  
            System.out.println("Enter your Username");  
  
            username=in.next();
```

```
System.out.println("Enter your password");
pass=in.next();
System.out.println("Login successfull");
```

case 2:

```
int acco, mobile;
String name, address, city;
System.out.println("Welcome to this bank");
System.out.println("Enter your account number");
acco=in.nextInt();
System.out.println("Enter your mobile number");
mobile=in.nextInt();
System.out.println("Enter your Name");
name=in.nextLine();
System.out.println("Enter your Address");
address=in.nextLine();
System.out.println("Enter your City");
city=in.nextLine();
System.out.println("Your account has been created");
break;
```

case 3:

```
int amount;
```

```
System.out.println("Enter amount you want to deposite");
amount=in.nextInt();

if(amount<0)

{
    System.out.println("Enter valide amount");

}

else{

    balance=balance+amount;

    System.out.println("deposite amount in your account is"+amount);

}System.out.println("current ammount in your account is"+balance);
```

case 4:

```
System.out.println("Enter amount you want to withdraw is");

amount=in.nextInt();

if(amount<0 || amount>balance)

{
    System.out.println("Enter a valide amount");

}

else{

    balance=balance-amount;

    System.out.println("Withdraw amount in your account is"+amount);

    System.out.println("Current balance in your account is"+balance);
```

```
    }

    break;

case 5:

    System.out.println("Your last transactions are");

    System.out.println("current balance is"+balance);

    break;

case 6:      quit = true;

    break;

default:

    System.out.println("Wrong choice:");

    break;

}

while(!quit);

}

}
```

Command Prompt - java BankAccount

```
F:\JavaPrograms>javac BankAccount.java

F:\JavaPrograms>java BankAccount
1.Login credentials for bank office
2. Opening a new account in bank
3. deposite
4. Withdraw Money
5. Transaction
your choice, 6 to quit:
1
welcome to bank
Enter your Username
kiran
Enter your password
kiran
Login successfull
Welcome to this bank
Enter your account number
4785961
Enter your mobile number
8596742
Enter your Name
Enter your Address
Basmat
Enter your City
basmat
Your account has been created
1.Login credentials for bank office
2. Opening a new account in bank
3. deposite
4. Withdraw Money
5. Transaction
your choice, 6 to quit:
5
Your last transactions are
current balance is0.0
1.Login credentials for bank office
2. Opening a new account in bank
3. deposite
4. Withdraw Money
5. Transaction
your choice, 6 to quit:
```



19-0

## Assignment No. - 2

Q1) Write a program to search an element using binary search.

```
import java.util.Scanner;

public class BinarySearch1
{
    public static void binarySearch( int arr[], int first, int last, int key)
    {
        int mid=(first+last)/2;

        while(first<=last)

        {
            if (arr[mid]<key)

            {
                first=mid+1;
            }

            else if(arr[mid]==key)

            {
                System.out.println("Element is found at index" +mid);
                break;
            }

            else{
                last=mid-1;
            }
        }

        mid=(first+last)/2;
    }
}
```

```
if(first>last)
{
    System.out.println("Element is not found");
}

}

public static void main(String args[])
{
    int temp;
    Scanner s=new Scanner(System.in);
    System.out.println("Enter a size of element");
    int size=s.nextInt();
    int arr[]=new int[size];
    System.out.println("Enter a element of an array");
    for(int i=0; i<arr.length;i++)
    {
        arr[i]=s.nextInt();
    }
    for(int i=0; i<arr.length;i++)
    {
        for(int j=i+1; j<arr.length;j++)
        {
            if(arr[i]>arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}
```

```
arr[j]=temp;  
}  
}  
}  
  
System.out.println("Element of an array");  
  
for(int i=0; i<arr.length; i++)  
{  
    System.out.println(""+arr[i]);  
}  
  
System.out.println("Enter a key");  
  
int key=s.nextInt();  
  
int last=arr.length-1;  
  
binarySearch(arr, 0, last,key);  
}  
}
```

```
Command Prompt
F:\JavaPrograms> javac BinarySearch1.java

F:\JavaPrograms>java BinarySearch1
Enter a size of element
4
Enter a element of an array
45
41
5
65
Element of an array
5
41
45
65
Enter a key
41
Element is found at index1

F:\JavaPrograms>
```



Q2) Write a program to accept string as command line argument and store and display string in reverse order.

```
public class StringReverse {  
    public static void main(String args[])  
    {  
        String value="",reverse = "";  
        int length;  
        for(int i=0;i<args.length;i++)  
        {  
            value =value+" "+args[i];  
        }  
        length = value.length();  
        for (int i = length - 1 ; i >= 0 ; i--)  
        {  
            reverse = reverse + value.charAt(i);  
        }  
        System.out.println("Reverse of the string: " + reverse);  
    }  
}
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command entered is "java StringReverse hello this is java program". The output displayed is "Reverse of the string: margorp avaj si siht olleh". The window has a standard title bar and a black background. The taskbar at the bottom shows various pinned icons and the date/time "19-03-2021 21:02".

```
F:\JavaPrograms>java StringReverse hello this is java program
Reverse of the string: margorp avaj si siht olleh
F:\JavaPrograms>
```

Q3) Write a program which will read string and substring from user and find occurrence of substring in a string.

```
import java.util.Scanner;

class SubString1

{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a string:");
    }
}
```

```
String str=s.nextLine();

System.out.println("Enter a sub string:");

String str1=s.nextLine();

int lastindex=0;

int count=0;

while(lastindex!=-1)

{

lastindex=str.indexOf(str1,lastindex);

if(lastindex!=-1)

{

count++;

lastindex+=str1.length();

}

}

System.out.println("Occurrences of sub string:"+count);

}

}
```

```
F:\JavaPrograms>java SubString
Enter a string:
java
Enter a sub string:
ppt
Occurrences of sub string in main string:0
F:\JavaPrograms>
```

Q4) Define a class Student having data members name, roll no and percentage. Accept for five objects and display the merit list containing merit number, roll no and percentage by using array of objects. Display data in tabular format.

```
import java.util.Scanner;  
  
class Student {  
  
    String sName;  
  
    int sRollNo;  
  
    int sPercent;  
  
    int meritNo;
```

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

Student()
{
}

Student(String sName,int sRollNo,int sPercent)
{
    this.sName=sName;
    this.sRollNo=sRollNo;
    this.sPercent=sPercent;
}

public void accept()
{

    System.out.println("Enter Student Name: ");
    sName=sc.next();
    System.out.println("Enter Student Roll No: ");
    sRollNo=sc.nextInt();
    System.out.println("Enter Student Percentage: ");
    sPercent=sc.nextInt();
}

public void calMerit(double sPercent,int meritNo)
{
```

```
}

public static void display(Student []b){

    int temp=0;

    for(int i=0;i<b.length;i++){

        {

            for(int j=i+1;j<b.length;j++){

                if(b[i].sPercent< b[j].sPercent)

                    {

                        temp=b[i].sPercent;

                        b[i].sPercent=b[j].sPercent;

                        b[j].sPercent=temp;

                    }

            }

        }

    }

    int merit=0;

    System.out.println("\n*****Merit List*****");System.out.println("\nMerit No.\t\tRoll
No.\t\tPercentage");

    for(int i=0;i<b.length;i++)

    {

        merit++;

    }
```

```
System.out.println("\n"+merit+"\t\t"+b[i].sRollNo+"\t\t"+b[i].sPercent);

}

}

public static void main(String[] args)

{

Student[] s=new Student[5];

for(int i=0;i<s.length;i++)

{

s[i]=new Student();

s[i].accept();

}

display(s);

}

}
```

```

C:\ Command Prompt
Enter Student Roll No:
22
Enter Student Percentage:
88
Enter Student Name:
Ram
Enter Student Roll No:
23
Enter Student Percentage:
67
Enter Student Name:
santosh
Enter Student Roll No:
24
Enter Student Percentage:
65
Enter Student Name:
umesh
Enter Student Roll No:
25
Enter Student Percentage:
65
Enter Student Name:
kanha
Enter Student Roll No:
26
Enter Student Percentage:
98

****Merit List****

Merit No.          Roll No.          Percentage
1                  22                 98
2                  23                 88
3                  24                 67
4                  25                 65
5                  26                 65

```

F:\JavaPrograms>

Q6) Define a class Employee having private members – id, name, department, salary. Define default and parameterized constructors. Create a subclass called “Manager” with private member bonus. Define methods accept and display in both the classes. Create n objects of the Manager class and display the details of the manager having the maximum total salary (salary+bonus)

```
import java.util.*;

class Employee1

{

    private int id;

    private String nm;

    private String dept;

    private double salary;

    double total;

    Scanner sc=new Scanner(System.in);

    Employee1()

    {

    }

    Employee1(int id,String nm,String dept,double salary)

    {

        this.id=id;

        this.nm=nm;

        this.dept=dept;

        this.salary=salary;

    }

    public void display()
```

```
    }

    System.out.println("\nEmployee Details:\nEmployee ID:"+id+"\nEmployee
Name:"+nm+"\nDepartment:"+dept+"\nSalary: "+salary);

}

public void accept()

{

    System.out.println("Enter Id:");

    id=sc.nextInt();

    System.out.println("Enter Name:");

    nm=sc.next();

    System.out.println("Enter Dept:");

    dept=sc.next();

    System.out.println("Enter Salary:");

    salary=sc.nextDouble();

}

public double calSal()

{

    total=total+salary;

    return total;

}
```

```
public class Manager extends Employee1

{
    private double bonus;

    public Manager()

    {
        super();
    }

    public Manager(int id, String nm, String dept, double salary,double bonus) {
        super(id, nm, dept, salary);

        this.bonus=bonus;
    }

    public void accept() {
        super.accept();

        System.out.println("Enter Bonus Amount: ");

        bonus=sc.nextDouble();

        super.total=bonus;
    }

    public void display() {
        super.display();

        System.out.println("\nBonus: "+bonus+"\nTotal Salary: "+total);
    }
}
```

```
public static void calMax(Manager [] m)
{
    double src=m[0].total;
    int temp=0;
    for(int i=1;i<m.length;i++){
        if(src<m[i].total)
        {
            src=m[i].total;
            temp=i;
        }
    }
    System.out.println("The Employee having the maximum Total salary is :");
    m[temp].display();
}

public static void main(String[] args)
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the number of employees: ");
    int n=sc.nextInt();
    Manager[] m=new Manager[n];
    for(int i=0;i<n;i++)
    {
```

```
m[i]=new Manager();  
m[i].accept();  
m[i].calSal();  
}  
calMax(m);  
}  
}
```

```
Command Prompt  
Enter Name: ram  
Enter Dept: 2  
Enter Salary: 522  
Enter Bonus Amount: 522  
Enter Id: 5  
Enter Name: rohit  
Enter Dept: 4  
Enter Salary: 500  
Enter Bonus Amount: 500  
The Employee having the maximum Total salary is :  
Employee Details:  
Employee ID:4  
Employee Name:ram  
Department:2  
Salary: 522.0  
Bonus: 522.0  
Total Salary: 1044.0  
F:\JavaPrograms>
```

Q7) Write a program to define a class Employee with employee id, name. Derive class SalaryInfo with designation, salary. Accept data for 5 persons and display the name of employee having salary greater than 5000 .

```
import java.util.Scanner;

class Employee2 {

    int eID;
    String eName;
    Scanner sc=new Scanner(System.in);

    Employee2(){
        }

    Employee2(int eID,String eName)
    {
        this.eID=eID;
        this.eName=eName;
    }

    public void accept()
```

```
{  
    System.out.println("Enter Id:");  
    eID=sc.nextInt();  
    System.out.println("Enter Name:");  
    eName=sc.next();  
}  
  
public void display()  
{  
    System.out.print(eID+"\t\t"+eName);  
}  
  
}  
  
public class SalaryInfo extends Employee2  
{  
    int eSal;  
    String designation;  
    SalaryInfo()  
    {  
        super();  
    }  
}
```

```
SalaryInfo(int eID, String eName, int eSal, String designation)

{
    super(eID, eName);
    this.eSal = eSal;
    this.designation = designation;
}

public void Accept()

{
    Scanner s1 = new Scanner(System.in);

    super.accept();
    System.out.println("Enter Salary :");

    eSal = s1.nextInt();

    System.out.println("Enter Designation :");

    designation = s1.next();
}

public void display()

{
```

```
super.display();

System.out.println("\t"+eSal+"\t\t"+designation);

}

static void maxSalary(SalaryInfo s[])

{

System.out.println("Id\tName :Date :Salary\tDesignatio
n");

for(int i=0;i<s.length;i++)

{

if(s[i].eSal>5000)

{

s[i].display();

}

}

}

public static void main(String args[])

{

System.out.println("Enter the 5 Entry of Employee :");
```

```
SalaryInfo s[]=new SalaryInfo[5];
for(int i=0;i<5;i++)
{
    s[i]=new SalaryInfo();
    s[i].Accept();
}
SalaryInfo.maxSalary(s);
}
```

```
Command Prompt
Enter Name:
ram
Enter Salary :
20000
Enter Designation :
Manager
Enter Id:
2
Enter Name:
Rohit
Enter Salary :
30000
Enter Designation :
Manager
Enter Id:
3
Enter Name:
Rohini
Enter Salary :
5000
Enter Designation :
web developer
Enter Id:
4
Enter Name:
Ramesh
Enter Salary :
60000
Enter Designation :
Manager
Enter Id:
5
Enter Name:
Rahul
Enter Salary :
4000
Enter Designation :
software developer
Id      Name :      Date :      Salary      Designation
1        ram    20000    Manager
2       Rohit   30000    Manager
4       Ramesh  60000    Manager
F:\JavaPrograms>
```

Q9) Create a base class called vehicle which contains properties called color, wheels. Create a child class car and which has properties called model\_no & make. Use the object of the child class which will define the different properties of a Car.

```
import java.util.*;
class Vehicle {
    String color;
```

```
int wheels;  
  
Vehicle(String clr,int wh){  
  
color=clr;  
  
wheels=wh;  
  
}  
  
}  
  
public class Car extends Vehicle {  
  
int modleNo;  
  
int make;  
  
Car(String clr, int wh,int mdl,int mk) {  
  
super(clr, wh);  
  
modleNo=mdl;  
  
make=mk;  
  
}  
  
public void display()  
{  
  
System.out.println("The Details Of the Car as Follows:\nColour:  
"+color+"Wheels: "+wheels+"\nModle  
No:"+modleNo+"\nMake"+make);
```

```
}

public static void main(String[] args) {
    Scanner s=new Scanner(System.in);
    System.out.println("Enter Colour of car:");
    String cl=s.next();
    System.out.println("Enter How Many Wheels:");
    int wh=s.nextInt();
    System.out.println("Enter Model No of car:");
    int mdn=s.nextInt();
    System.out.println("Enter Year OF Making Car:");
    int mk=s.nextInt();
    Car c=new Car(cl,wh-mdn,mk);
    c.display();
}

}
```

```
Command Prompt
F:\JavaPrograms>javac Car.java
F:\JavaPrograms>java Car
Enter Colour of car:
red
Enter How Many Wheels:
4
Enter Model No of car:
6589
Enter Year Of Making Car:
2020
The Details Of the Car as Follows:
Colour: redWheels: 4
Model No:6589
Make2020
F:\JavaPrograms>
```



## **Assignment No. - 3**

### **1. Write a program to practice string class and its methods.**

```
class StringMe
{
    public static void main(String args[])
    {
        String first="java";
        System.out.println("First String: "+first);
        String second="programming";
        System.out.println("second string: "+second);
        String joinString = first.concat(second);
        System.out.println("Joined String : " +joinString);
        boolean equal= first.equals(second);
        System.out.println("Equal string : " +equal);
        int comp=first.compareTo(second);
        System.out.println("Compare To String :" +comp);
    }
}
```

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\SRP>f:
F:>cd JavaPrograms
F:\JavaPrograms>javac StringMe.java
F:\JavaPrograms>java StringMe
First String: java
second string: programming
Joined String : javaprogramming
Equal string : false
Compare To String :-6
F:\JavaPrograms>
```



## 2. write a program to sorting a given list in ascending order.

```
import java.util.*;
class SortingSt
{
    void sortStrings()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the value of n: ");
    }
}
```

```
int n = s.nextInt();

String[] str = new String[n];

System.out.println("Enter strings: ");

for(int i = 0; i < n; i++)

{

    str[i] = new String(s.next());

}

for(int i = 0; i < n; i++)

{

    for(int j = i+1; j < n; j++)

    {

        if(str[i].compareTo(str[j])>0)

        {

            String temp = str[i];

            str[i] = str[j];

            str[j] = temp;

        }

    }

}

System.out.println("Sorted list of strings is:");

for(int i = 0; i < n ; i++)

{

    System.out.println(str[i]);

}

}
```

```
}
```

```
class Driver
```

```
{
```

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

```
    {
```

```
        SortingSt obj = new SortingSt();
```

```
        obj.sortStrings();
```

```
    }
```

```
}
```

```
Command Prompt
```

```
F:\JavaPrograms>javac Driver.java
```

```
F:\JavaPrograms>java Driver
```

```
Enter the value of n:
```

```
3
```

```
Enter strings:
```

```
kiran
```

```
jyoti
```

```
rani
```

```
Sorted list of strings is:
```

```
jyoti
```

```
kiran
```

```
rani
```

```
F:\JavaPrograms>
```



**3. Program to create two interface account with two methods set() and display() and interface person with methods store() and display(). Derive a class customer from person & account. Accept name, account number, balance and display all information related to account along with interest.**

```
import java.util.Scanner;

interface account
{
    void set();
    void display();
}

interface person
{
    void store();
    void display();
}

class Customer implements account,person
{
    Scanner s=new Scanner(System.in);
    String nm;
    long accno;
    int bal;
    double rate;
    double interest;
```

```
public void store()
{
    System.out.println("Enter a name of customer:");
    nm=s.next();
}

public void set()
{
    System.out.println("Enter a account no.:");
    accno=s.nextLong();

    System.out.println("Enter a balance:");
    bal=s.nextInt();

    System.out.println("Enter a interest rate:");
    rate=s.nextDouble();

}

public void display()
{
    System.out.println("Customer Name:"+nm);
    System.out.println("Account Number:"+accno);
    System.out.println("Account Balance:"+bal);
    interest=(bal*rate)/100;
    System.out.println("Interest:"+interest);
}

public static void main(String args[])
{
    Scanner s1=new Scanner(System.in);
```

```

Customer c=new Customer();

System.out.println("\nEnter a details of customer:");

c.store();

c.set();

System.out.println("\nAccount Details of Customer:");

c.display();

}

}

```

```

C:\ Command Prompt
F:\JavaPrograms>javac Customer.java
F:\JavaPrograms>java Customer
Enter a details of customer:
Enter a name of customer:
kiran
Enter a account no.:
32658745869
Enter a balance:
500
Enter a interest rate:
20

Account Details of Customer:
Customer Name:kiran
Account Number:32658745869
Account Balance:500
Interest:100.0
F:\JavaPrograms>

```



**4. Write a java program to find the details of the students eligible to enroll for the examination (Students, Department in combination give the eligibility criteria for the enrolment class) using interfaces.**

```
import java.util.Scanner;

interface enroll

{
    void eligible();
}

class Student1 implements enroll

{
    Scanner s=new Scanner(System.in);

    int rollno;

    String name;

    int present;

    int per;

    int fee;

    public void eligible()

    {
        System.out.println("Enter a roll no. of student:");

        rollno=s.nextInt();

        System.out.println("Enter a name of student:");
    }
}
```

```
name=s.next();

System.out.println("Enter a present percentage:");

present=s.nextInt();

System.out.println("Enter a percentage obtain in last sem:");

per=s.nextInt();

System.out.println("Enter a fees paid:");

fee=s.nextInt();

System.out.println("Student details:");

System.out.println(" Student roll no.:"+rollno);

System.out.println(" Student Name:"+name);

if(present>=75&&per>=40&&fee>=50000)

{

    System.out.println("You are elidible for exam.");

}

else

{

    System.out.println(" You are not elidible for exam.");

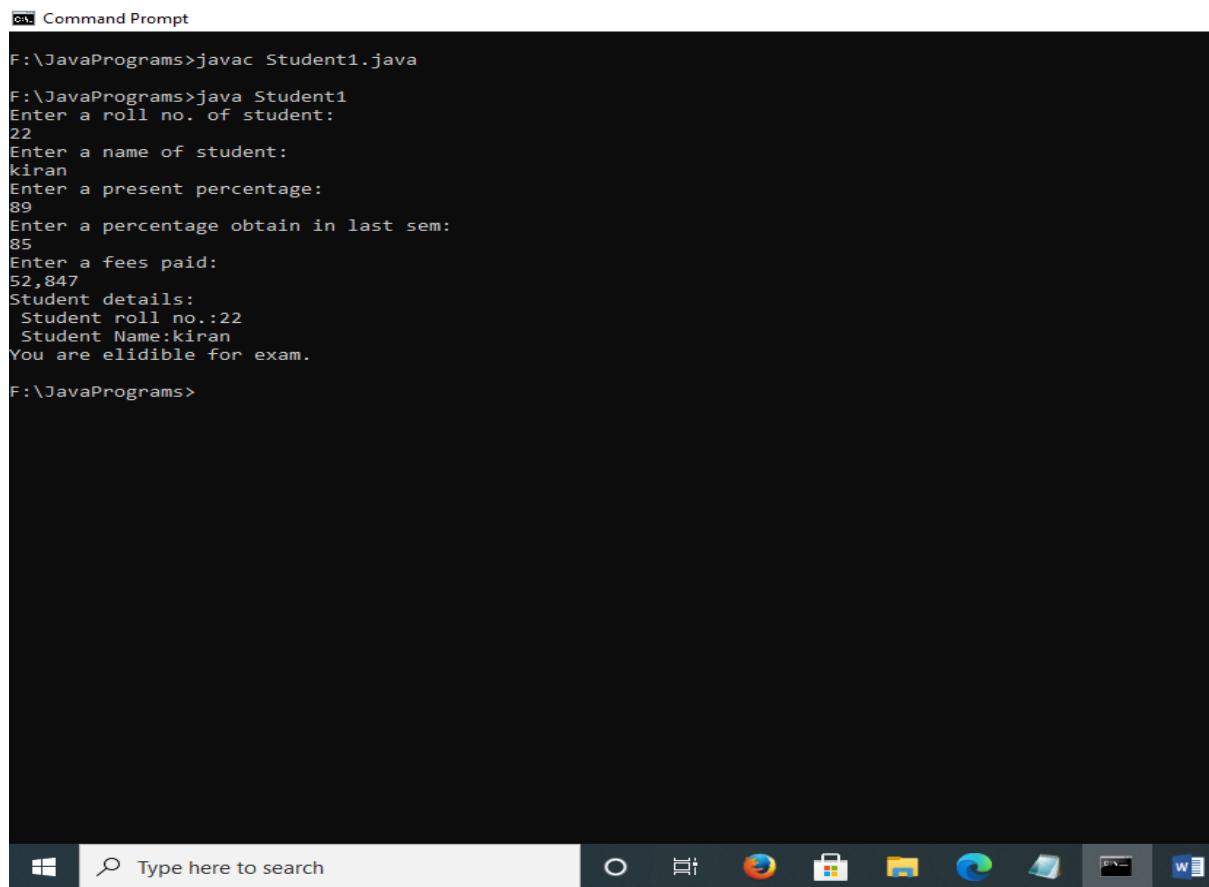
}

public static void main(String args[])

{

    Student1 st=new Student1();
```

```
        st.eligible();  
  
    }  
  
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command F:\JavaPrograms>javac Student1.java is run, followed by F:\JavaPrograms>java Student1. The program prompts for student details: roll number (22), name (kiran), present percentage (89), last semester percentage (85), and fees paid (52,847). It then displays the student details and a message indicating the student is eligible for the exam.

```
Command Prompt  
F:\JavaPrograms>javac Student1.java  
F:\JavaPrograms>java Student1  
Enter a roll no. of student:  
22  
Enter a name of student:  
kiran  
Enter a present percentage:  
89  
Enter a percentage obtain in last sem:  
85  
Enter a fees paid:  
52,847  
Student details:  
  Student roll no.:22  
  Student Name:kiran  
You are eligible for exam.  
F:\JavaPrograms>
```

**5. Create an interface called arithmetic, which defines methods for sum, multiplication, division, subtraction, percentage and implement of them in calculation class.**

```
import java.util.Scanner;  
  
interface arithmetic
```

```
{  
    void sum(int a,int b);  
    void multiplication(int x,int y);  
    void division(double p,double q);  
    void subtraction(int c,int d);  
    void percentage(double t);
```

```
}
```

```
class Calculation implements arithmetic
```

```
{
```

```
    int add,sub,mul;  
    double per,div;
```

```
    public void sum(int a,int b)
```

```
{  
    add=a+b;  
    System.out.println("Addition:"+add);
```

```
}  


```
    public void subtraction(int c,int d)
```



```
{  
    sub=c-d;  
    System.out.println("Subtraction:"+sub);
```


```

```
}

public void multiplication(int x,int y)

{

    mul=x*y;

    System.out.println("Multiplication:"+mul);

}

public void division(double p,double q)

{

    div=p/q;

    System.out.println("Division:"+div);

}

public void percentage(double t)

{

    per=t/100;

    System.out.println("Percentage:"+per+"%");

}

public static void main(String args[])

{

    Scanner s=new Scanner(System.in);

    Calculation cal=new Calculation();
```

```
System.out.println("\nAddition");

System.out.println("Enter a first Number:");

int a=s.nextInt();

System.out.println("Enter a second Number:");

int a1=s.nextInt();

cal.sum(a,a1);

System.out.println("\nSubstraction");

System.out.println("Enter a first Number:");

int s1=s.nextInt();

System.out.println("Enter a second Number:");

int s2=s.nextInt();

cal.subtraction(s1,s2);

System.out.println("\nMultiplication");

System.out.println("Enter a first Number:");

int m=s.nextInt();

System.out.println("Enter a second Number:");

int m1=s.nextInt();

cal.multiplication(m,m1);

System.out.println("\nDivision");

System.out.println("Enter a first Number");
```

```
double d=s.nextDouble();

System.out.println("Enter a second Number:");

double d1=s.nextDouble();

cal.division(d,d1);

System.out.println("Enter a Number:");

double t=s.nextDouble();

cal.percentage(t);

}

}
```

```
 Command Prompt  
F:\JavaPrograms>javac Calculation.java  
F:\JavaPrograms>java Calculation  
  
Addition  
Enter a first Number:  
5  
Enter a second Number:  
4  
Addition:9  
  
Subtraction  
Enter a first Number:  
6  
Enter a second Number:  
5  
Subtraction:1  
  
Multiplication  
Enter a first Number:  
8  
Enter a second Number:  
2  
Multiplication:16  
  
Division  
Enter a first Number:  
6  
Enter a second Number:  
2  
Division:3.0  
Enter a Number:  
5  
Percentage:0.05%  
F:\JavaPrograms>
```



**6. Create class publisher which accepts a publisher\_id and publisher\_name. Create another class book which accepts authorname and bookname. Create another class category**

**which inherits books and accepts info about category\_code and prints all details of book including publishername, pubid, authername, bookname, category\_code.**

```
import java.util.Scanner;

class Publisher
{
    int publisher_id;
    String publisher_name;
    Publisher(int publisher_id, String publisher_name)
    {
        this.publisher_id = publisher_id;
        this.publisher_name = publisher_name;
    }
    void display()
    {
        System.out.println("Publisher
Id:" + publisher_id + "Publisher Name:" + publisher_name);
    }
}

class Book
```

```
String authorname,bookname;

Book(String authorname, String bookname)

{

    this.authorname=authorname;

    this.bookname=bookname;

}

void display()

{

    System.out.println("Author

Name:"+authorname+"\nBook Name:"+bookname);

}

}

class Category extends Book

{

    int category_code;

    Category(String authorname, String bookname, int

category_code)

    {

        super(authorname,bookname);

        this.category_code=category_code;

    }

    void display()
```

```
{  
    super.display();  
    System.out.println("Category code:"+category_code);  
}  
  
public static void main(String args[])  
{  
    Scanner s=new Scanner(System.in);  
    System.out.println("Enter a Publisher Id:");  
    int pid=s.nextInt();  
    System.out.println("Enter a Publisher Name:");  
    String pnm=s.next();  
    System.out.println("Enter a Author Name:");  
    String anm=s.next();  
    System.out.println("Enter a book name:");  
    String bnm=s.next();  
    System.out.println("Enter a Category code:");  
    int ccd=s.nextInt();  
    Publisher p=new Publisher(pid,pnm);  
    p.display();  
    Category c=new Category(anm, bnm, ccd);  
    c.display();  
}
```

}

```
 Command Prompt
F:\JavaPrograms>javac Category.java
F:\JavaPrograms>java Category
Enter a Publisher Id:
4
Enter a Publisher Name:
xyz
Enter a Author Name:
kk
Enter a book name:
gd
Enter a Category code:
54
Publisher Id:4Publisher Name:xyz
Author Name:kkBook Name:gd
Category code:54
F:\JavaPrograms>
```



**7. Create an inner class Animal which has a method called eating to display eating habits. Use this method in the outer class.**

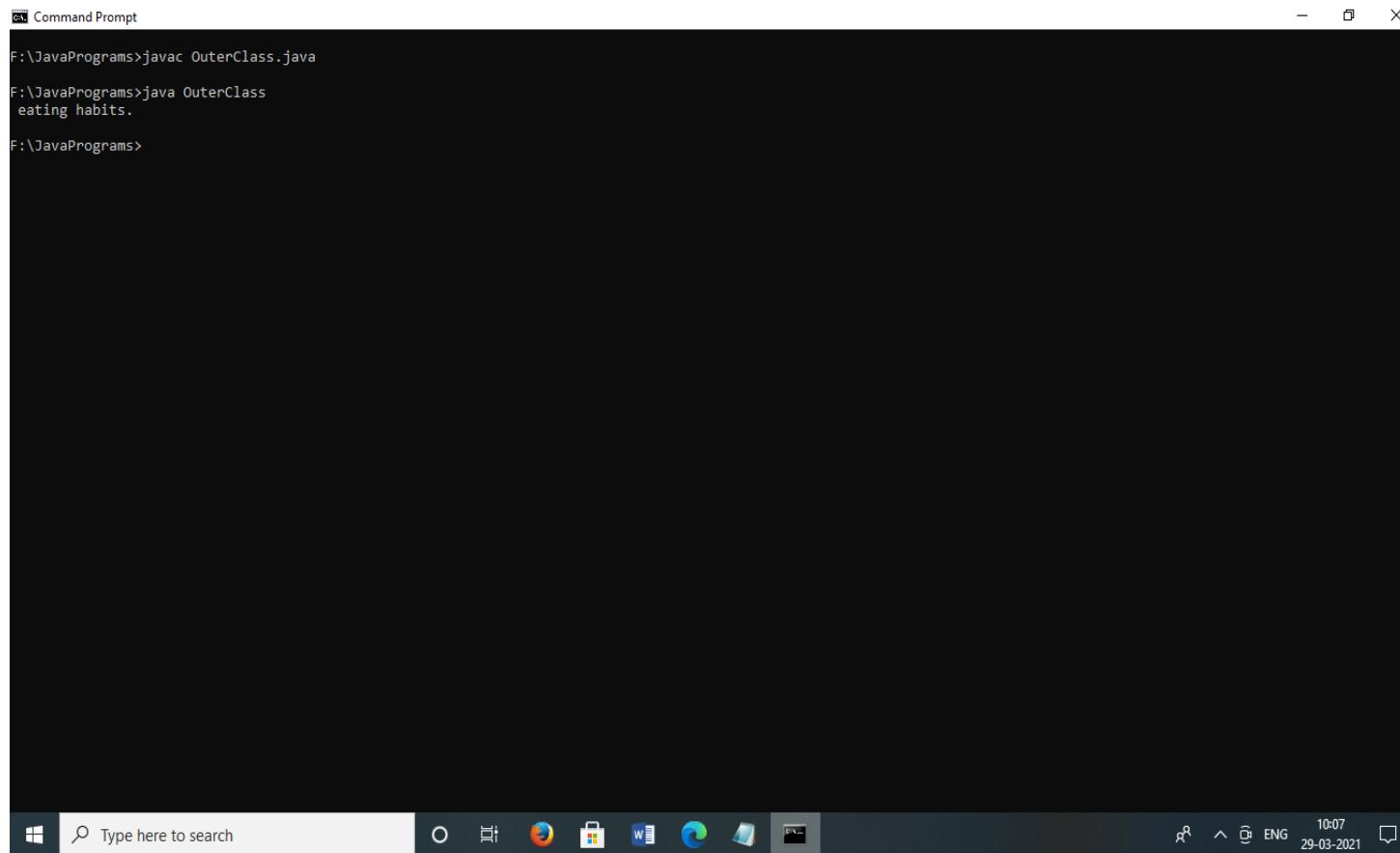
```
class OuterClass

{
    class Inner

    {
        void eating()

        {
            System.out.println(" eating habits.");
        }
    }

    public static void main(String args[])
    {
        OuterClass s=new OuterClass();
        OuterClass.Inner k=s.new Inner();
        k.eating();
    }
}
```



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text:  
F:\JavaPrograms>javac OuterClass.java  
F:\JavaPrograms>java OuterClass  
eating habits.  
F:\JavaPrograms>

The taskbar at the bottom of the screen includes icons for File Explorer, Task View, Edge browser, Microsoft Store, Word, Excel, and Powerpoint. The system tray shows the date and time as 29-03-2021 10:07.

## **8. Implement Dynamic Method Dispatch in Java using Abstract Class To find the area of various shape :Rectangle Circle Ellipse Square and Triangle.**

abstract class Area

{

    abstract void area(double a,double b);

    abstract void display();

```
}

class rectangle extends Area

{

    double a,b,ans;

    public void area(double a,double b)

    {

        this.a=a;

        this.b=b;

    }

    public void display()

    {ans=a*b;

        System.out.println("Area of rectangle:"+ans);

    }

}

class circle extends Area

{

    double a,b,ans;

    public void area(double a,double b)

    {

        this.a=a;

        this.b=b;

    }

}
```

```
}

public void display()

{

    ans=3.14*a*a;

    System.out.println("Area of circle:"+ans);

}

class ellipse extends Area

{

    double a,b,ans;

    public void area(double a,double b)

    {

        this.a=a;

        this.b=b;

    }

    public void display()

    {

        ans=3.14*a*b;

        System.out.println("Area of ellipse:"+ans);

    }

}
```

```
}
```

```
class square extends Area
```

```
{
```

```
    double a,b,ans;
```

```
    public void area(double a,double b)
```

```
    {this.a=a;
```

```
        this.b=b;
```

```
}
```

```
    public void display()
```

```
{
```

```
        ans=a*a;
```

```
        System.out.println("Area of square:"+ans);
```

```
}
```

```
}
```

```
class triangle extends Area
```

```
{
```

```
    double a,b,ans;
```

```
    public void area(double a,double b)
```

```
{
```

```
        this.a=a;
```

```
        this.b=b;  
    }  
  
    public void display()  
    {  
        ans=0.5*a*b;  
        System.out.println("Area of triangle:"+ans);  
    }  
  
}  
  
class Dynamic_Dispatch  
{  
  
    public static void main(String args[])  
    {  
        rectangle r=new rectangle();  
        circle c=new circle();  
        ellipse e=new ellipse();  
        square s=new square();  
        triangle t=new triangle();  
  
        r.area(2.5,8.4);  
        c.area(5,1);  
        e.area(6.8,9.5);  
    }  
}
```

```
s.area(7,1);

t.area(10.45,25.2);

r.display();

c.display();

e.display();

s.display();

t.display();

}

}

}
```

```
Command Prompt
F:\JavaPrograms>javac Dynamic_Dispatch.java
F:\JavaPrograms>java Dynamic_Dispatch
Area of rectangle:21.0
Area of circle:78.5
Area of ellipse:202.844
Area of square:49.0
Area of triangle:131.67
F:\JavaPrograms>
```

**9. Write a java program to accept Employee name from the user and check whether it is valid or not. If it is not valid then throw user defined Exception “Name is Invalid” otherwise display it.**

```
import java.util.*;

class InvalidNameException extends Exception

{

}

class EmployeeC

{

    public static void main(String args[])

    {

        Scanner s=new Scanner(System.in);

        System.out.println("Enter the Name");

        String name=s.next();

        try

        {

            for(int i=0;i<name.length();i++)
```

```
{  
    int ch=(int)name.charAt(i);  
  
    if((ch>=65&&ch<=90)|| (ch>=97&&ch<=122))  
    {  
    }  
    else  
    {  
        throw new InvalidNameException();  
    }  
}  
System.out.println("Employee Name is :" + name);  
}  
catch(InvalidNameException e)  
{  
    System.out.println("Invalid user Name ");  
}  
}
```

```
C:\ Select Command Prompt  
F:\JavaPrograms>javac EmployeeC.java  
F:\JavaPrograms>java EmployeeC  
Enter the Name  
kiran  
Employee Name is :kiran  
F:\JavaPrograms>
```



**10. Write a program that generates a custom exception if age entered for voting in election is less than 18 years.**

```
import java.util.Scanner;  
  
class AgeException extends Exception  
{
```

```
public AgeException(String str)

{
    System.out.println(str);
}

}

public class Voting

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

        System.out.print("Enter your age : ");

        int age = s.nextInt();

        try
        {
            if(age < 18)

                throw new AgeException("Invalid age!!");

            else

                System.out.println("Valid age...");

        }
        catch(AgeException a)
        {
    }
```

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

```
C:\ Command Prompt  
F:\JavaPrograms>javac Voting.java  
F:\JavaPrograms>java Voting  
Enter your age : 17  
Invalid age!!  
AgeException  
F:\JavaPrograms>java Voting  
Enter your age : 20  
Valid age...  
F:\JavaPrograms>_
```



**11. Define a thread called “PrintText\_Thread” for printing text on command prompt for n number of times. Create three threads and run them. Pass the text and n as parameters to the thread constructor. Also set priorities of thread Example:**

- First thread prints “Hello” 10 times

- Second thread prints “My name is .....” 20 times
- Third thread prints “I am in MCA-II” 30 times

```
class Mthread extends Thread  
{  
    String msg="";  
    int n;  
    Mthread(String msg,int n)  
    {  
        this.msg=msg;  
        this.n=n;  
    }  
    public void run()  
    {  
        try
```

```
{  
    for(int i=1;i<=n;i++)  
    {  
        System.out.println(msg+" "+i+" times");  
        Thread.sleep(1000);  
    }  
}  
catch(Exception e){}  
}  
  
}  
  
public class PrintText_Thread  
{  
    public static void main(String args[])  
    {  
        int n=Integer.parseInt(args[0]);  
        Mthread t1=new Mthread("Hello",n);  
        t1.start();  
        Mthread t2=new Mthread("My name is kiran",n+10);  
        t2.start();  
    }  
}
```

```
Mthread t3=new Mthread("I am in MCA-I",n+20);  
t3.start();  
}  
}
```

c:\ Command Prompt

```
My name is kiran 2 times
I am in MCA-I 2 times
Hello 2 times
My name is kiran 3 times
I am in MCA-I 3 times
Hello 3 times
My name is kiran 4 times
I am in MCA-I 4 times
Hello 4 times
My name is kiran 5 times
I am in MCA-I 5 times
Hello 5 times
My name is kiran 6 times
I am in MCA-I 6 times
My name is kiran 7 times
I am in MCA-I 7 times
My name is kiran 8 times
I am in MCA-I 8 times
My name is kiran 9 times
I am in MCA-I 9 times
My name is kiran 10 times
I am in MCA-I 10 times
My name is kiran 11 times
I am in MCA-I 11 times
My name is kiran 12 times
I am in MCA-I 12 times
My name is kiran 13 times
I am in MCA-I 13 times
My name is kiran 14 times
I am in MCA-I 14 times
My name is kiran 15 times
I am in MCA-I 15 times
I am in MCA-I 16 times
I am in MCA-I 17 times
I am in MCA-I 18 times
I am in MCA-I 19 times
I am in MCA-I 20 times
I am in MCA-I 21 times
I am in MCA-I 22 times
I am in MCA-I 23 times
I am in MCA-I 24 times
I am in MCA-I 25 times
```

F:\JavaPrograms>



Type here to search



**12. Write a program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the.**

```
import java.util.Random;

class RandomNumberThread extends Thread

{

    public void run()

    {

        Random random = new Random();

        for (int i = 0; i < 10; i++)

        {

            int randomInteger = random.nextInt(100);

            System.out.println("Random Integer

generated:"+randomInteger);

            if((randomInteger%2) == 0)

            {
```

```
        SquareThread sThread = new
SquareThread(randomInteger);

        sThread.start();

    }

else

{

    CubeThread cThread = new
CubeThread(randomInteger);

    cThread.start();

}

try

{

    Thread.sleep(1000);

}

catch (InterruptedException e)

{

    System.out.println(e);

}

}
```

```
}

class SquareThread extends Thread

{

    int number;

    SquareThread(int randomNumber)

    {

        number = randomNumber;

    }

    public void run()

    {

        System.out.println("Square of " + number + " = " + (number * number));

    }

}

class CubeThread extends Thread

{

    int number;

    CubeThread(int randomNumber)

    {

        number = randomNumber;

    }

}
```

```
    }

    public void run()

    {

        System.out.println("Cube of " + number + " = " +  
number * number * number);

    }

}

public class MultiThreadingTest

{

    public static void main(String args[])

    {

        RandomNumberThread rnThread = new  
RandomNumberThread();

        rnThread.start();

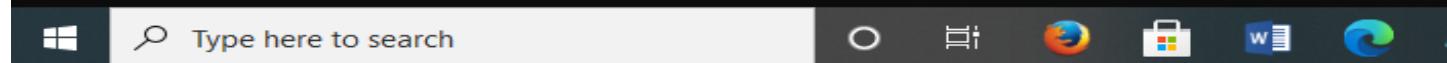
    }

}
```

Command Prompt

```
F:\JavaPrograms>javac MultiThreadingTest.java
F:\JavaPrograms>java MultiThreadingTest
Random Integer generated:8
Square of 8 = 64
Random Integer generated:24
Square of 24 = 576
Random Integer generated:19
Cube of 19 = 6859
Random Integer generated:10
Square of 10 = 100
Random Integer generated:92
Square of 92 = 8464
Random Integer generated:48
Square of 48 = 2304
Random Integer generated:56
Square of 56 = 3136
Random Integer generated:75
Cube of 75 = 421875
Random Integer generated:37
Cube of 37 = 50653
Random Integer generated:92
Square of 92 = 8464
```

```
F:\JavaPrograms>_
```



**14. Write a program to implement Producer-Consumer problem using threads.**

```
class Test

{

    private int contents;

    private boolean available = false;

    public synchronized int get()

    {

        while (available == false)

        {

            try

            {

                wait();

            }

            catch (InterruptedException e) {}

        }

        available = false;

        notifyAll();

        return contents;

    }

    public synchronized void put(int value)

    {
```

```
while (available == true)

{
    try
    {
        wait();
    }

    catch(InterruptedException e) { }

    contents = value;
    available = true;
    notifyAll();
}

}

class Consumer extends Thread

{
    private CubbyHole cubbyhole;
    private int number;
    public Consumer(CubbyHole c, int number)

    {
        cubbyhole = c;
```

```
    this.number = number;  
}  
  
public void run()  
{  
    int value = 0;  
  
    for (int i = 0; i < 10; i++)  
    {  
        value = cubbyhole.get();  
  
        System.out.println("Consumer #" + this.number + "  
got: " + value);  
    }  
}  
  
}  
  
class Producer extends Thread  
{  
    private CubbyHole cubbyhole;  
  
    private int number;  
  
    public Producer(CubbyHole c, int number)  
    {  
        cubbyhole = c;
```

```
        this.number = number;

    }

    public void run()

    {

        for (int i = 0; i < 10; i++)

        {

            cubbyhole.put(i);

            System.out.println("Producer #" + this.number +

" put: " + i);

            try

            {

                sleep((int)(Math.random() * 100));

            }

            catch (InterruptedException e) { }

        }

    }

}

public class ProducerConsumerTest

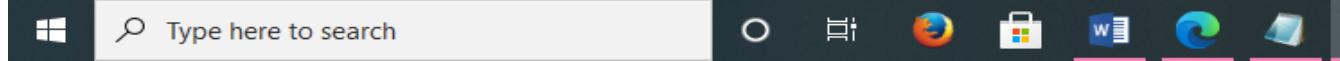
{

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

```
{  
    CubbyHole c = new CubbyHole();  
    Producer p1 = new Producer(c, 1);  
    Consumer c1 = new Consumer(c, 1);  
    p1.start();  
    c1.start();  
}  
}  
}
```

```
ct. Select Command Prompt
F:\JavaPrograms>javac ProducerConsumerTest.java
F:\JavaPrograms>java ProducerConsumerTest
Consumer #1 got: 0
Producer #1 put: 0
Producer #1 put: 1
Consumer #1 got: 1
Producer #1 put: 2
Consumer #1 got: 2
Consumer #1 got: 3
Producer #1 put: 3
Consumer #1 got: 4
Producer #1 put: 4
Consumer #1 got: 5
Producer #1 put: 5
Producer #1 put: 6
Consumer #1 got: 6
Consumer #1 got: 7
Producer #1 put: 7
Producer #1 put: 8
Consumer #1 got: 8
Producer #1 put: 9
Consumer #1 got: 9
```

```
F:\JavaPrograms>_
```



**15. Define a class Account (acNo, name, balance). Define methods - withdraw(), deposit(). Create an object and perform operations. Write a program to implement inter-thread communication.**

## **16. Write a program to demonstrate thread synchronization.**

```
class threadSync
{
    synchronized void printTable(int n)
    {
        for(int i=1;i<=5;i++)
        {
            System.out.println(n*i);
            try
            {
                Thread.sleep(400);
            }
            catch(Exception e)
            {

```

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

    }

}

}

class MyThread1 extends Thread

{

    threadSync t;

    MyThread1(threadSync t)

    {

        this.t=t;

    }

    public void run()

    {

        t.printTable(5);

    }

}

class MyThread2 extends Thread

{

    threadSync t;
```

```
MyThread2(threadSync t)

{
    this.t=t;

}

public void run()

{
    t.printTable(100);

}

}

public class TestSynchronization

{
    public static void main(String args[])

    {
        threadSync obj = new threadSync();

        MyThread1 t1=new MyThread1(obj);

        MyThread2 t2=new MyThread2(obj);

        t1.start();

        t2.start();

    }

}
```

```
Command Prompt
F:\JavaPrograms>javac TestSynchronization.java
F:\JavaPrograms>java TestSynchronization
5
10
15
20
25
100
200
300
400
500
F:\JavaPrograms>_
```



**17. Write a program to demonstrate inter communication between thread using Banking domain.**

```
class Customer
{
    int amount = 10000;
    synchronized void withdraw(int amount)
    {
```

```
System.out.println("Available Balance " + this.amount);

System.out.println("Going to withdraw." + amount);

if (this.amount < amount)

{

System.out.println("Insufficient Balance waiting for deposit.");

try

{

wait();

}

catch (Exception e)

{

System.out.println("Interruption Occured");

}

}

this.amount -= amount;

System.out.println("Detected amount: " + amount);

System.out.println("Balance amount : " + this.amount);

}

synchronized void deposit(int amount)

{

System.out.println("Going to deposit " + amount);

this.amount += amount;

System.out.println("Available Balance " + this.amount);

System.out.println("Transaction completed.\n");

notify();

}
```

```
}

class InterThreadDemo

{

public static void main(String arg[])
{
final Customer c = new Customer();

new Thread()
{
public void run()
{
c.withdraw(15000);

}
}.start();

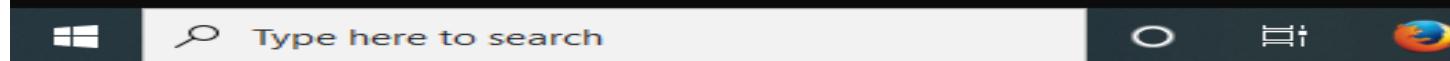
new Thread()
{
public void run()
{
c.deposit(10000);
}

}.start();

new Thread()
{
public void run()
{
c.deposit(10000);
}

}
```

```
}.start();  
}  
}  
  
C:\ Command Prompt  
Microsoft Windows [Version 10.0.18363.1440]  
(c) 2019 Microsoft Corporation. All rights reserved.  
C:\Users\SRP>f:  
F:\>cd JavaPrograms  
F:\JavaPrograms>javac InterThreadDemo.java  
F:\JavaPrograms>java InterThreadDemo  
Available Balance 10000  
Going to withdraw 15000  
Insufficient Balance waiting for deposit.  
Going to deposit 10000  
Available Balance 20000  
Transaction completed.  
  
Going to deposit 10000  
Available Balance 30000  
Transaction completed.  
  
Detected amount: 15000  
Balance amount : 15000  
F:\JavaPrograms>
```



## **Assignment No. - 4**

**1. Write a program to demonstrate try, catch and finally where code copy content of one file to another file using byte stream.**

```
import java.io.*;  
  
class CopyFile  
{  
  
    public static void main(String args[])throws IOException  
    {  
  
        int i;  
  
        FileInputStream fin;  
  
        FileOutputStream fout;  
  
        try  
        {  
  
            try  
            {  
  
                fin=new FileInputStream(args[0]);  
  
            }  
  
            catch(FileNotFoundException e)  
            {  
  
                System.out.println("Input File not found");  
  
                return;  
            }  
  
            // Read from file  
            // ...  
  
            // Write to file  
            // ...  
  
        }  
    }  
}
```

```
    }

    try

    {

        fout=new FileOutputStream(args[1]);

    }

    catch(FileNotFoundException e)

    {

        System.out.println("Error opening output file");

        return;

    }

}catch(ArrayIndexOutOfBoundsException e)

{

    System.out.println("Usage: CopyFile From To");

    return;

}

try

{

    do

    {

        i=fin.read();

        if(i!=-1)

            fout.write(i);

    }

}
```

```

        }while(i!=-1);

    }catch(IOException e)

    {

        System.out.println("File Error");

    }

    finally

    {

        System.out.println("File copied successfully");

    }

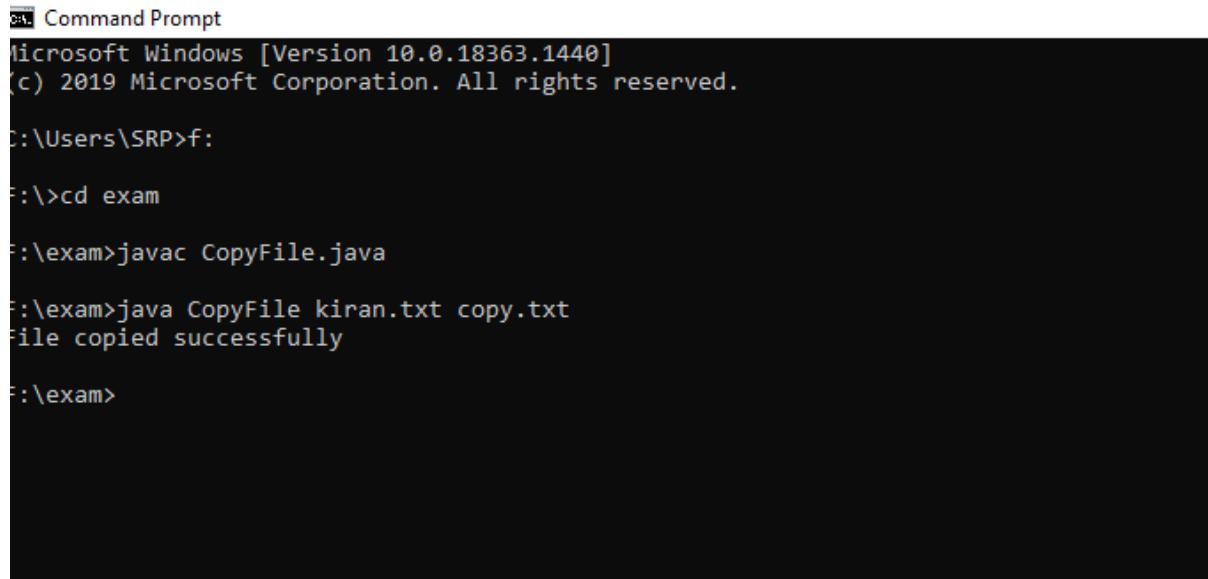
    fin.close();

    fout.close();

}

}

```



```

C:\ Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\SRP>f:

C:\>cd exam

C:\exam>javac CopyFile.java

C:\exam>java CopyFile kiran.txt copy.txt
file copied successfully

C:\exam>

```

2. Write a program to demonstrate all file related methods

```
import java.io.*;

class FileMethods{

    public static void main(String[] args)

    {

        File file = new File("FileMethodFile.txt");

        try {

            boolean value = file.createNewFile();

            if (value) {

                System.out.println("\nThe new file is created.");

            }

            else {

                System.out.println("The file already exists.");

            }

        }

        catch(Exception e) {}

        System.out.println();

        try {

            FileOutputStream out = new FileOutputStream("FileMethodFile.txt");

            String c= "This is the Java Program";

            byte b[]=c.getBytes();

            out.write(b);

            out.close();

        }

    }

}
```

```
}

catch(Exception e){}

System.out.println("Data is added to the file");

System.out.println();

File new1 = new File("new");

boolean value = file.renameTo(new1);

if(value) {

    System.out.println("The name of the file is changed.");

}

else {

    System.out.println("The name cannot be changed.");

}

System.out.println();

char[] array = new char[100];

try {

    FileReader input = new FileReader("new");

    input.read(array);

    System.out.println("Data in the file:");

    System.out.println(array);

    input.close();

}

catch(Exception e) {}
```

```
System.out.println();

boolean value1 = new1.delete();

if(value1) {

System.out.println("The File is deleted.");

}

else {

System.out.println("The File is not deleted.");

}

System.out.println();

File file1 = new File("f:/exam/kiran padole");

String[] fileList = file1.list();

System.out.println("The File in 'f:/exam/kiran padole' are: ");

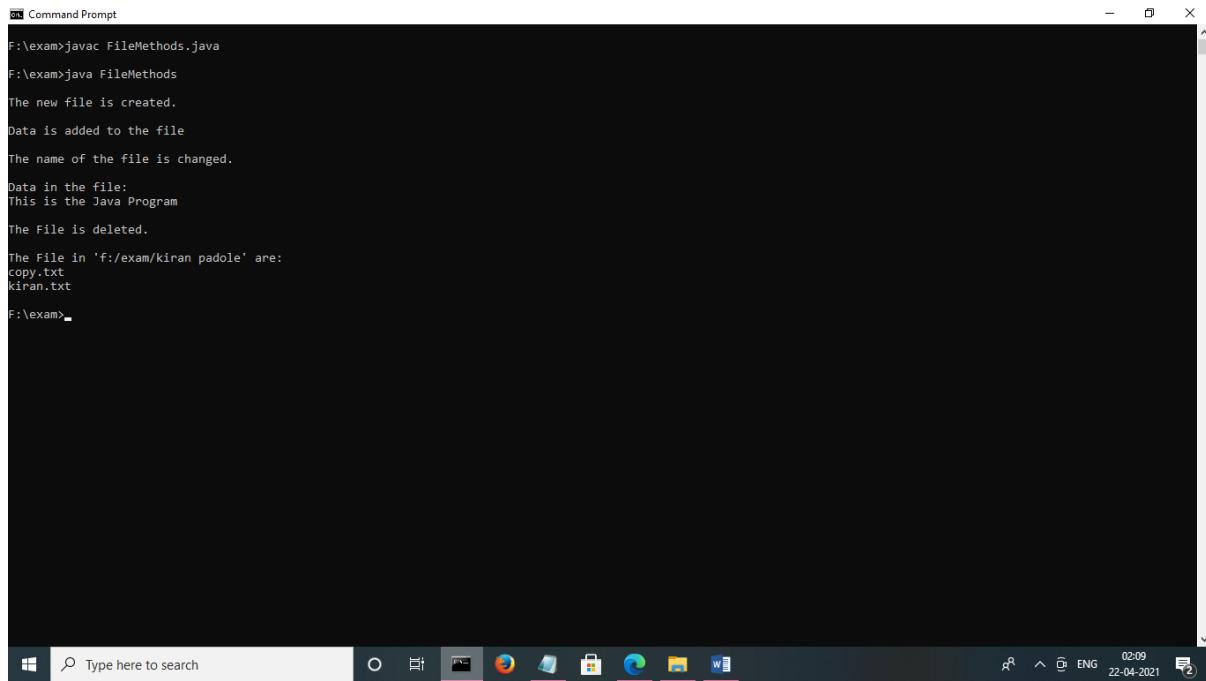
for(String str : fileList)

{

System.out.println(str);

}

}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window displays the following output:

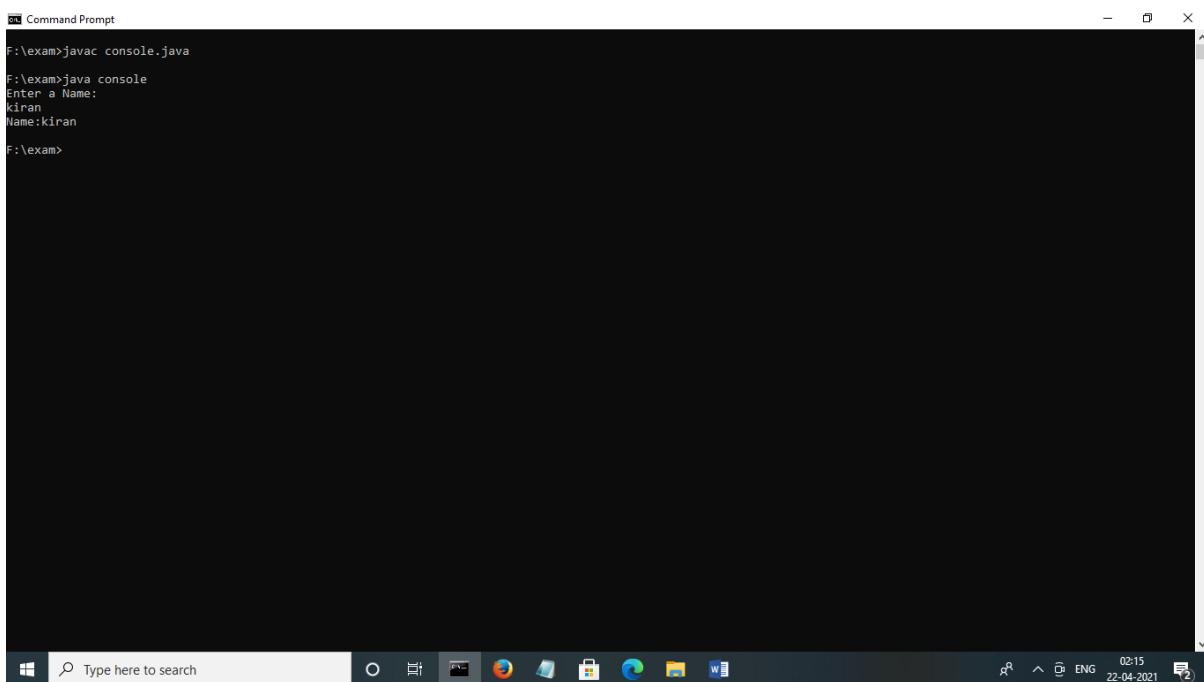
```
F:\exam>javac FileMethods.java
F:\exam>java FileMethods
The new file is created.
Data is added to the file
The name of the file is changed.
Data in the file:
This is the Java Program
The file is deleted.
The files in 'f:/exam/kiran padole' are:
copy.txt
kiran.txt
F:\exam>
```

### **3. Write a program to read from console and display the same on console using Character stream.**

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class console
{
    public static void main(String[] args) throws IOException
    {

```

```
        BufferedReader reader = new BufferedReader(new  
InputStreamReader(System.in));  
  
        System.out.println("Enter a Name:");  
  
        String name = reader.readLine();  
  
        System.out.println("Name:" + name);  
  
    }  
  
}
```



```
F:\exam>javac console.java  
F:\exam>java console  
Enter a Name:  
Kiran  
Name: Kiran  
F:\exam>
```

**4. Write a program to demonstrate the concept of serialization and deserialization where class student contains property stud\_no, stud\_name, subject and marks.**

```
import java.io.*;  
  
class SerializeDeserialize1 implements java.io.Serializable  
{  
    public int id;  
    public String sname;  
    public String sub;  
    public int marks;  
    public SerializeDeserialize1(int id, String sname ,String sub,int marks)  
    {  
        this.id = id;  
        this.sname = sname;  
        this.sub = sub;  
        this.marks = marks;  
    }  
}  
  
class SerializeDeserializeDemo1  
{  
    public static void main(String[] args)  
    {  
        SerializeDeserialize1 object = new SerializeDeserialize1(10, "kiran",  
"java" , 50);
```

```
String filename = "file.ser";

try
{
    FileOutputStream file = new FileOutputStream(filename);
    ObjectOutputStream out = new ObjectOutputStream(file);
    out.writeObject(object);
    out.close();
    file.close();
    System.out.println("Object has been serialized");
}

catch(IOException ex)
{
    System.out.println("IOException is caught");
}

SerializeDeserialize1 object1 = null;

// Deserialization

try
{
    FileInputStream file = new FileInputStream(filename);
    ObjectInputStream in = new ObjectInputStream(file);
    object1 = (SerializeDeserialize1)in.readObject();
    in.close();file.close();
}
```

```
        System.out.println("Object has been deserialized ");

        System.out.println("Id = " + object1.id);

        System.out.println("Name = " + object1.sname);

        System.out.println("Subject = " + object1.sub);

        System.out.println("Mark = " + object1.marks);

    }

    catch(IOException ex)

    {

        System.out.println("IOException is caught");

    }

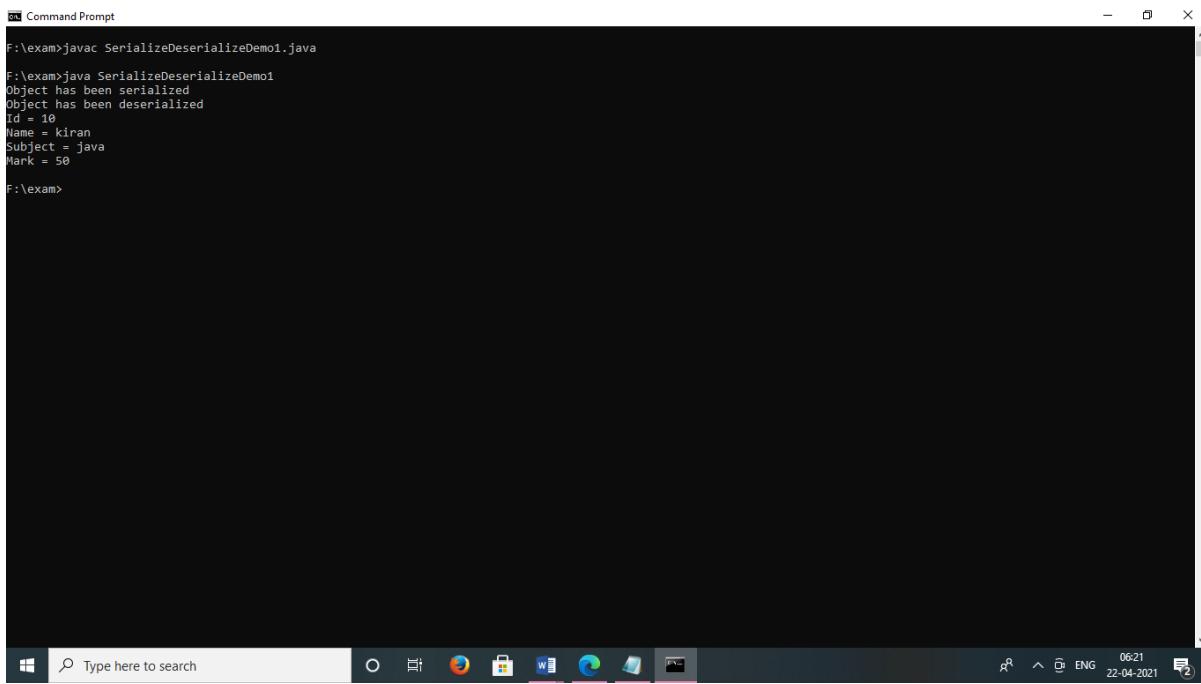
    catch(ClassNotFoundException ex)

    {

        System.out.println("ClassNotFoundException is caught");

    }

}
```



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the output of a Java program. The command entered was "javac SerializeDeserializeDemo1.java". The output indicates that the object has been serialized and then deserialized, displaying the values for Id, Name, Subject, and Mark.

```
F:\exam>javac SerializeDeserializeDemo1.java
F:\exam>java SerializeDeserializeDemo1
Object has been serialized
Object has been deserialized
Id = 10
Name = kiran
Subject = java
Mark = 50
F:\exam>
```

## 5. Write java program to read a file and separate vowels and consonants to other files.

```
import java.io.*;

public class VowelConsonant

{

    public static void main(String[] args)throws IOException

    {

        try

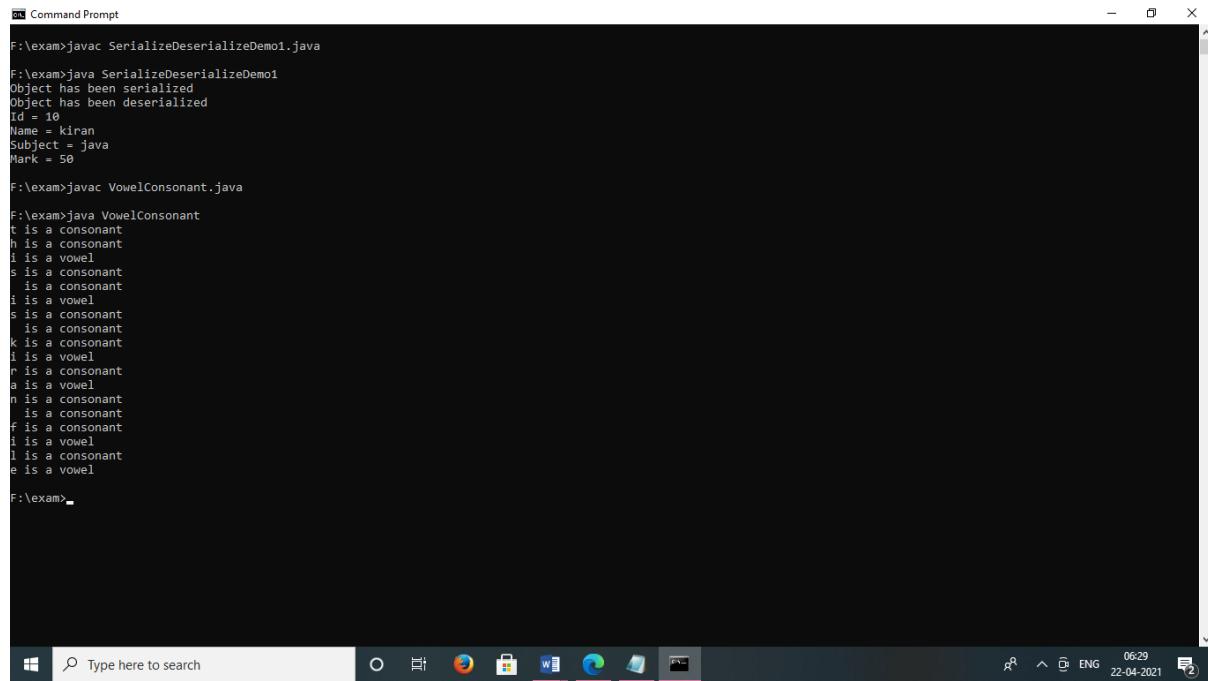
        {
```

```
FileInputStream fstream = new  
FileInputStream("f:/exam/kiran padole/kiran.txt");  
DataInputStream in = new DataInputStream(fstream);  
BufferedReader br = new BufferedReader(new  
InputStreamReader(in));  
FileWriter ostream = new  
FileWriter("f:/exam/kiran padole/vowels.txt");  
BufferedWriter out = new BufferedWriter(ostream);  
FileWriter opstream = new  
FileWriter("f:/exam/kiran padole/consonants.txt");  
BufferedWriter outt = new BufferedWriter(opstream);  
String str=br.readLine();  
char ch;  
for (int i=0;i<str.length();i++)  
{  
    ch = str.charAt(i);  
    if(( ch == 'a') ||( ch == 'e') ||( ch == 'i') ||( ch == 'o') ||( ch ==  
'u'))  
    {  
        out.write(ch);  
        System.out.println(ch+" is a vowel");  
    }  
}
```

```
        else
        {
            outt.write(ch);
            System.out.println(ch+" is a consonant");
        }
    }

    out.close();
    outt.close();
    br.close();
}

catch(Exception e)
{
    System.err.println(e);
}
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command F:\exam>javac SerializeDeserializeDemo1.java is run, followed by F:\exam>java SerializeDeserializeDemo1, which outputs "Object has been serialized" and "Object has been deserialized". Then, F:\exam>javac VowelConsonant.java is run, followed by F:\exam>java VowelConsonant, which lists various letters as either vowels or consonants. The desktop taskbar at the bottom shows icons for File Explorer, Edge, and other applications.

```
F:\exam>javac SerializeDeserializeDemo1.java
F:\exam>java SerializeDeserializeDemo1
Object has been serialized
Object has been deserialized
Id = 10
Name = kiran
Subject = java
Mark = 50

F:\exam>javac VowelConsonant.java
F:\exam>java VowelConsonant
t is a consonant
h is a consonant
i is a vowel
s is a consonant
i is a vowel
s is a consonant
i is a vowel
k is a consonant
l is a vowel
i is a vowel
r is a consonant
a is a vowel
n is a consonant
i is a consonant
f is a consonant
i is a vowel
l is a consonant
e is a vowel

F:\exam>
```

## 6. Write a program to demonstrate shallow and deep cloning.

```
import java.io.*;

class Test4

{

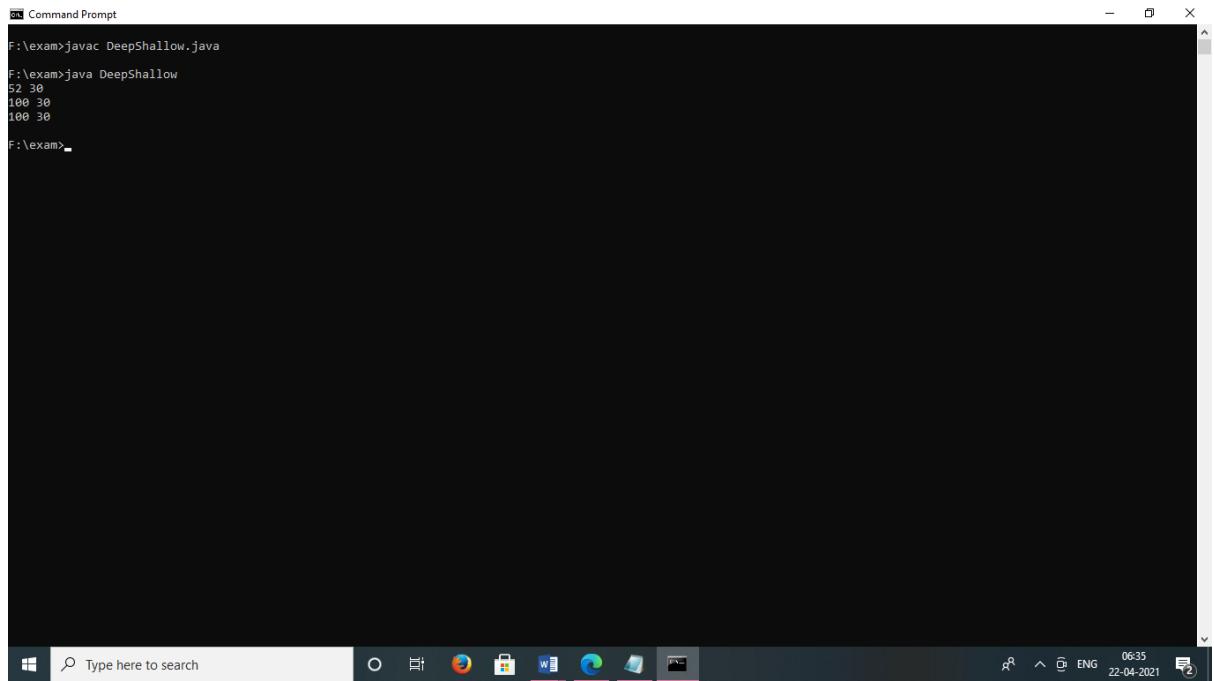
    int x,y;

    Test4()

    {

        x=52;
```

```
y=30;  
}  
}  
  
class DeepShallow  
{  
    public static void main(String args[])  
    {  
        Test4 ob1=new Test4();  
        System.out.println(ob1.x+" "+ob1.y);  
        Test4 ob2=ob1;  
        ob2.x=100;  
        System.out.println(ob1.x+" "+ob1.y);  
        System.out.println(ob2.x+" "+ob2.y);  
    }  
}
```

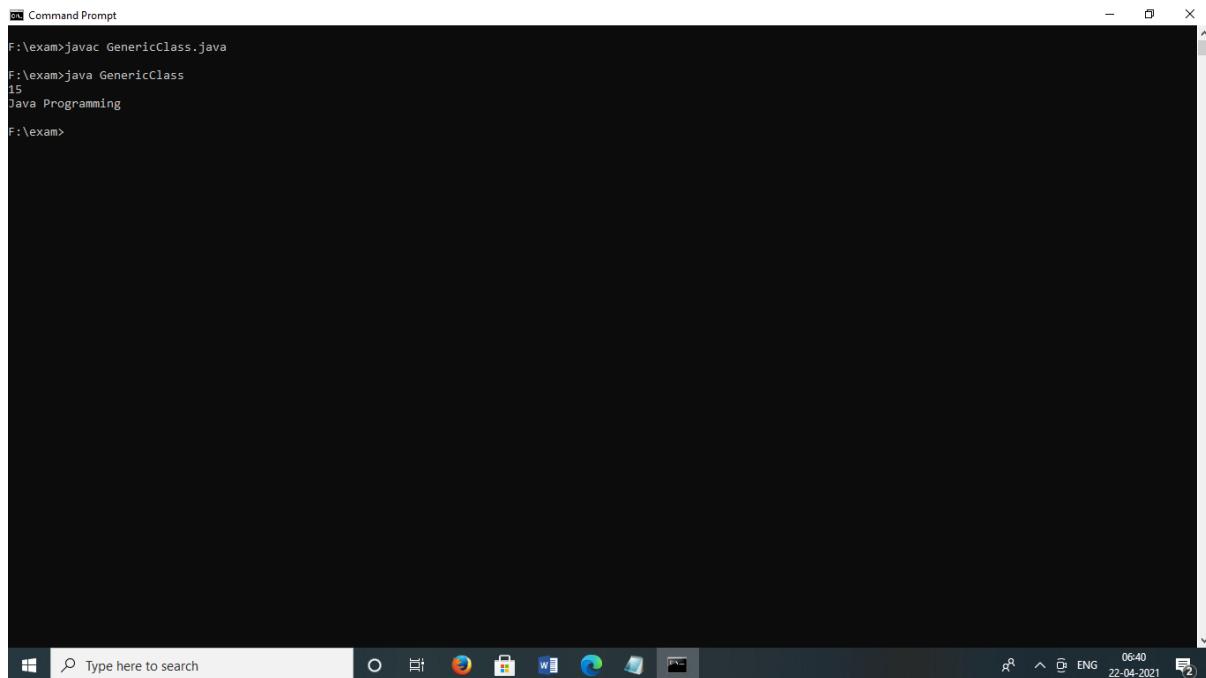


```
F:\exam>javac DeepShallow.java
F:\exam>java DeepShallow
52 30
100 30
100 30
F:\exam>
```

## 7. Write a Java program to show working of user defined Generic classes.

```
class Test<T>
{
    T obj;
    Test(T obj)
    {
        this.obj = obj;
    }
    public T getObject()
    {
```

```
return this.obj;  
}  
}  
  
class GenericClass  
{  
  
public static void main (String[] args)  
{  
  
Test <Integer> iObj = new Test<Integer>(15);  
  
System.out.println(iObj.getObject());  
  
Test <String> sObj = new Test<String>("Java Programming");  
  
System.out.println(sObj.getObject());  
}  
}
```



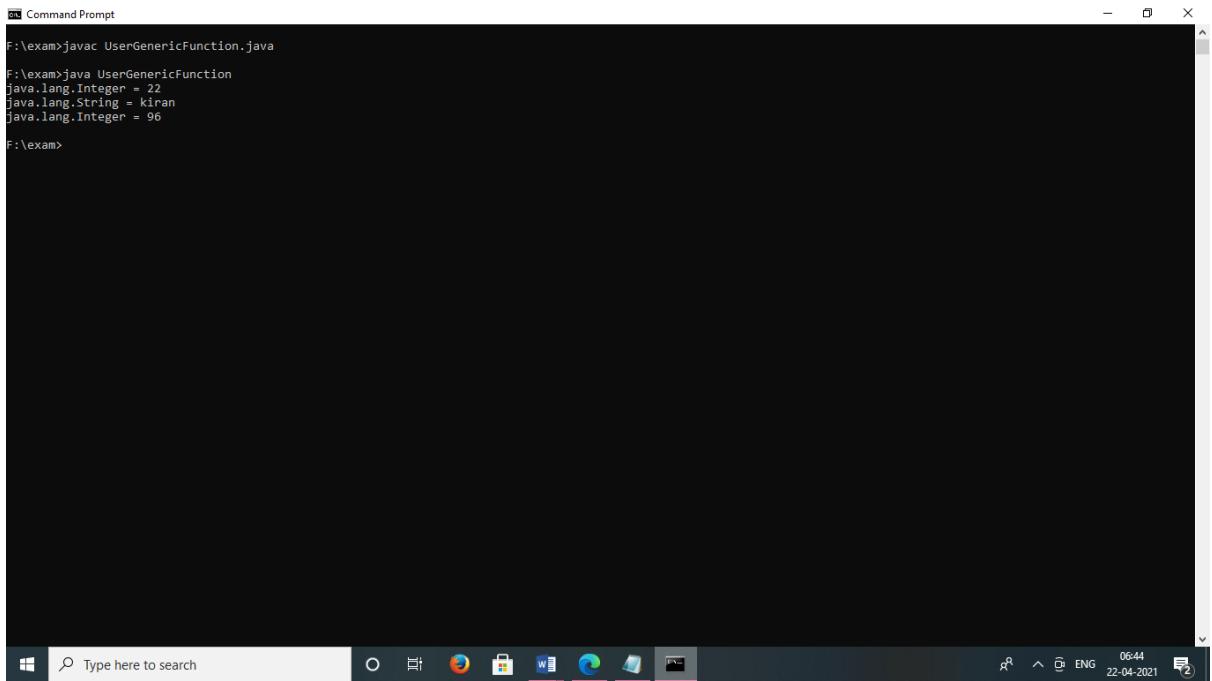
## **8. Write a Java program to show working of user defined Generic function.**

```
class UserGenericFunction

{
    static <T> void genericDisplay (T element)

    {
        System.out.println(element.getClass().getName() +" = " + element);
    }

    public static void main(String[] args)
    {
        genericDisplay(22);
        genericDisplay("kiran ");
        genericDisplay(96);
    }
}
```



```
F:\exam>javac UserGenericFunction.java
F:\exam>java UserGenericFunction
java.lang.Integer = 22
java.lang.String = kiran
java.lang.Integer = 96
F:\exam>
```

## 9. Write a Java program to show working of user defined Generic functions using Array.

```
import java.util.*;

class ArrayGenerics

{
    public static void main(String args[])
    {
        ArrayList<String> list=new ArrayList<String>();
        list.add("kiran");
        list.add("padole");
```

```
String s=list.get(1);

System.out.println("element is: "+s);

Iterator<String> itr=list.iterator();

while(itr.hasNext())

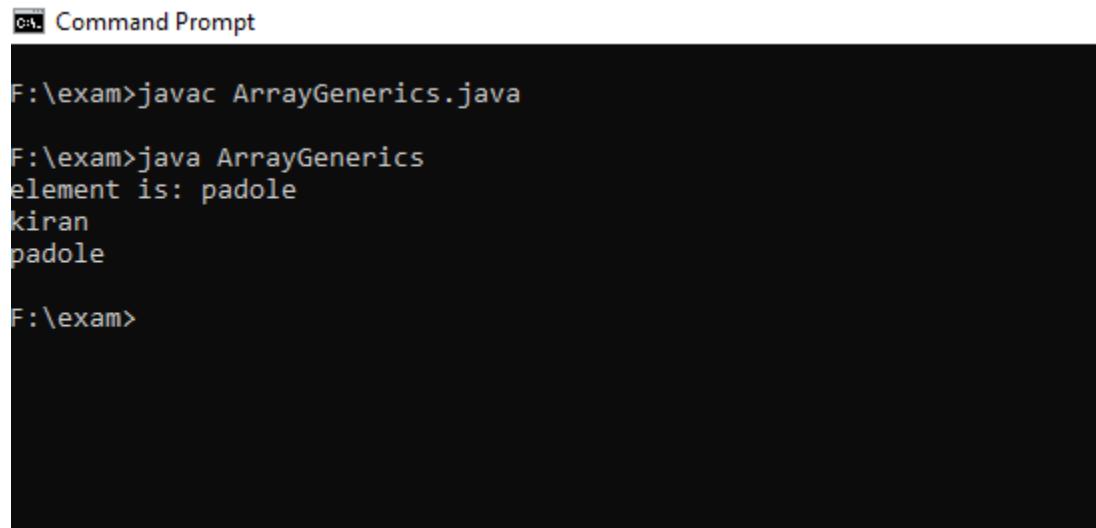
{

    System.out.println(itr.next());

}

}

}
```



The screenshot shows a Command Prompt window titled "Command Prompt". The user has navigated to the directory "F:\exam" and run the command "javac ArrayGenerics.java". After compilation, they run "java ArrayGenerics" and the program outputs three names: "padole", "kiran", and "padole". The prompt then returns to "F:\exam>".

```
F:\exam>javac ArrayGenerics.java
F:\exam>java ArrayGenerics
element is: padole
kiran
padole
F:\exam>
```

## 10. Write java program for email validation using regex.

```
import java.util.Scanner;

import java.util.regex.Matcher;
```

```
import java.util.regex.Pattern;

public class RegEmail

{

    public static void main(String[] args)

    {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Your Email: ");

        String phone = sc.next();

        String regex = "^[a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+$";

        Pattern pattern = Pattern.compile(regex);

        Matcher matcher = pattern.matcher(phone);

        if(matcher.matches())

        {

            System.out.println("Email Valid..!!");

        }

        else

        {

            System.out.println("Invalid Email");

        }

    }

}
```

```
c:\ Command Prompt
F:\exam>javac RegEmail.java
F:\exam>java RegEmail
Enter Your Email: padolekiran5@gmail.com
Email Valid..!!
F:\exam>
}
```

## **11. Write java program for mobile number validation using regex.**

```
import java.util.Scanner;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class RegEmail

{

    public static void main(String[] args)

    {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Your Email: ");

        String phone = sc.next();
```

```
String regex = "^[a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+$";  
Pattern pattern = Pattern.compile(regex);  
Matcher matcher = pattern.matcher(phone);  
if(matcher.matches())  
{  
    System.out.println("Email Valid..!!");  
}  
else  
{  
  
    System.out.println("Invalid Email");  
}  
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command line shows the path "F:\exam>" followed by the command "javac PhoneNumberEx.java". The output of the program is displayed below the command line, showing the user input "Enter your name number" followed by "Kiran padole", "Enter your phone number" followed by "9865898523", and the resulting output "Given phone number is valid".

```
F:\exam>javac PhoneNumberEx.java  
F:\exam>java PhoneNumberEx  
Enter your name number  
Kiran padole  
Enter your phone number  
9865898523  
Given phone number is valid  
F:\exam>
```

**12. Write java program to accept a string and pattern from user using regex :**

**a. display the index of occurrences of pattern in the String.**

```
import java.util.regex.*;
class StringPattern
{
    public static void main(String args[])
    {
        String content = "My Name is " +"KIRAN.";
        String pattern = "My.*";
        boolean isMatch = Pattern.matches(pattern, content);
        System.out.println("The text contains 'My'== " + isMatch);
    }
}
```

Command Prompt

```
F:\exam>javac StringPattern.java
```

```
F:\exam>java StringPattern
The text contains 'My'== true
```

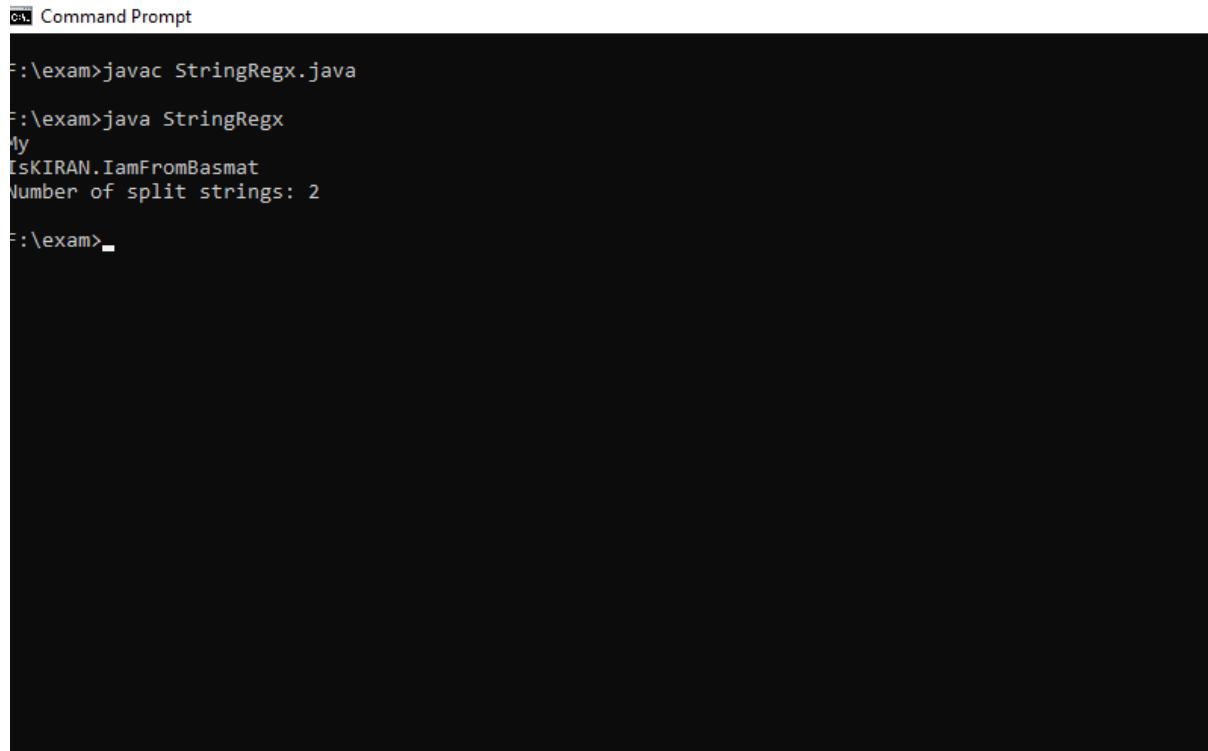
```
F:\exam>
```

```
}
```

**b. number of times pattern is occurring is String.**

```
import java.util.regex.*;
class StringRegx
{
    public static void main(String args[])
    {
        String text = "MyNameIsKIRAN.IamFromBasmat";
        String patternString = "Name";
        Pattern pattern = Pattern.compile(patternString,
        Pattern.CASE_INSENSITIVE);
        String[] myStrings = pattern.split(text);
```

```
for(String temp: myStrings)  
{  
    System.out.println(temp);  
}  
  
System.out.println("Number of split strings: "+myStrings.length);  
}  
}
```



```
F:\exam>javac StringRegx.java  
F:\exam>java StringRegx  
My  
IsKIRAN.IamFromBasmat  
Number of split strings: 2  
F:\exam>
```

**13. Write java program to accept user name and password store it in file. Accept password if following conditions holds true using regex:**

**a. Minimum 8 characters should be taken**

- b. It must have atleast 1 digit**
- c. It must have atleast 1 special Character**
- d. It must have atleast 1 Capital case letter.**

```
import java.util.regex.*;  
import java.io.FileOutputStream;  
import java.util.*;  
class passwordValiadtion  
{  
    public static void main(String args[])  
    {  
        Scanner sc=new Scanner(System.in);  
        try{  
  
            FileOutputStream fout=new  
FileOutputStream("f:/exam/kiran padole/userid_txt");  
            FileOutputStream fout1=new  
FileOutputStream("f:/exam/kiran padole/pass.txt");  
            String username;  
            String pass;
```

```
System.out.print("Enter Your User_Name :");

username=sc.next();

System.out.print("Enter Your Password :");

pass=sc.next();

String regularEx="^(?=.*[0-9])(?=.*[A-Z])(?=.*[@#$%^&-+=()])(?=.*\\S+$.{8,$}";

boolean ans = pass.matches(regularEx);

if(ans)

{

    try{

        System.out.println("Your Password is Correct");

        byte b1[]=username.getBytes();

        byte b[]=pass.getBytes();

        fout.write(b1);

        fout1.write(b);

        fout.close();

        System.out.println("success...Usename And Password is stored in

your file");

    }

    catch(Exception e){

        e.printStackTrace();

    }

}
```

```

else

{

System.out.println("Error...Please Check Your Password Once Again");

}

}

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

}
}

```

Command Prompt

```

F:\exam>javac passwordValidation.java

F:\exam>java passwordValidation
Enter Your User_Name :KIRS
Enter Your Password :Kiran1234
Error...Please Check Your Password Once Again

F:\exam>java passwordValidation
Enter Your User_Name :KRAM
Enter Your Password :kIRAN@123
Your Password is Correct
success...Username And Password is stored in your file

F:\exam>_

```

**14. Write java program to split a string by whitespace and display the splitted string [use regex].**

```
class SplitExample{

    public static void main(String args[]){

        String str = new String("28/12/2013");

        System.out.println("split(String regex):");

        String array1[]= str.split("/");

        for (String temp: array1){

            System.out.println(temp);

        }

        System.out.println("split(String regex, int limit) with limit=2:");

        String array2[]= str.split("/", 2);

        for (String temp: array2){

            System.out.println(temp);

        }

        System.out.println("split(String regex, int limit) with limit=0:");

        String array3[]= str.split("/", 0);
```

```
for (String temp: array3){

    System.out.println(temp);

}

System.out.println("split(String regex, int limit) with limit=-5:");

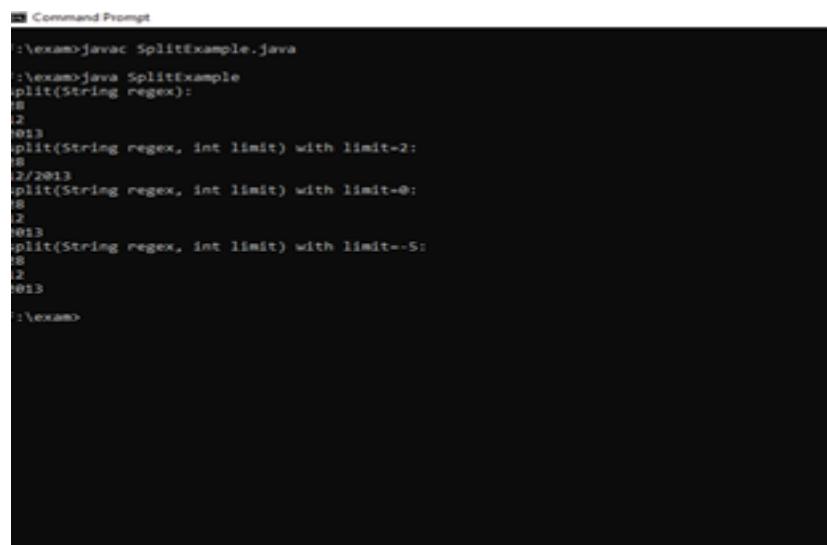
String array4[]= str.split("/", -5);

for (String temp: array4){

    System.out.println(temp);

}

}
```



The screenshot shows a Command Prompt window with the following text:

```
Command Prompt
:C:\exam>javac SplitExample.java
:C:\exam>java SplitExample
split(String regex):
5
2
@13
split(String regex, int limit) with limit=2:
5
2
@13
split(String regex, int limit) with limit=@:
5
2
@13
split(String regex, int limit) with limit=-5:
5
2
@13

:C:\exam>
```

## **15. Write a program to sort elements in an ArrayList.**

```
import java.util.*;  
  
public class SortArrayList  
{  
  
    public static void main(String args[])  
    {  
  
        ArrayList<String> list = new ArrayList<String>();  
  
        list.add("India");  
  
        list.add("United-States-America");  
  
        list.add("Brazil");  
  
        list.add("Dubai");  
  
        list.add("Germany");  
  
        list.add("Belgium");  
  
        list.add("France");  
  
        list.add("Nepal");  
  
        System.out.println("Before Sorting: "+ list);  
  
        Collections.sort(list);  
  
        System.out.println("After Sorting: "+ list);  
  
    }  
}
```

```
ca Command Prompt
F:\exam>javac SortArrayList.java
F:\exam>java SortArrayList
Before Sorting: [India, United-States-America, Brazil, Dubai, Germany, Belgium, France, Nepal]
After Sorting: [Belgium, Brazil, Dubai, France, Germany, India, Nepal, United-States-America]
F:\exam>
}
```

## 16. Write a Java program to compare two array lists.

```
import java.util.ArrayList;

public class arrayList

{
    public static void main(String[] args)

    {
        ArrayList<String> ArrayList1 = new ArrayList<String>();
        ArrayList<String> ArrayList2 = new ArrayList<String>();

        ArrayList1.add("item 1");
        ArrayList1.add("item 2");
        ArrayList1.add("item 3");
        ArrayList1.add("item 4");
```

```
ArrayList2.add("item 1");

ArrayList2.add("item 2");

ArrayList2.add("item 3");

ArrayList2.add("item 4");

System.out.println(" ArrayList1 = " + ArrayList2);

System.out.println(" ArrayList1 = " + ArrayList1);

if (ArrayList1.equals(ArrayList2) == true)

{

    System.out.println(" Array List are equal");

}

else

{

    System.out.println(" Array List are not equal");

}

System.out.println("\n Lets insert one more item in Array List 1");

ArrayList1.add("item 5");

System.out.println(" ArrayList1 = " + ArrayList1);

System.out.println(" ArrayList = " + ArrayList2);

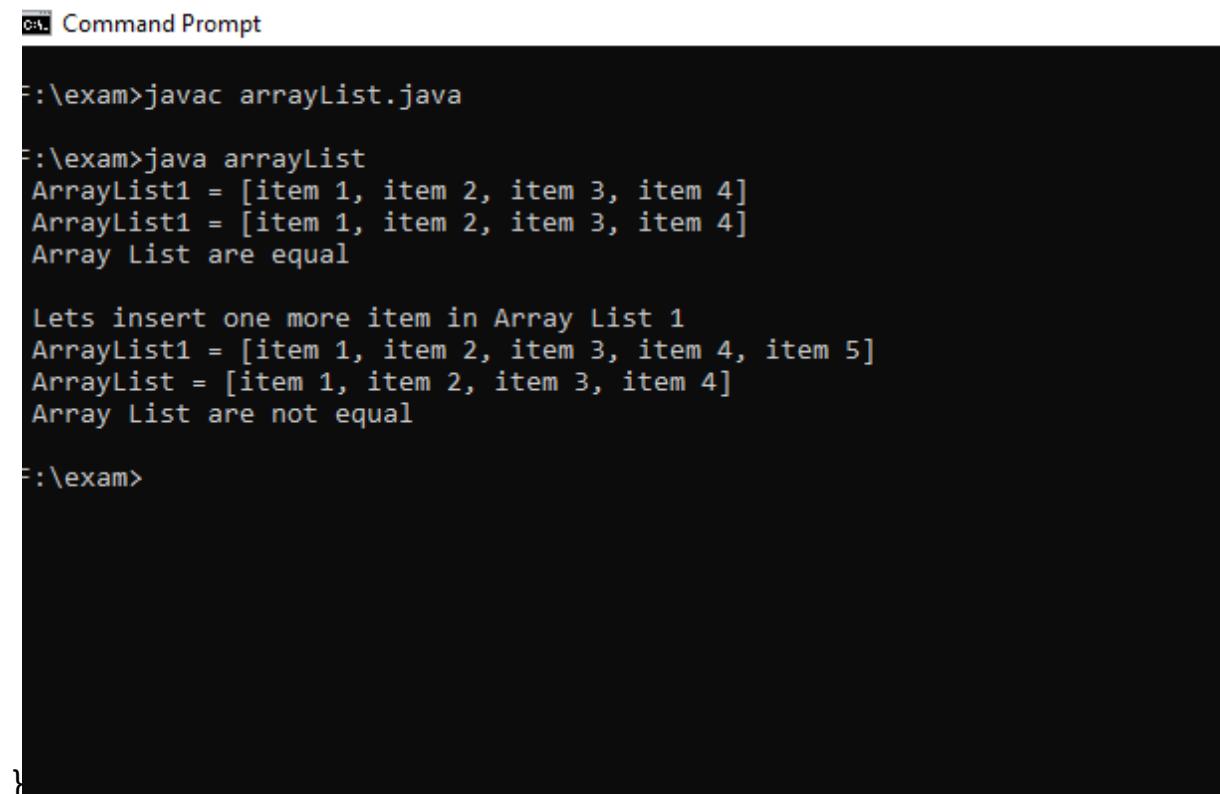
if (ArrayList1.equals(ArrayList2) == true)

{

    System.out.println(" Array List are equal");

}
```

```
else  
{  
    System.out.println(" Array List are not equal");  
}  
}  
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command line shows the execution of a Java program named "arrayList.java". The output displays two ArrayLists being compared, showing they are initially equal. Then, one item is inserted into the first ArrayList, making them unequal. Finally, the command prompt returns to the user.

```
c:\ Command Prompt  
F:\exam>javac arrayList.java  
F:\exam>java arrayList  
ArrayList1 = [item 1, item 2, item 3, item 4]  
ArrayList2 = [item 1, item 2, item 3, item 4]  
Array List are equal  
  
Lets insert one more item in Array List 1  
ArrayList1 = [item 1, item 2, item 3, item 4, item 5]  
ArrayList2 = [item 1, item 2, item 3, item 4]  
Array List are not equal  
F:\exam>
```

## 17. Write a Java program to get the first and last occurrence of the specified elements in a linked list.

```
import java.util.LinkedList;  
  
import java.util.Iterator;
```

```
public class Linkedlist
{
    public static void main(String[] args)
    {
        LinkedList<String> l_list = new LinkedList<String>();
        l_list.add("India");
        l_list.add("USA");
        l_list.add("UAE");
        l_list.add("Belgium");
        l_list.add("Brazil");
        System.out.println("Original linked list:" + l_list);
        Object first_element = l_list.getFirst();
        System.out.println("First Element is: "+first_element);
        Object last_element = l_list.getLast();
        System.out.println("Last Element is: "+last_element);
    }
}
```

Command Prompt

```
F:\exam>javac Linkedlist.java  
F:\exam>java Linkedlist  
Original linked list:[India, USA, UAE, Belgium, Brazil]  
First Element is: India  
Last Element is: Brazil  
F:\exam>
```

## 18. Write a Java program to compare two sets and retain elements which are same on both sets.

```
import java.util.*;  
  
public class Set  
{  
    public static void main(String[] args)  
    {  
        HashSet<String> h_set1 = new HashSet<String>();  
        h_set1.add("India");  
        h_set1.add("USA");  
        h_set1.add("UAE");  
        h_set1.add("Belgium");
```

```
System.out.println("First HashSet content: "+h_set1);

HashSet<String>h_set2 = new HashSet<String>();

h_set2.add("USA");

h_set2.add("India");

h_set2.add("Germany");

h_set2.add("Austria");

System.out.println("Second HashSet content: "+h_set2);

h_set1.retainAll(h_set2);

System.out.println("HashSet content:");

System.out.println(h_set1);

}
```

```
C:\ Command Prompt
F:\exam>javac Set.java
F:\exam>java Set
First HashSet content: [USA, Belgium, UAE, India]
Second HashSet content: [USA, Austria, Germany, India]
HashSet content:
[USA, India]
F:\exam>
```

**19. Write a program to print occurrence every word in a sentence using HashMap.**

```
import java.io.*;
import java.util.*;
class OccurenceOfChar
{
    static void characterCount(String inputString)
    {
        HashMap<Character, Integer> charCountMap = new HashMap<Character,
        Integer>();
        char[] strArray = inputString.toCharArray();
        for(char c : strArray)
        {
            if(charCountMap.containsKey(c))
            {
                charCountMap.put(c, charCountMap.get(c) + 1);
            }
            else
            {
                charCountMap.put(c, 1);
            }
        }
        for(Map.Entry entry : charCountMap.entrySet())
        {

```

```

        System.out.println(entry.getKey() + " " + entry.getValue());
    }
}

public static void main(String[] args)
{
    Scanner s=new Scanner(System.in);
    System.out.println("Enter a string:");
    String str = s.nextLine();
    characterCount(str);
}

```

c:\ Command Prompt

```

F:\exam>javac OccurenceOfChar.java

F:\exam>java OccurenceOfChar
Enter a string:
kiran
a 1
r 1
i 1
k 1
n 1

```

F:\exam>

**20. Write a Java program to create a reverse order view of the elements contained in a given tree set.**

```
import java.util.TreeSet;
import java.util.Iterator;
public class Treeset
{
    public static void main(String[] args)
    {
        TreeSet<String> t_set = new TreeSet<String>();
        t_set.add("Red");
        t_set.add("Green");
        t_set.add("Black");
        t_set.add("Pink");
        t_set.add("orange");
        System.out.println("Original tree set:" + t_set);
        Iterator it = t_set.descendingIterator();
        System.out.println("Elements in Reverse Order:");
        while(it.hasNext())
        {
            System.out.println(it.next());
        }
    }
}
}import java.util.TreeSet;
```

```
import java.util.Iterator;

public class Treeset

{

    public static void main(String[] args)

    {

        TreeSet<String> t_set = new TreeSet<String>();

        t_set.add("Red");

        t_set.add("Green");

        t_set.add("Black");

        t_set.add("Pink");

        t_set.add("orange");

        System.out.println("Original tree set:" + t_set);

        Iterator it = t_set.descendingIterator();

        System.out.println("Elements in Reverse Order:");

        while(it.hasNext())

        {

            System.out.println(it.next());

        }

    }

}
```

Command Prompt

```
F:\exam>javac Treeset.java

F:\exam>java Treeset
Original tree set:[Black, Green, Pink, Red, orange]
Elements in Reverse Order:
orange
Red
Pink
Green
Black

F:\exam>
```

## Assignment No. 5

1.Create one registration form using Frame having fields username, password, confirm password mobile number, address state, gender on click event all selected data should appear in label.

```
import java.awt.*;
import java.awt.event.*;
public class RegistrationForm extends Frame implements ActionListener
{
    Label l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16,l17,l18;
    TextField t1,t2,t3,t4,t5,t6;
    Button b1,b2;
    RegistrationForm()
    {
        setTitle("REGISTRATION-FORM");
        l1=new Label("USERNAME");
        l2=new Label("PASSWORD");
        l3=new Label("MOBILE NUMBER");
        l4=new Label("ADDRESS");
        l5=new Label("STATE");
        l6=new Label("GENDER");
        l7=new Label("USERNAME");
        l8=new Label("PASSWORD");
        l9=new Label("MOBILE NUMBER");
        l10=new Label("ADDRESS");
```

```
I11=new Label("STATE");
I12=new Label("GENDER");
I13=new Label(" ");
I14=new Label(" ");
I15=new Label(" ");
I16=new Label(" ");
I17=new Label(" ");
I18=new Label(" ");

t1=new TextField();
t2=new TextField();
t3=new TextField();
t4=new TextField();
t5=new TextField();
t6=new TextField();

b1=new Button("SUBMIT");
b2=new Button("EXIT");

add(I1);
add(I2);
add(I3);
add(I4);
add(I5);
add(I6);
add(I7);
add(I8);
add(I9);
```

```
add(l10);
add(l11);
add(l12);
add(l13);
add(l14);
add(l15);
add(l16);
add(l17);
add(l18);
add(t1);
add(t2);
add(t3);
add(t4);
add(t5);
add(t6);
add(b1);
add(b2);
l1.setBounds(50,50,80,30);
l2.setBounds(50,100,80,30);
l3.setBounds(50,150,80,30);
l4.setBounds(50,200,80,30);
l5.setBounds(50,250,80,30);
l6.setBounds(50,300,80,30);
t1.setBounds(180,50,80,30);
t2.setBounds(180,100,80,30);
```

```
t3.setBounds(180,150,80,30);

t4.setBounds(180,200,80,30);

t5.setBounds(180,250,80,30);

t6.setBounds(180,300,80,30);

b1.setBounds(180,350,80,30);

b2.setBounds(300,350,80,30);

l7.setBounds(50,400,80,30);

l8.setBounds(50,450,80,30);

l9.setBounds(50,500,80,30);

l10.setBounds(50,550,80,30);

l11.setBounds(50,600,80,30);

l12.setBounds(50,650,80,30);

l13.setBounds(180,400,80,30);

l14.setBounds(180,450,80,30);

l15.setBounds(180,500,80,30);

l16.setBounds(180,550,80,30);

l17.setBounds(180,600,80,30);

l18.setBounds(180,650,80,30);

b1.addActionListener(this);

b2.addActionListener(this);

setLayout(null);

setSize(800,800);

setVisible(true);

}

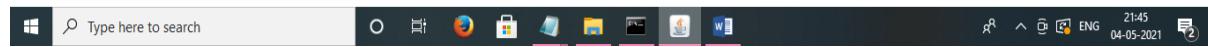
public void actionPerformed(ActionEvent e)
```

```
{  
if(e.getSource()==b1)  
{  
l13.setText(t1.getText());  
l14.setText(t2.getText());  
l15.setText(t3.getText());  
l16.setText(t4.getText());  
l17.setText(t5.getText());  
l18.setText(t6.getText());  
}  
}  
  
public static void main(String args[]){  
}  
RegistrationForm f=new RegistrationForm();  
}  
}
```

REGISTRATION-FORM

USERNAME	kiran
PASSWORD	pass
MOBILE NUMBER	356848965
ADDRESS	sdfdgdd
STATE	pune
GENDER	female
SUBMIT	
EXIT	

USERNAME kiran  
PASSWORD pass  
MOBILE NUMBER 356848965  
ADDRESS sdfdgdd  
STATE pune  
GENDER female



2. Write a java program to create the GUI for Customer Registration form. Perform the following validation.

- I. Password should be minimum 6 character containing at least one Uppercaseletter, one digit and one Symbol.
- ii. Confirm password and password fields should be match.
- iii. The captcha should generate two random 2 digit numbers accept the sum from the user. If above conditions are met, display “Registration successful” otherwise “Registration Failed” after the user clicks the submit button.

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.swing.*;
public class Register extends JFrame
{
    JFrame f;
    Register()
    {
        setTitle("Customer Registration Form");
        JLabel name, pass, pass1, captcha, num1, num2, yes, no;
        JTextField tname, lname, tcaptcha;
        JPasswordField tpass, tpass1;
        JButton sub;
        int r1,r2,sum;
        name = new JLabel("Name: ");
        name.setBounds(50, 50, 250, 30);
        add(name);
        tname = new JTextField();
        tname.setBounds(50, 80, 250, 30);
        add(tname);
```

```
name = new JLabel("Login Name: ");
name.setBounds(50, 110, 250, 30);
add(name);
```

```
Iname = new JTextField();
Iname.setBounds(50, 140, 250, 30);
add(Iname);
```

```
pass = new JLabel(" Password:");
pass.setBounds(50, 170, 250, 30);
add(pass);
tpass = new JPasswordField();
tpass.setBounds(50, 200, 250, 30);
add(tpass);
```

```
pass1 = new JLabel("Confirm Password: ");
pass1.setBounds(50, 230, 250, 30);
add(pass1);
tpass1 = new JPasswordField();
tpass1.setBounds(50, 260, 250, 30);
add(tpass1);
```

```
Random r = new Random();
```

```
r1 = r.nextInt(50);

r2 = r.nextInt(50);

sum = r1+r2;

System.out.println(sum);

captcha = new JLabel("captcha");

captcha.setBounds(50,290,500,30);

num1 = new JLabel(""+r1+"");

num1.setBounds(50, 320, 20, 30);

num2 = new JLabel(" "+ "+r2+"");

num2.setBounds(60, 350, 50, 30);

tcaptcha = new JTextField();

tcaptcha.setBounds(50, 380, 100, 30);

sub = new JButton("Submit");

sub.setBounds(50, 410, 250, 30);

sub.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent e) {

        char[] p0 = tpass.getPassword();
```

```

String p1 = new String(p0);

System.out.println(p1);

String regex = "^(?=.*[0-9])"
    + "(?=.*[a-z])(?=.*[A-Z])"
    + "(?=.*[@#$%^&+=])"
    + ".{6,32}$";

Pattern p = Pattern.compile(regex);

Matcher m = p.matcher(p1);

boolean b = m.matches();

String p2 = String.valueOf(tpass1.getPassword());

System.out.println(p1);

System.out.println(b);

if(b)

{

    if(p1.equals(p2))

    {

        int c = Integer.parseInt(tcaptcha.getText());

        if(c==sum)

        {

            System.out.println("Correct");

            JOptionPane.showMessageDialog(f, "Registered

Successfully");

```

```
        }

        else

        {

            System.out.println("Incorrect");

            JOptionPane.showMessageDialog(f, "Captcha

Incorrect");

        }

    }

    else

    {

        System.out.println("Enter same values in both

passwords!!");

        JOptionPane.showMessageDialog(f, "Enter same values

in password and confirm password!!!");

    }

}

else

{

    System.out.println("Password Invalid");

    JOptionPane.showMessageDialog(f, "Invalid Password");

}

}

});
```

```
        add(sub);

        add(captcha);

        add(num1);

        add(num2);

        add(tcaptcha);

        setSize(400, 500);

        setLayout(null);

        setVisible(true);

        setDefaultCloseOperation(EXIT_ON_CLOSE);

    }

}

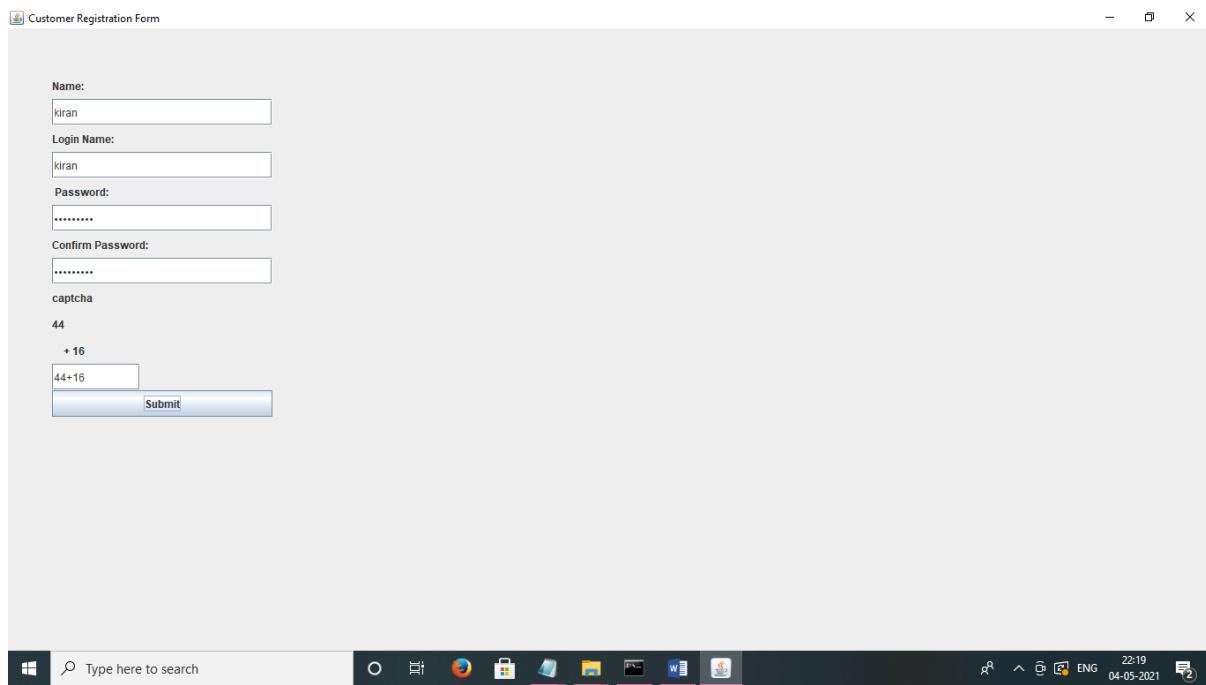
public static void main(String[] args)

{

    new Register();

}

}
```



3. Write a java program to implement a simple arithmetic calculator.  
Perform appropriate validation.

```
package com.dyp.javaassignment1;

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

public class SimpleCalculator extends JFrame implements ActionListener
{
    JButton b10,b11,b12,b13,b14,b15;
```

```
 JButton b[]=new JButton[10];  
  
int i,r,n1,n2;  
  
JTextField res;  
  
char op;  
  
public SimpleCalculator()  
{  
super("Calulator");  
  
setLayout(new BorderLayout());  
  
JPanel p=new JPanel();  
  
p.setLayout(new GridLayout(4,4));  
  
for(int i=0;i<=9;i++)  
{  
b[i]=new JButton(i+"");  
  
p.add(b[i]);  
  
b[i].addActionListener(this);  
}  
  
b10=new JButton("+");  
  
p.add(b10);  
  
b10.addActionListener(this);  
  
  
b11=new JButton("-");  
  
p.add(b11);  
  
b11.addActionListener(this);  
  
  
b12=new JButton("*");
```

```
p.add(b12);
b12.addActionListener(this);

b13=new JButton("/");
p.add(b13);
b13.addActionListener(this);

b14=new JButton("=");
p.add(b14);
b14.addActionListener(this);

b15=new JButton("C");
p.add(b15);
b15.addActionListener(this);

res=new JTextField(20);
add(p, BorderLayout.CENTER);
add(res, BorderLayout.NORTH);
setVisible(true);
setSize(800,800);
}

public void actionPerformed(ActionEvent ae)
{
    JButton pb=(JButton)ae.getSource();
    if(pb==b15)
```

```

{

r=n1=n2=0;

res.setText("");


}

else

if(pb==b14)

{

n2=Integer.parseInt(res.getText());

eval();

res.setText(""+r);

}

else

{

boolean opf=false;

if(pb==b10)

{ op='+';

opf=true;

}

if(pb==b11)

{ op='-';opf=true; }

if(pb==b12)

{ op='*';opf=true; }

if(pb==b13)

{ op='/';opf=true; }
}

```

```
if(opf==false)

{
    for(i=0;i<10;i++)

    {
        if(pb==b[i])

        {

String t=res.getText();

t+=i;

res.setText(t);

    }

}

}

else

{

n1=Integer.parseInt(res.getText());

res.setText("");

}

}

}

int eval()

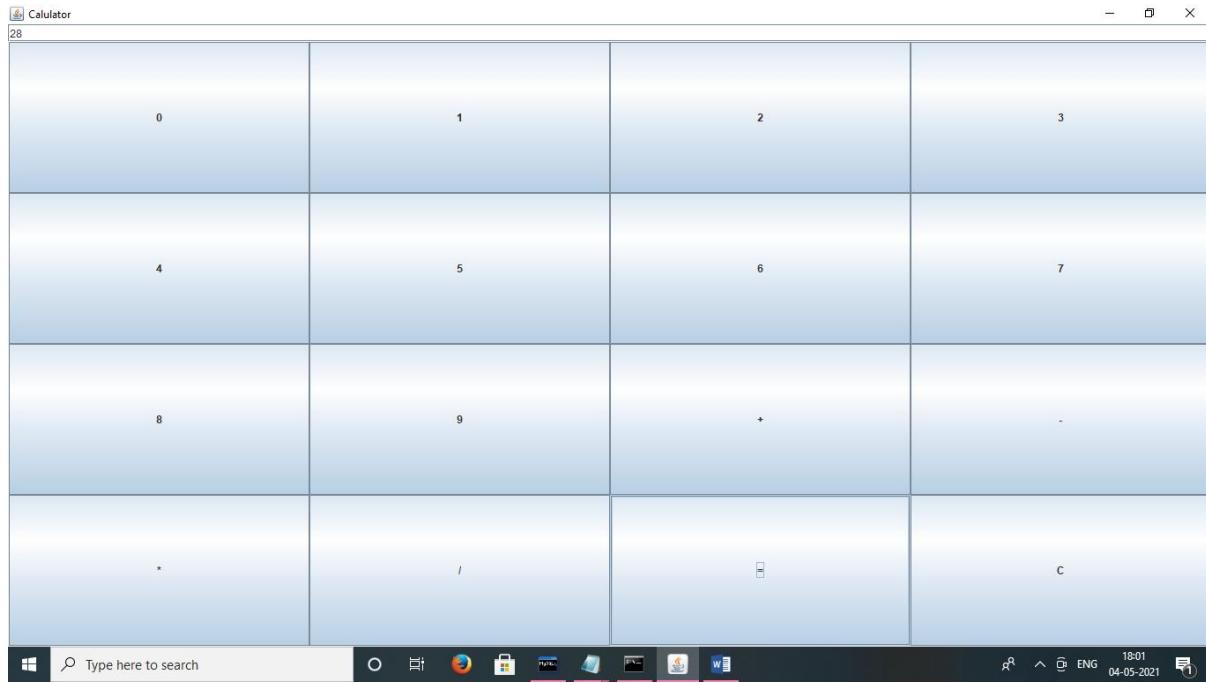
{

switch(op)

{

case '+':
```

```
r=n1+n2;  
break;  
  
case '-':  
r=n1-n2;  
break;  
  
case '*':  
r=n1*n2;  
break;  
  
case '/':  
r=n1/n2;  
break;  
}  
  
return 0;  
}  
  
public static void main(String arg[])  
{  
    new SimpleCalculator();  
}  
}
```



4. Write a program to create two lists and transfer elements from one list to another. Multiple selection are allowed. The add button allows an element to be added and the Remove button allows an element to be removed (Accepted in an input dialog). Do not add duplicate elements.

5. Write a java program having 3 scrollbar for colors Red, blue, green having range from 0-255 perform event handing for displaying color on text area as the scroll bar is moved.

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

public class ScrollColor implements AdjustmentListener {

    private static void createAndShowGUI() {

        JFrame frame = new JFrame("JScrollBar");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setBounds(20,30,200,250);
        frame.getContentPane().setLayout(null);

        ScrollColor app = new ScrollColor();
        app.sbar1 = new JScrollBar(java.awt.Adjustable.VERTICAL, 127,
        1,0,255);
        app.sbar1.setBounds(10,20, 10, 200);
        app.sbar1.setBackground(Color.red);
        app.sbar1.addAdjustmentListener(app);
        frame.getContentPane().add(app.sbar1);

        app.sbar2 = new JScrollBar(java.awt.Adjustable.VERTICAL, 127,
        1,0,255);
        app.sbar2.setBounds(30,20, 10, 200);
```

```
    app.sbar2.setBackground(Color.green);

    app.sbar2.addAdjustmentListener(app);

    frame.getContentPane().add(app.sbar2);

    app.sbar3 = new JScrollBar(java.awt.Adjustable.VERTICAL, 127,
1,0,255);

    app.sbar3.setBounds(50,20, 10, 200);

    app.sbar3.setBackground(Color.blue);

    app.sbar3.addAdjustmentListener(app);

    frame.getContentPane().add(app.sbar3);

    app.panel = new JPanel();

    app.panel.setBounds(80,20,50,200);

    app.panel.setBackground(new Color(0,0,0));

    frame.getContentPane().add(app.panel);

    frame.setVisible(true);

}

public void adjustmentValueChanged(AdjustmentEvent e)

{

    panel.setBackground(new Color(sbar1.getValue(),sbar2.getValue(),
sbar3.getValue()));

}
```

```
}
```

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

```
    SwingUtilities.invokeLater(new Runnable() {
```

```
        public void run() {
```

```
            createAndShowGUI();
```

```
        }
```

```
    });
```

```
}
```

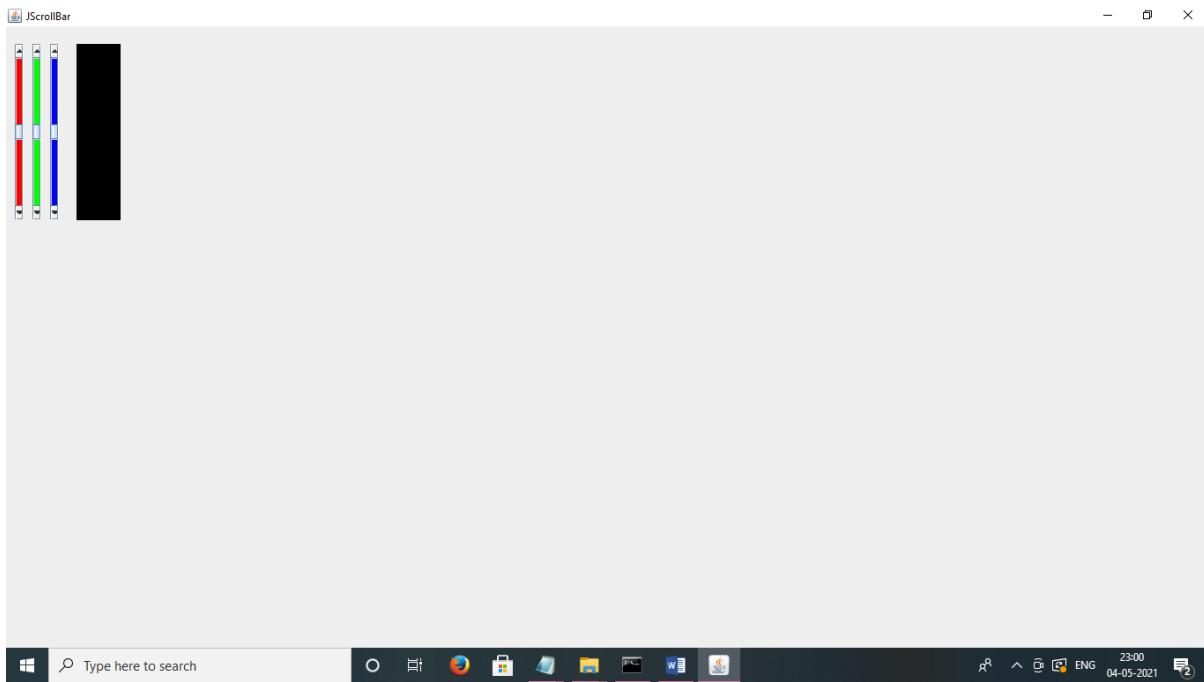
```
JScrollBar sbar1;
```

```
JScrollBar sbar2;
```

```
JScrollBar sbar3;
```

```
JPanel panel;
```

```
}
```



6. Write a java program to accept host address from user and print it's IP address in Label on click of button. Use JColorChooser for changing the background color.

```
import javax.swing.*;  
  
import java.awt.*;  
  
import java.awt.event.*;  
  
public class LabelExample1 extends Frame implements ActionListener{  
  
    JTextField tf; JLabel l; JButton b;  
  
    LabelExample1(){  
  
        tf=new JTextField();  
  
        tf.setBounds(50,50, 150,20);  
  
        l=new JLabel();
```

```
l.setBounds(50,100, 250,20);

b=new JButton("Find IP");

b.setBounds(50,150,95,30);

b.addActionListener(this);

add(b);add(tf);add(l);

setSize(400,400);

setLayout(null);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

try{

String host=tf.getText();

String ip=java.net.InetAddress.getByName(host).getHostAddress();

l.setText("IP of "+host+" is: "+ip);

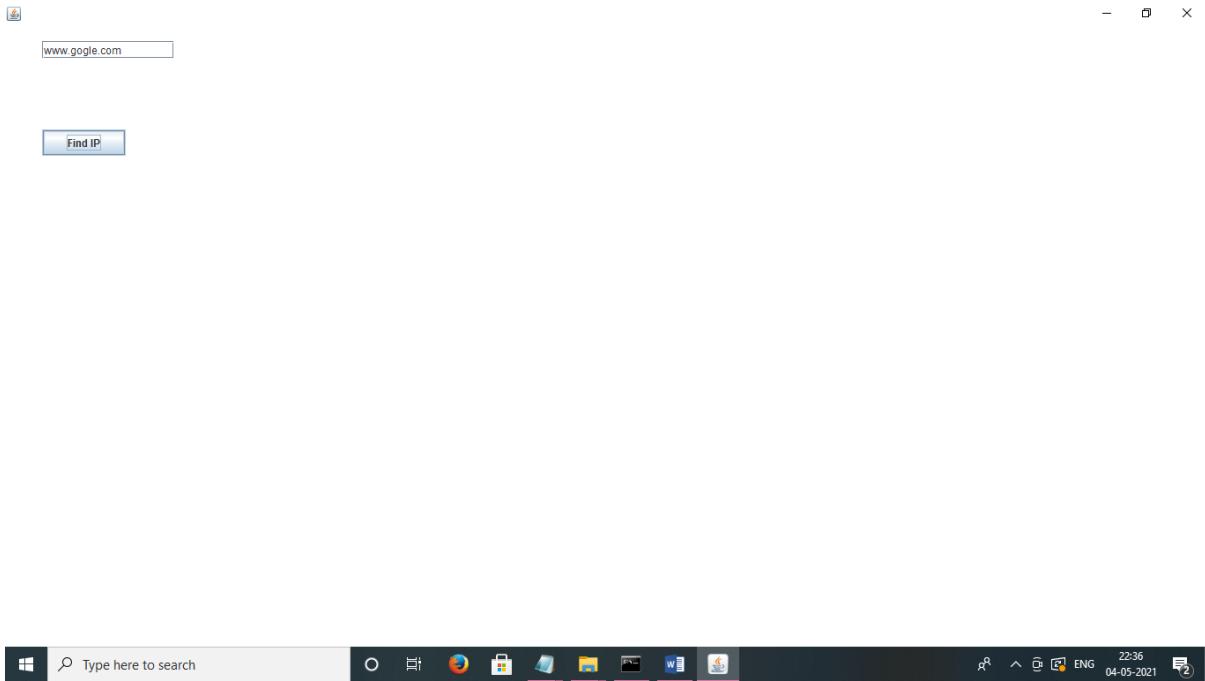
}catch(Exception ex){System.out.println(ex);}

}

public static void main(String[] args) {

new LabelExample1();

} }
```



7. Write a program for design a student feedback from having various swing component like Label, JButtonField, JTextarea, JComboBox, JRadioButton, ImageIcon, JOptionPane, etc. and use setToolTipText() also, and on click submit button contents should be displayed.

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

```
class FeedbackPage extends JFrame implements ActionListener {
```

```
Container c;  
  
JLabel title;  
  
JLabel name;  
  
JTextField tname;  
  
JLabel mno;  
  
JTextField tmno;  
  
JLabel gender;  
  
JRadioButton male;  
  
JRadioButton female;  
  
ButtonGroup gengp;  
  
JLabel add;  
  
JTextArea tfed;  
  
JCheckBox term;  
  
JButton sub;  
  
JButton reset;  
  
JTextArea tout;  
  
JLabel res;
```

```
public FeedbackPage()  
{
```

```
setTitle("Feedback Form");

setBounds(300, 90, 900, 600);

setDefaultCloseOperation(EXIT_ON_CLOSE);

c = getContentPane();

c.setLayout(null);

title = new JLabel("Feedback Form");

title.setSize(300, 30);

title.setLocation(300, 30);

c.add(title);

name = new JLabel("Name");

name.setSize(100, 20);

name.setLocation(100, 100);

c.add(name);

tname = new JTextField();

tname.setSize(190, 20);

tname.setLocation(200, 100);
```

```
c.add(tname);

mno = new JLabel("Mobile");
mno.setSize(100, 20);
mno.setLocation(100, 150);
c.add(mno);

tmno = new JTextField();
tmno.setSize(150, 20);
tmno.setLocation(200, 150);
c.add(tmno);

gender = new JLabel("Gender");
gender.setSize(100, 20);
gender.setLocation(100, 200);
c.add(gender);

male = new JRadioButton("Male");
male.setSelected(true);
male.setSize(75, 20);
male.setLocation(200, 200);
```

```
c.add(male);

female = new JRadioButton("Female");
female.setSelected(false);
female.setSize(80, 20);
female.setLocation(275, 200);
c.add(female);
```

```
gengp = new ButtonGroup();
gengp.add(male);
gengp.add(female);
```

```
add = new JLabel("Write Feedback");
```

```
add.setSize(100, 20);
add.setLocation(100, 300);
c.add(add);
```

```
tfed = new JTextArea();
```

```
tfed.setSize(200, 75);
tfed.setLocation(200, 300);
tfed.setLineWrap(true);
c.add(tfed);

term = new JCheckBox("Accept to submit feedback");
term.setSize(250, 20);
term.setLocation(150, 400);
c.add(term);

sub = new JButton("Submit");
sub.setSize(100, 20);
sub.setLocation(150, 450);
sub.addActionListener(this);
c.add(sub);

reset = new JButton("Reset");
reset.setSize(100, 20);
reset.setLocation(270, 450);
reset.addActionListener(this);
```

```
c.add(reset);

tout = new JTextArea();

tout.setSize(300, 400);
tout.setLocation(500, 100);
tout.setLineWrap(true);
tout.setEditable(false);
c.add(tout);

res = new JLabel("");
res.setSize(500, 25);
res.setLocation(100, 500);
c.add(res);

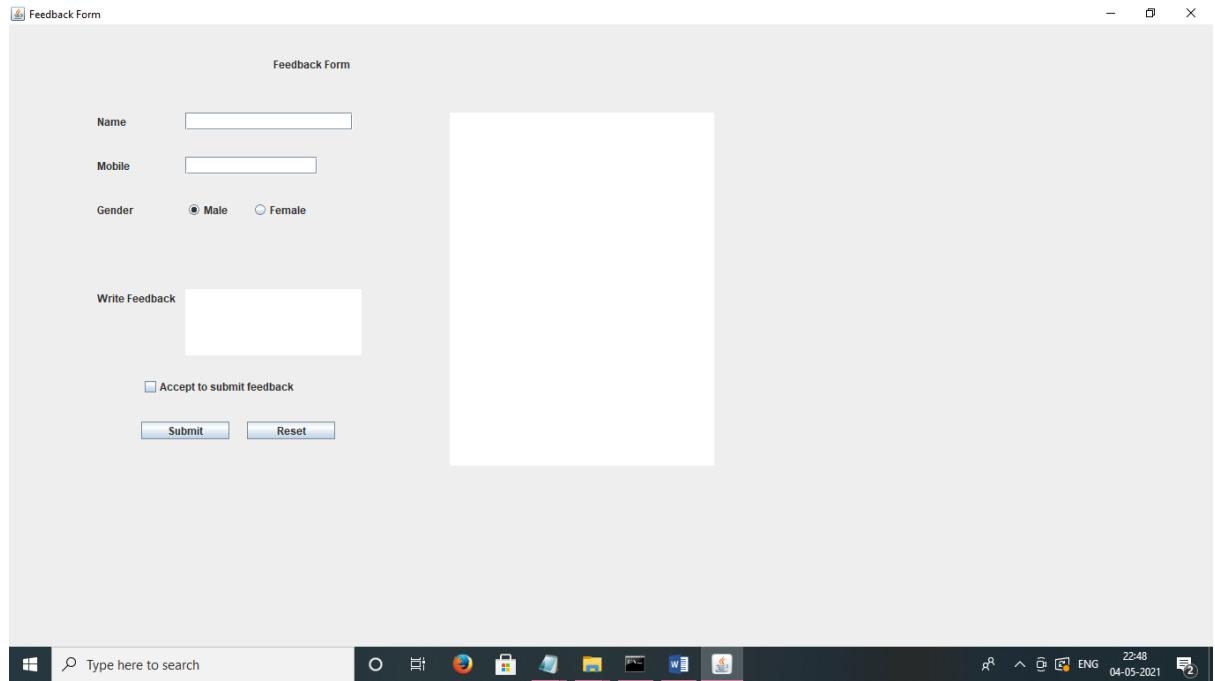
setVisible(true);

}

public void actionPerformed(ActionEvent e)
```

```
{  
    if (e.getSource() == sub) {  
        if (term.isSelected()) {  
            String data1;  
            String data  
            = "Name : "  
            + tname.getText() + "\n"  
            + "Mobile : "  
            + tmno.getText() + "\n";  
            if (male.isSelected())  
                data1 = "Gender : Male"  
                + "\n";  
            else  
                data1 = "Gender : Female"  
                + "\n";  
  
            String data3 = "Address : " + tfed.getText();  
            tout.setText(data + data1 + data3);  
            tout.setEditable(false);  
            res.setText("Successfully Submitted Feedback..");  
        }  
        else {  
    }
```

```
tout.setText("");  
  
}  
  
}  
  
else if (e.getSource() == reset) {  
  
    String def = "";  
  
    tname.setText(def);  
  
    tfed.setText(def);  
  
    tmno.setText(def);  
  
    res.setText(def);  
  
    tout.setText(def);  
  
    term.setSelected(false);  
  
}  
  
}  
  
public static void main(String[] args)  
{  
  
    FeedbackPage f = new FeedbackPage();  
  
}  
  
}
```



9. Write a swing program for designing notepad and performing relevant actions like cut, copy , select all and etc.

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

```
public class NotepadSwing extends JFrame implements  
ActionListener,ItemListener,MenuListener,KeyListener
```

```
{
```

```
FileOutputStream fos;  
  
FileInputStream fis;  
  
JFileChooser dialog;  
  
 JOptionPane msgbox;  
  
boolean modified=false;  
  
String str;  
  
JMenuBar menubar;  
  
JMenu file,edit,help;  
  
 JMenuItem newfile,savefile,openfile,exit,cut,copy,paste,about,selall;  
  
JTextArea disp;  
  
 JScrollPane scrlpane;  
  
public NotepadSwing()  
{  
  
    disp=new JTextArea();  
  
    scrlpane=new  
    JScrollPane(disp); //,ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED,ScrollPaneConstan  
ts.HORIZONTAL_SCROLLBAR_AS_NEEDED);  
  
    menubar=new JMenuBar();  
  
    dialog=new JFileChooser();  
  
    msgbox=new JOptionPane();  
  
    file=new JMenu("File"); file.setMnemonic('F');  
  
    edit=new JMenu("Edit"); edit.setMnemonic('E');  
  
    help=new JMenu("Help"); help.setMnemonic('H');  
  
    newfile=new JMenuItem("New"); newfile.setMnemonic('N');  
  
    openfile=new JMenuItem("Open"); openfile.setMnemonic('O');
```

```
savefile=new JMenuItem("Save"); savefile.setMnemonic('S');

exit=new JMenuItem("Exit"); exit.setMnemonic('x');

cut=new JMenuItem("Cut"); cut.setMnemonic('C');

copy=new JMenuItem("Copy"); copy.setMnemonic('o');

paste=new JMenuItem("Paste"); paste.setMnemonic('P');

selall=new JMenuItem("Select All"); selall.setMnemonic('A');

about=new JMenuItem("About"); about.setMnemonic('A');

cut.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X,InputEvent.CTRL_MASK,true));

copy.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C,InputEvent.CTRL_MASK,true));

paste.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_P,InputEvent.CTRL_MASK,true));

openfile.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O,InputEvent.CTRL_MASK,true));

savefile.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_S,InputEvent.CTRL_MASK,true));

newfile.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N,InputEvent.CTRL_MASK,true));

file.add(newfile);

file.add(openfile);

file.add(savefile);

file.addSeparator();

file.add(exit);

edit.add(cut);

edit.add(copy);
```

```
edit.add(paste);
edit.addSeparator();
edit.add(selall);
help.add(about);
menubar.add(file);
menubar.add(edit);
menubar.add(help);
Container c=getContentPane();
getRootPane().setJMenuBar(menubar);
c.add(scrlpane);
setSize(500,500);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setVisible(true);
setTitle("Untitled - Notepad");
savefile.addActionListener(this);
openfile.addActionListener(this);
exit.addActionListener(this);
cut.addActionListener(this);
copy.addActionListener(this);
paste.addActionListener(this);
selall.addActionListener(this);
newfile.addActionListener(this);
edit.addMenuListener(this);
disp.addKeyListener(this);
```

```
}

public void keyPressed(KeyEvent e)

{

//System.out.println("Press");

}

public void keyReleased(KeyEvent e)

{

//System.out.println("Release");

}

public void keyTyped(KeyEvent e)

{

String str=KeyEvent.getKeyModifiersText(InputEvent.CTRL_MASK);

if(!str.equals("Ctrl"))

modified=true;

}

public void menuSelected(MenuEvent ml)

{

//System.out.println("Sele");

if(ml.getSource()==edit)

{

if(disp.getSelectedText()==null)

{

cut.setEnabled(false);

copy.setEnabled(false);


```

```
    }

    else

    {

        cut.setEnabled(true);

        copy.setEnabled(true);

    }

}

public void menuDeselected(MenuEvent ml)

{

    //System.out.println("DeSele");

}

public void menuCanceled(MenuEvent ml)

{

    //System.out.println("Can");

}

public void actionPerformed(ActionEvent ae){

    if(ae.getSource()==savefile)

    {

        if(getTitle().equals("Untitled - Notepad"))

            saveFile(0);

        else if(modified)

            saveFile(1);

    }

}
```

```
else if(ae.getSource()==openfile)
{
    int x;
    if(modified)
    {
        x=isModified();
        if(x==1)
        {
            if(getTitle().equals("Untitled - Notepad"))
                saveFile(0);
        }
        else
            saveFile(1);
        openFile();
    }
    else if(x!=3) openFile();
}
else openFile();

}

else if(ae.getSource()==exit)
{
    int x;
    if(modified)
    {
        x=isModified();
```

```
if(x==1)

{
    if(getTitle().equals("Untitled - Notepad"))

        saveFile(0);

    else

        saveFile(1);

}
else if(x==2) System.exit(0);

}

else System.exit(0);

}

else if(ae.getSource()==newfile)

{
    int x;

    if(modified)

    {
        x=isModified();

        if(x==1)

        {
            if(getTitle().equals("Untitled - Notepad"))

                saveFile(0);

            else

                saveFile(1);
        }
    }
}
```

```
else if(x!=3){

    disp.setText("");
    setTitle("Untitled - Notepad");
    modified=false;
}

}

else{

    disp.setText("");
    setTitle("Untitled - Notepad");
    modified=false;
}

}

else if(ae.getSource()==cut)

{

    disp.cut();
}

else if(ae.getSource()==copy)

{

    disp.copy();
}

else if(ae.getSource()==paste)

{

    disp.paste();
}
```

```
else if(ae.getSource()==selall)

{
    disp.selectAll();
}

}

public void saveFile(int saveflag)

{
    File f;
    try
    {
        if(saveflag==0)

        {
            int x=dialog.showSaveDialog(this);
            if(x==0){

                f=dialog.getSelectedFile();
                fos=new FileOutputStream(f);
                setTitle(f.getPath());
                PrintStream ps=new PrintStream(fos);
                ps.print(disp.getText());
                modified=false;
            }
        }
    }
    else
    {

```

```
f=new File(getTitle());  
fos=new FileOutputStream(f);  
setTitle(f.getPath());  
PrintStream ps=new PrintStream(fos);  
ps.print(disp.getText());  
modified=false;  
}  
}  
}  
  
public void openFile()  
{  
int x=dialog.showOpenDialog(this);  
if(x==0){  
try  
{  
File f=dialog.getSelectedFile();  
fis=new FileInputStream(f);  
setTitle(f.getPath());  
disp.setText("");  
BufferedReader br=new BufferedReader(new InputStreamReader(fis));  
while((str=br.readLine())!=null)  
disp.append(str+"\n");  
disp.replaceRange("",disp.getText().length()-1,disp.getText().length());  
modified=false;  
}
```

```
        }catch(Exception e){}
    }
}

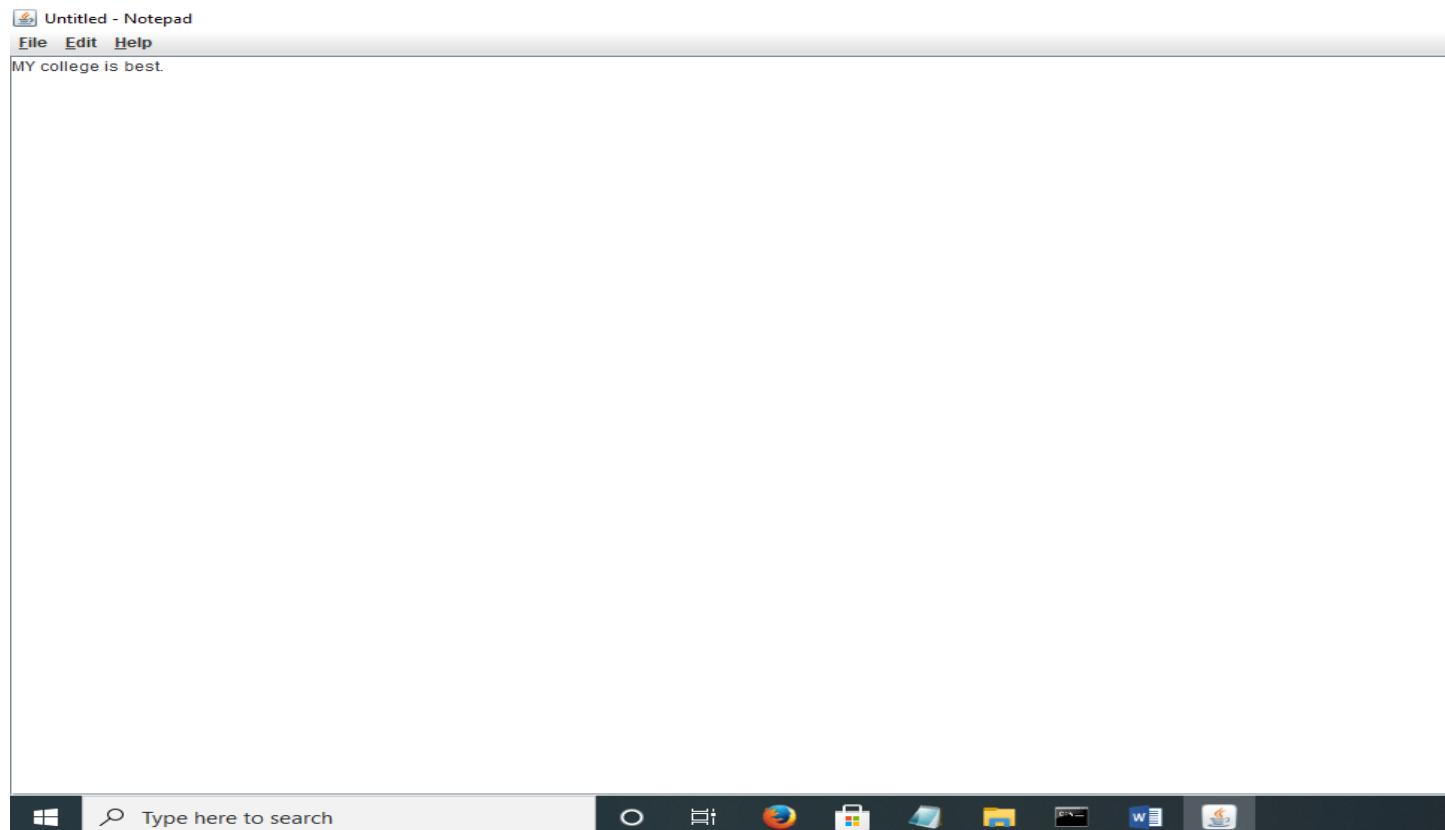
public int isModified()
{
    int x=JOptionPane.showConfirmDialog(this, "The text in the "+getTitle()+" has
changed\nDo you want to save it", "Notepad",
JOptionPane.YES_NO_CANCEL_OPTION,JOptionPane.WARNING_MESSAGE);

    if(x==JOptionPane.YES_OPTION) return 1;
    else if(x==JOptionPane.NO_OPTION) return 2;
    else return 3;
}

public void itemStateChanged(ItemEvent ie){}

public static void main(String[] args)
{
    new NotepadSwing();
}
```

}



10. Write a program for opening a file dialog on click on menu item and display content of selected file in a text area.

11. Write a panel class called EyePanel (and any other classes you need, such as event listener) to implement a panel that draws two eyes that look up. To the center, or down depending on whether the mouse cursor is above, inside, or below the eyes. The following three screenshot show the eyes of each of the three positions.

```
import java.awt.Color;  
  
import java.awt.Dimension;
```

```
import java.awt.Graphics;  
import java.awt.Point;  
import java.awt.event.MouseEvent;  
import java.awt.event.MouseMotionAdapter;  
  
import javax.swing.*;  
  
public class MovingEyes implements Runnable {  
  
    private static final int drawingWidth = 400;  
    private static final int drawingHeight = 400;  
    private static final int eyeballHeight = 150;  
    private static final int eyeballWidthMargin = 125;  
    private static final int eyeballOuterRadius = 50;  
    private static final int eyeballInnerRadius = 20;  
  
    private DrawingPanel drawingPanel;  
  
    private Eye[] eyes;  
  
    private JFrame frame;  
  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(new MovingEyes());
```

```
}
```

```
public MovingEyes() {  
  
    this.eyes = new Eye[2];  
  
    this.eyes[0] = new Eye(new Point(eyeballWidthMargin, eyeballHeight));  
  
    this.eyes[1] = new Eye(new Point(drawingWidth - eyeballWidthMargin - 50,  
        eyeballHeight));  
  
}
```

```
@Override
```

```
public void run() {  
  
    frame = new JFrame("Moving Eyes");  
  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    drawingPanel = new DrawingPanel();
```

```
    frame.add(drawingPanel);
```

```
    frame.pack();
```

```
    frame.setLocationByPlatform(true);
```

```
    frame.setVisible(true);
```

```
}
```

```
public class DrawingPanel extends JPanel {
```

```
public DrawingPanel() {  
    this.addMouseListener(new EyeballListener());  
    this.setBackground(Color.CYAN);  
    this.setPreferredSize(new Dimension(drawingWidth, drawingHeight));  
}  
  
@Override  
protected void paintComponent(Graphics g) {  
    super.paintComponent(g);  
  
    g.setColor(Color.BLACK);  
  
    for (Eye eye : eyes) {  
        drawCircle(g, eye.getOrigin(), eyeballOuterRadius);  
        fillCircle(g, eye.getEyeballOrigin(), eyeballInnerRadius);  
    }  
}  
  
private void drawCircle(Graphics g, Point origin, int radius) {  
    g.drawOval(origin.x - radius, origin.y - radius, radius + radius,  
               radius + radius);  
}
```

```
private void fillCircle(Graphics g, Point origin, int radius) {  
    g.fillOval(origin.x - radius, origin.y - radius, radius + radius,  
               radius + radius);  
}  
  
}  
  
public class Eye {  
    private final Point origin;  
    private Point eyeballOrigin;  
  
    public Eye(Point origin) {  
        this.origin = origin;  
        this.eyeballOrigin = origin;  
    }  
  
    public Point getEyeballOrigin() {  
        return eyeballOrigin;  
    }  
  
    public void setEyeballOrigin(Point eyeballOrigin) {  
        this.eyeballOrigin = eyeballOrigin;  
    }  
}
```

```
    public Point getOrigin() {  
        return origin;  
    }  
  
}
```

```
public class EyeballListener extends MouseMotionAdapter {
```

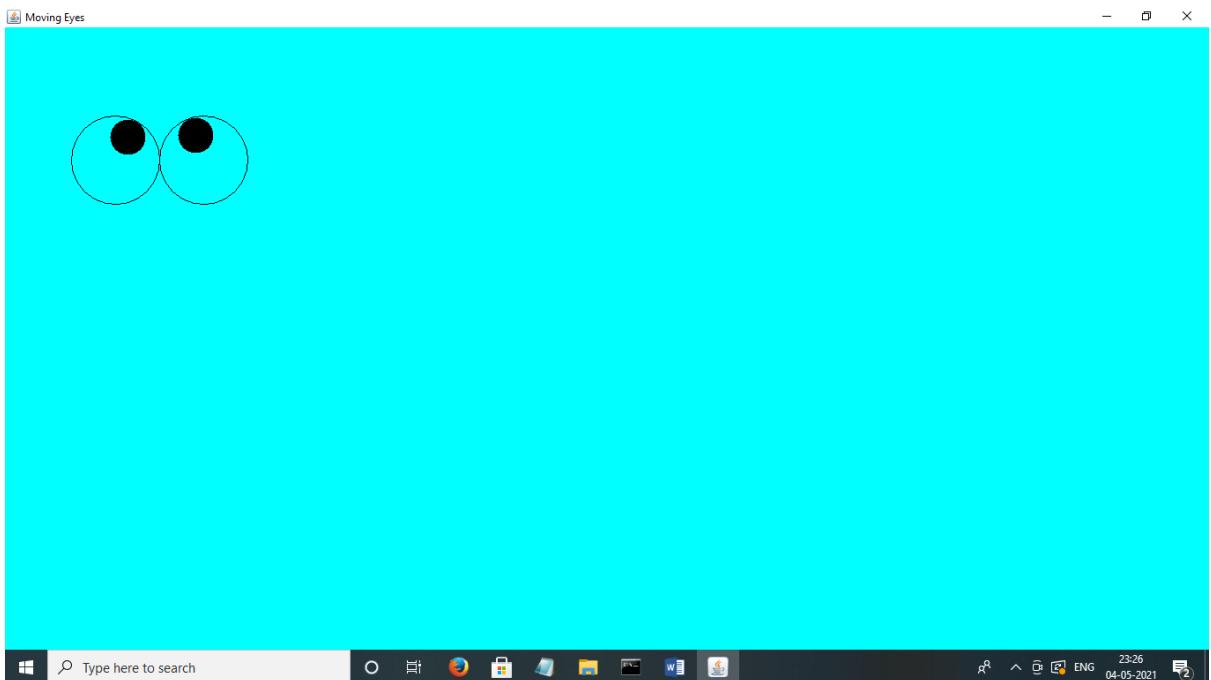
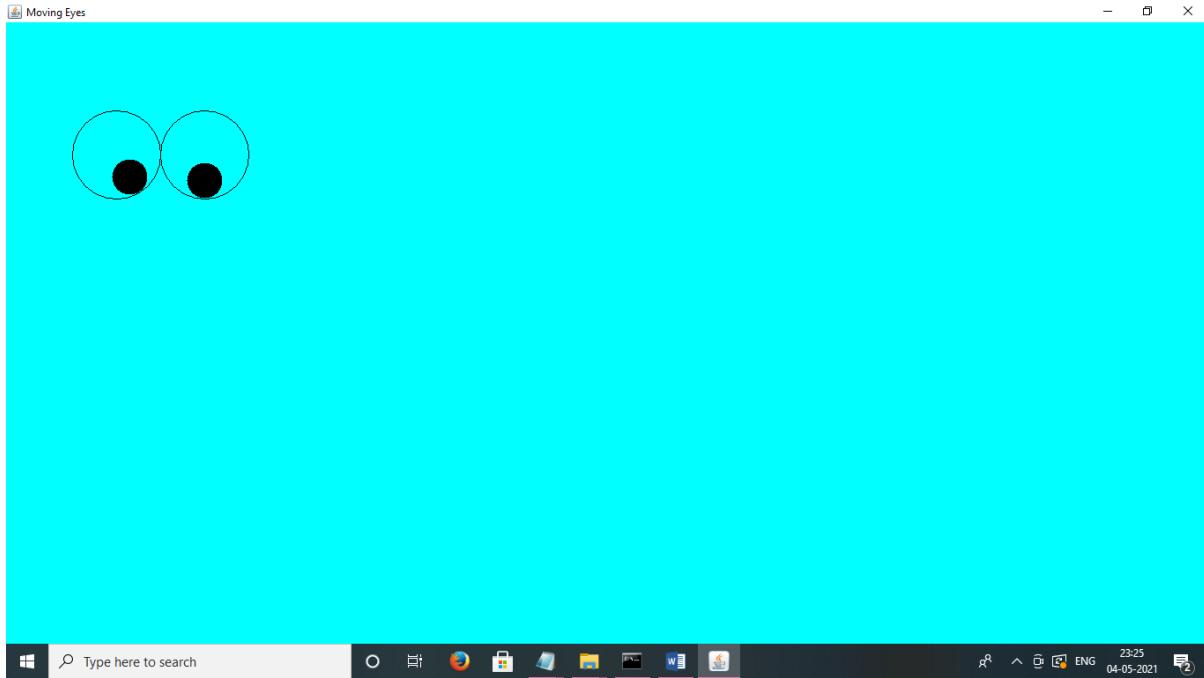
```
    private final double eyeballDistance = eyeballOuterRadius  
        - eyeballInnerRadius - 1;
```

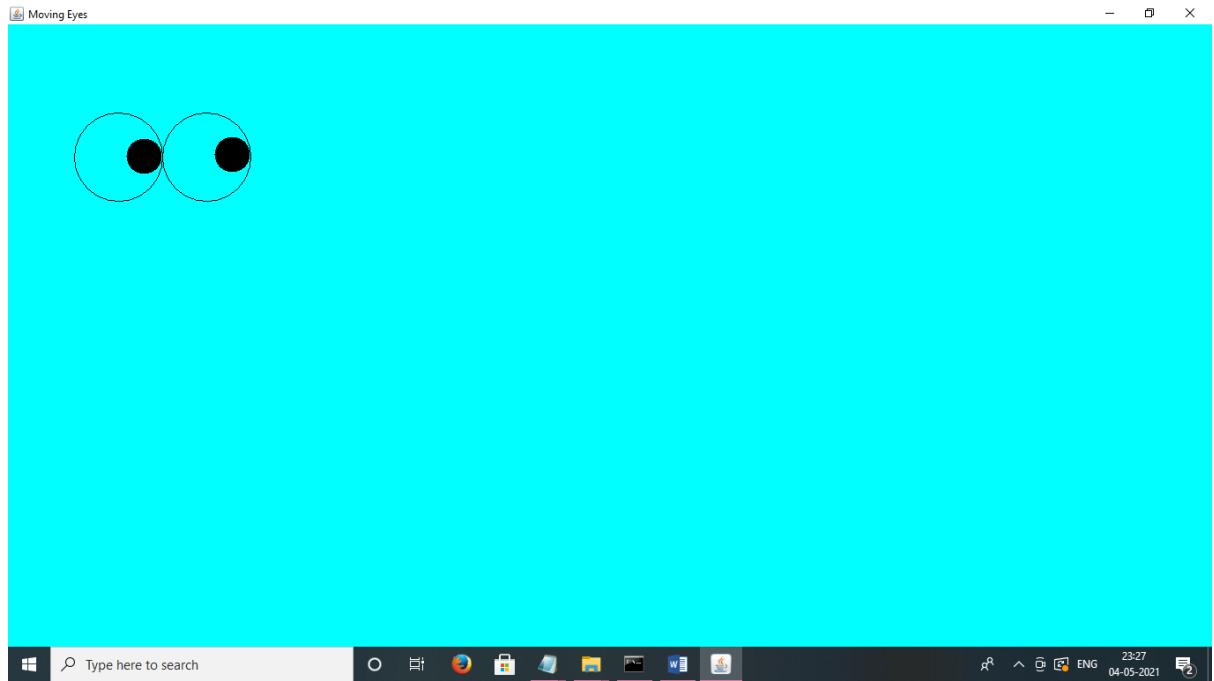
```
    @Override
```

```
    public void mouseMoved(MouseEvent event) {  
        Point p = event.getPoint();  
        for (Eye eye : eyes) {  
            Point origin = eye.getOrigin();  
            double theta = Math.atan2((double) (p.y - origin.y),  
                (double) (p.x - origin.x));  
            int x = (int) Math.round(Math.cos(theta) * eyeballDistance)  
                + origin.x;  
            int y = (int) Math.round(Math.sin(theta) * eyeballDistance)  
                + origin.y;  
            eye.setEyeballOrigin(new Point(x, y));  
        }  
    }
```

```
    drawingPanel.repaint();
```

```
} }
```





## JDBC connectivity

1.write a java program to create table student in a database named as fymca. It should have 5 columns. Stud\_id, stud\_name, stud\_last, division and grade.

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

public class createTable {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Connection con;
        try {
```

```

        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fymca", "root", "root");

        System.out.println("DB Connected...");
        Statement s = con.createStatement();

        String s1 = "create table Student(stud_id int(10) not null,
stud_name varchar(32), stud_last varchar(32), division varchar(10), Grade
varchar(10))";

        boolean b = s.execute(s1);

        System.out.println(b+"Table Created...");

        s.close();
        con.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}

```

2. write a program for performing CRUD(create, Update, select, delete) operation on student table.(add atleast 5 rows).

```

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

public class insertrecord {

    public static void main(String[] args) {
        try {
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fymca","root","root");
            System.out.println("Connected...");

            Statement s = con.createStatement();

```

```

        int i = s.executeUpdate("insert into Student values(1,'kiran',
'padole', 'B', 'A1')");

        System.out.println(i+" Data Inserted...");

        i = s.executeUpdate("insert into Student values(2,'jyoti',
'pund', 'B', 'A1')");

        System.out.println(i+" Data Inserted...");

        i = s.executeUpdate("insert into Student values(3,'swati',
'shinde', 'B', 'B1')");

        System.out.println(i+" Data inserted...");

        i = s.executeUpdate("insert into Student values(4,'deepa',
'sukhia', 'B', 'B2')");

        System.out.println(i+" Data inserted...");

        i= s.executeUpdate("insert into Student values(5,'ram', 'padole',
'B', 'C')");

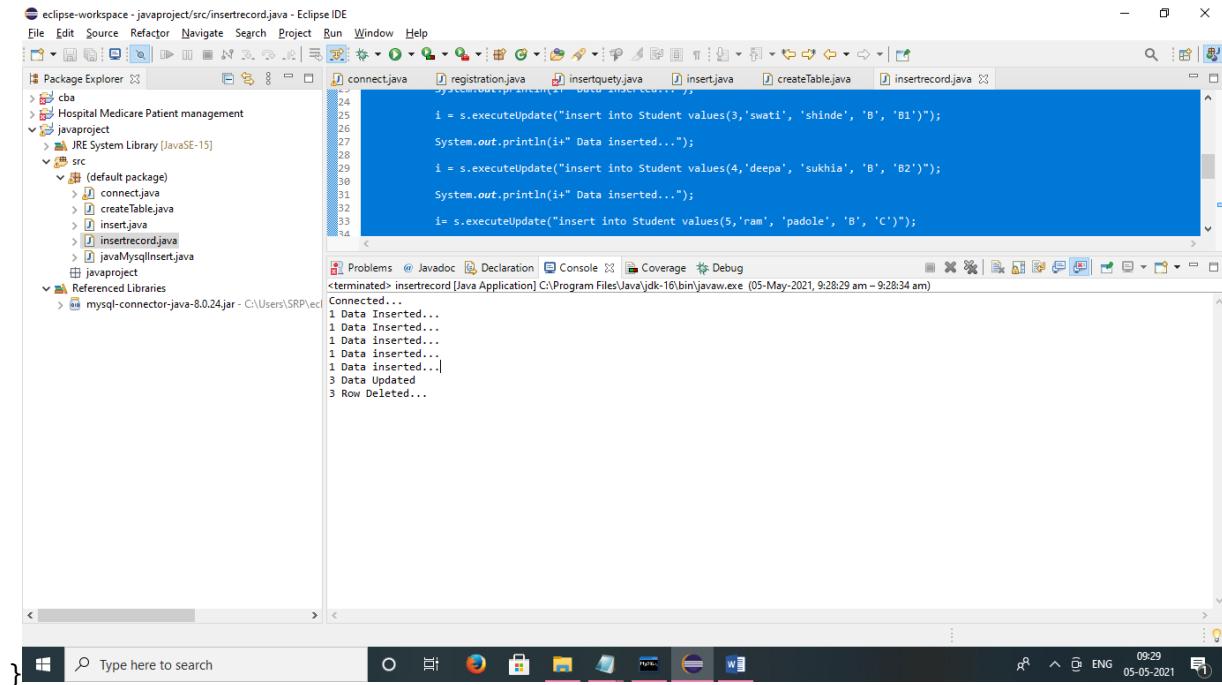
        System.out.println(i+" Data inserted...");

        i = s.executeUpdate("Update student set grade = 'A2' where stud_id =
2");

        System.out.println(i+" Data Updated");
        i = s.executeUpdate("Delete from student where stud_id = 5");
        System.out.println(i+" Row Deleted...");

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```



3. write a program to delete a specific row from a student table.

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

public class deleteTable {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try {
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fymca", "root", "root");
            Statement s = con.createStatement();

            int i = s.executeUpdate("Delete from student where stud_id = 2");
            System.out.println(" Row Deleted... ");

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - javaproject/src/deleteTable.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbars:** Standard, Java, Database, Text Editors, Navigator, Outline, Problems, Javadoc, Declaration, Console, Coverage, Debug.
- Left Sidebar (Package Explorer):** Shows the project structure:
  - javaProject
  - JRE System Library [JavaSE-15]
  - src
    - (default package)
      - connect.java
      - createtable.java
      - deletetable.java
      - dropable.java
      - insert.java
      - insertrecord.java
      - javysqlinsert.java
    - javaproject
  - Referenced Libraries
    - mysql-connector-java-8.0.24.jar - C:\Users\SRP\ec...
- Central Area (Code Editor):** Displays the Java code for `deleteTable.java`. The code connects to a MySQL database and executes a SQL query to delete a row from the `student` table where `stud_id = 2`.

```
1
2
3import java.sql.Connection;
4import java.sql.DriverManager;
5import java.sql.SQLException;
6import java.sql.Statement;
7
8public class deleteTable {
9
10    public static void main(String[] args) {
11        // TODO Auto-generated method stub
12
13        try {
14            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fymca", "root", "root");
15            Statement s = con.createStatement();
16
17            int i = s.executeUpdate("Delete from student where stud_id = 2");
18            System.out.println(" Row Deleted..");
19
20        } catch (SQLException e) {
21            // TODO Auto-generated catch block
22            e.printStackTrace();
23        }
24    }
25}
```
- Bottom Status Bar:** Writable, Smart Insert, 23 : 10 : 612, 09:37, ENG, 05-05-2021

4. write a java program to alter student table by adding column email\_id, address and drop a table.

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

public class altertable {

    public static void main(String[] args) {
        try {
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fymca", "root", "root");
            Statement st = con.createStatement();
            st.execute("Alter table student add Email_d varchar(30)");
            st.execute("Alter table student add address varchar(20)");
            System.out.println("Column Added...");
            st.execute("Drop Table student");
            System.out.println("Table Dropped");
        } catch (SQLException e) {
            // TODO Auto-generated catch block
        }
    }
}
```

```

        e.printStackTrace();
    }
}

}

eclipse-workspace - javaproject/src/altertable.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer InsertQuery... insert.java createTable... insertRecord... droptable.java altertable.java
src (default package)
> altertable.java
> connect.java
> deleteTable.java
> droptable.java
> insert.java
> insertRecord.java
> javaMysqlInsert.java
javaproject
Referenced Libraries mysql-connector-java-8.0.24.jar - C:\Users\SRP\ec...
1
2
3<import java.sql.Connection;
4<import java.sql.DriverManager;
5<import java.sql.SQLException;
6<import java.sql.Statement;
7
8 public class altertable {
9
10    public static void main(String[] args) {
11        try {
12            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fymca", "root", "root");
13            Statement st = con.createStatement();
14            st.execute("Alter table student add Email_id varchar(30)");
15            st.execute("Alter table student add address varchar(20)");
16            System.out.println("Column Added...");
17            st.execute("Drop Table student");
18            System.out.println("Table Dropped");
19        } catch (SQLException e) {
20            // TODO Auto-generated catch block
21            e.printStackTrace();
22        }
23    }
24
25
26
Problems Declaration Console Coverage Debug
<terminated> altertable [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe (05-May-2021, 9:48:28 am - 9:48:33 am)
Column Added...
Table Dropped
Windows Type here to search Writable Smart Insert 25: 2 [39]
R ENG 09:48 05-05-2021

```

5. write JDBC and swing program to create table, insert records and fetch student record from database and display it in text field.

```

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

```

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class FetchAndDisplay extends JFrame
{
    JFrame f;

    public FetchAndDisplay() {
        setTitle("Fetch Student Information");
        JLabel input, heading, stname, stmail, pass, country;
        JTextField tinput, tstname, tstmail, tpass, tcountry;
        JButton fetch;

        input = new JLabel("Enter Student Name: ");
        input.setBounds(30, 20, 150, 30);
        add(input);
```

```
tinput = new JTextField();  
tinput.setBounds(180, 20, 150, 30);  
add(tinput);  
  
fetch = new JButton("Fetch Record");  
fetch.setBounds(50,60,250,30);  
add(fetch);  
  
heading = new JLabel();  
heading.setText("<HTML><h1>Student Information Retrieving  
from Database!!!</h1></HTML>");  
heading.setForeground(Color.BLUE);  
heading.setBounds(30, 120, 600, 30);  
add(heading);  
  
stname = new JLabel("Student Name: ");  
stname.setBounds(30, 150, 150, 30);  
add(stname);  
  
tstname = new JTextField();
```

```
tstname.setBounds(180, 150, 150, 30);  
tstname.setEditable(false);  
add(tstname);
```

```
stmail = new JLabel("Mail_id:");  
stmail.setBounds(30, 200, 150, 30);  
add(stmail);
```

```
tstmail = new JTextField();  
tstmail.setBounds(180, 200, 150, 30);  
tstmail.setEditable(false);  
add(tstmail);
```

```
pass = new JLabel("Password:");  
pass.setBounds(30, 240, 150, 30);  
add(pass);
```

```
tpass = new JTextField();  
tpass.setBounds(180, 240, 150, 30);  
tpass.setEditable(false);  
add(tpass);
```

```
country = new JLabel("Country: ");
country.setBounds(30, 280, 150, 30);
add(country);

tcountry = new JTextField();
tcountry.setBounds(180, 280, 150, 30);
tcountry.setEditable(false);
add(tcountry);

fetch.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3360/fymca", "root",
"root");
            Statement st =
con.createStatement();
            String nm = tinput.getText();
            ResultSet rs =
st.executeQuery("Select * from student where sname = '"+nm+"'");
        }
    }
})
```

```
        while(rs.next())  
        {  
            String n = rs.getString(1);  
            tstname.setText(n);  
            String m = rs.getString(2);  
            tstmail.setText(m);  
            String p = rs.getString(3);  
            tpass.setText(p);  
            String c = rs.getString(4);  
            tcountry.setText(c);  
        }  
    } catch (SQLException e1) {  
        // TODO Auto-generated catch block  
        e1.printStackTrace();  
    }  
}  
});  
  
setSize(800, 800);  
setLayout(null);  
setVisible(true);
```

```

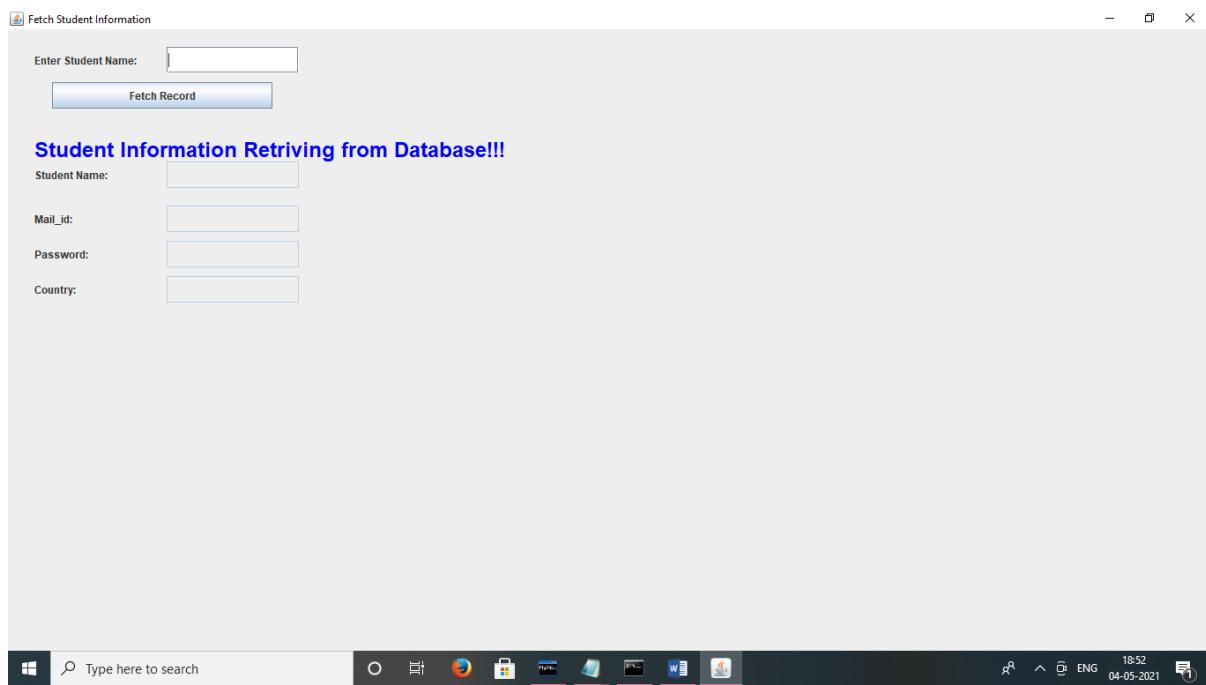
        setDefaultCloseOperation(EXIT_ON_CLOSE);

    }

    public static void main(String[] args) {
        new FetchAndDisplay();
    }

}

```



6. Design a following Phone Book Application Screen using swing. Display proper message if invalid data is entered like name left blank and negative phone number. Using MYSQL store the values in the table phone (Name, Address,

Phone, Email id) if valid data is entered for all the fields and perform the various operations like Add, Delete, Next and Previous as shown on the screen.

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
//import java.sql.DriverManager;
//import java.sql.ResultSet;
//import java.sql.SQLException;
//import java.sql.Statement;

import javax.swing.*;

public class PhoneBook extends JFrame
{

    JFrame f;
    PhoneBook() throws SQLException
    {
        setTitle("Phone Book Application");
        JLabel name, address, phone, email, res;
        JTextField tname, tphone, temail;
        JTextArea taddress;
        JButton add, del, nex, prev;

        name = new JLabel("Name: ");
        name.setBounds(30, 20, 100, 30);
        add(name);

        tname = new JTextField();
        tname.setBounds(130, 20, 350, 30);
        add(tname);

        address = new JLabel("Address: ");
        address.setBounds(30, 60, 100, 30);
        add(address);

        taddress = new JTextArea();
        taddress.setBounds(130, 60, 350, 80);
        add(taddress);

        phone = new JLabel("Phone: ");
        phone.setBounds(30, 150, 100, 30);
        add(phone);

        tphone = new JTextField();
        tphone.setBounds(130, 150, 350, 30);
        add(tphone);

        email = new JLabel("Email: ");
        email.setBounds(30, 190, 100, 30);
        add(email);
```

```

temail = new JTextField();
temail.setBounds(130, 190, 350, 30);
add(temail);

res = new JLabel("Enter Details...");
res.setBounds(75, 250, 350, 30);
add(res);

add = new JButton("ADD");
add.setBounds(500, 40, 100, 40);
add(add);

del = new JButton("DELETE");
del.setBounds(500, 90, 100, 40);
add(del);

nex = new JButton("NEXT");
nex.setBounds(500, 140, 100, 40);
add(nex);

prev = new JButton("PREVIOUS");
prev.setBounds(500, 190, 100, 40);
add(prev);

Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fymca", "root", "root");

add.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e)
    {
        String s = tname.getText();
        String s1 = taddress.getText();
        String s2 = tphone.getText();
        String s3 = temail.getText();
        if(!(s.equals("")) && s1.equals("") && s2.equals("") &&
s3.equals("")))
        {
            try {
                res.setText("");
                Statement st = con.createStatement();
                st.executeUpdate("Insert into phonebook
values('"+tname.getText()+"','"+taddress.getText()+"','"+tphone.getText()+"','"+temai
l.getText()+"')");
                res.setText("Details Added successfully...");
                JOptionPane.showMessageDialog(f, "Details added in
database.");
            } catch (SQLException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
        }
    }
});

```

```

        else
        {
            JOptionPane.showMessageDialog(f, "Can't keep any
empty field", "Alert", JOptionPane.ERROR_MESSAGE);
        }
    });
}

del.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        String s = tphone.getText();
        if(!(s.equals("")))) {
            try {
                res.setText("");
                Statement st = con.createStatement();
                st.executeUpdate("Delete from phonebook where tname
= '"+tphone.getText()+"'");
                res.setText("Record Deleted...");
                JOptionPane.showMessageDialog(f, "Record Deleted
from Phonebook.");
            }
            catch(SQLException e1) {
                e1.printStackTrace();
            }
        }
        else {
            JOptionPane.showMessageDialog(f, "Please Enter the
Phone number to be deleted!!!", "Alert", JOptionPane.ERROR_MESSAGE);
        }
    }
});

nex.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try
        {
            Statement st =
con.createStatement(resultSet.TYPE_SCROLL_SENSITIVE, resultSet.CONCUR_READ_ONLY);
            ResultSet rs = st.executeQuery("Select * from
phonebook");
            while(rs.next())
            {
                tname.setText(rs.getString(1));
                taddress.setText(rs.getString(2));
                tphone.setText(rs.getString(3));
                temail.setText(rs.getString(4));
            }
        }
    }
});

```

```

        }
        catch(SQLException e1) {
            e1.printStackTrace();
        }
    });
}

prev.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try
        {
            Statement st =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
            ResultSet rs = st.executeQuery("Select * from
phonebook");
            rs.last();
            while(rs.previous())
            {
                tname.setText(rs.getString(1));

                taddress.setText(rs.getString(2));

                tphone.setText(rs.getString(3));

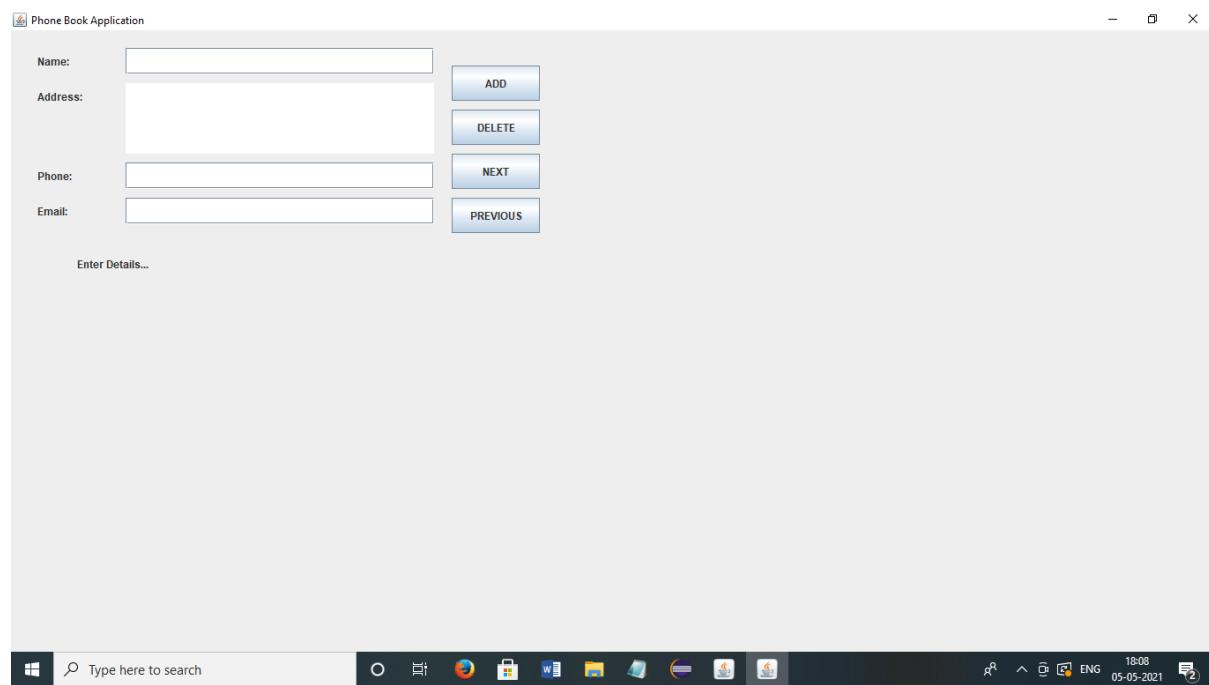
                temail.setText(rs.getString(4));

            }
        }
        catch(SQLException e1) {
            e1.printStackTrace();
        }
    });
}

setSize(800, 800);
setLayout(null);
setVisible(true);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}
public static void main(String[] args) throws SQLException {
    new PhoneBook();
}
}
}

```

For fe



7. Write a program for fetching data from database and displaying content of database in JTable.

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class fetchData implements ActionListener {

    JFrame frame1;
    JLabel nameLabel;
    JTextField nameTextField;    JButton fetchButton;    JButton resetButton;
    JFrame frame2;
    DefaultTableModel defaultTableModel;
    JTable table;
    Connection connection;
```

```

Statement statement;      int flag=0;

fetchData() {

    frame1 = new JFrame();
    frame1.setTitle("Search Database");
    frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    GridBagLayout bagLayout = new GridBagLayout();
    GridBagConstraints bagConstraints = new GridBagConstraints();
    frame1.setSize(500, 300);
    frame1.setLayout(bagLayout);

    bagConstraints.insets = new Insets(15, 40, 0, 0);

    nameLabel = new JLabel("Enter username.");
    bagConstraints.gridx = 0;
    bagConstraints.gridy = 0;
    frame1.add(nameLabel, bagConstraints);

    nameTextField = new JTextField(15);
    bagConstraints.gridx = 1;
    bagConstraints.gridy = 0;
    frame1.add(nameTextField, bagConstraints);

    fetchButton = new JButton("Fetch Data");
    bagConstraints.gridx = 0;
    bagConstraints.gridy = 1;
    bagConstraints.ipadx = 60;
    frame1.add(fetchButton, bagConstraints);

    resetButton = new JButton("Reset Data");
    bagConstraints.gridx = 1;
    bagConstraints.gridy = 1;
    frame1.add(resetButton, bagConstraints);

    fetchButton.addActionListener(this);
    resetButton.addActionListener(this);

    frame1.setVisible(true);
    frame1.validate();
}

public static void main(String[] args) {
    new fetchData();
}

@Override
public void actionPerformed(ActionEvent e) {

    if (e.getSource() == fetchButton) {

```

```

        String userName = nameTextField.getText().toString();
        frameSecond(userName);

    }

    if (e.getSource() == resetButton) {
        nameTextField.setText("");
    }

}

public void frameSecond(String userName) {

    frame2 = new JFrame("Database Results");
    frame2.setLayout(new FlowLayout());
    frame2.setSize(400, 400);

    defaultTableModel = new DefaultTableModel();
    table = new JTable(defaultTableModel);
    table.setPreferredScrollableViewportSize(new Dimension(300, 100));
    table.setFillsViewportHeight(true);
    frame2.add(new JScrollPane(table));
    defaultTableModel.addColumn("Username");
    defaultTableModel.addColumn("Roll No");
    defaultTableModel.addColumn("Department");




try {

    connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fymca", "root", "root");
    statement = connection.createStatement();
    String query = "select * from STUDENT where USERNAME = '" + userName +
"";;
    ResultSet resultSet = statement.executeQuery(query);

    while (resultSet.next()) {

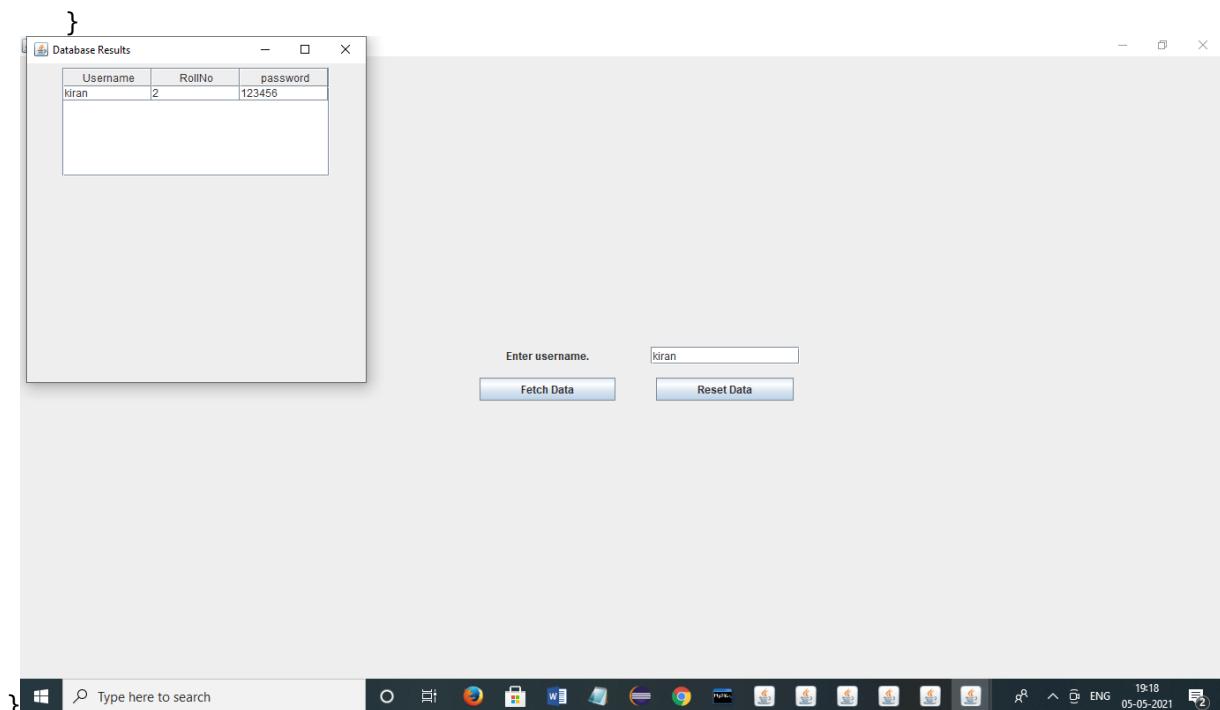
        String name = resultSet.getString("Username");
        String roll = resultSet.getString("RollNo");
        String dept = resultSet.getString("password");
        if (userName.equalsIgnoreCase(name)) {
            flag = 1;
            defaultTableModel.addRow(new Object[]{name, roll, dept});
            frame2.setVisible(true);
            frame2.validate();
            break;
        }
    }
}

```

```
}

if (flag == 0) {
    JOptionPane.showMessageDialog(null, "No Such Username Found");
}

} catch (SQLException throwables) {
    throwables.printStackTrace();
}
```



## Assignment No. - 6

1. Write a Servlet application for taking values from user through a web page and display the contents in tabular form.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form method="POST" action="welcome">
Enter your Full Name : <input type="text"
name="name"><br><br>
Enter your address : <input type="text"
name="address"><br><br>
Enter your college Name: <input type="text"
name="clg"><br><br>
<input type="submit" value="Submit">

</body>
</html>
```

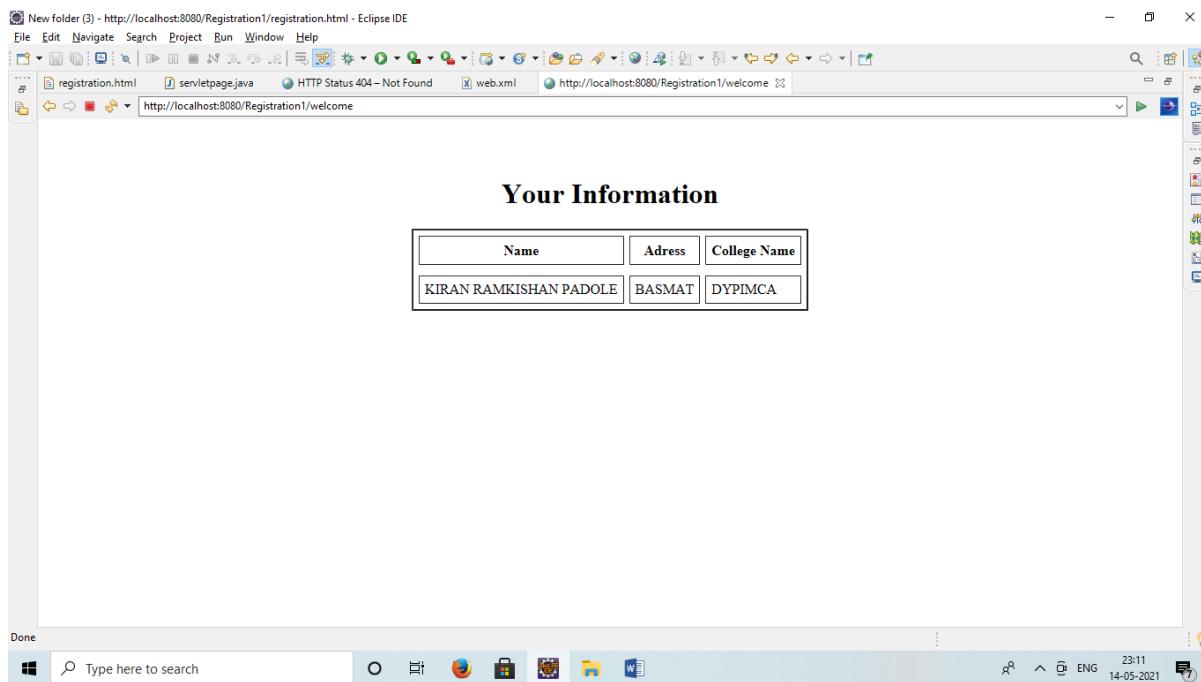
### Servletpage.java

```
import java.io.IOException;
import java.io.PrintWriter;
```

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

public class servletpage extends HttpServlet{

    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException
    {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        String name = request.getParameter("name");
        String address =
request.getParameter("address");
        String clg = request.getParameter("clg");
        out.println("<html><body><center><br><br>");
        out.println("<h1> Your Information </h1>");
        out.println("<table border=\"2\""
cellspacing="6" cellpadding="6" align="center">");
        out.println("<tr><font
color=black><th>Name</th><th>Adress</th><th>College
Name</th></font><tr>");
        out.println("<tr><td> " + name + "</td><td> " +
address + "</td><td> " + clg + "</td></tr>");
    }
}
```



1. Write a servlet application for accepting hobbies of student through a web page using check boxes and display selected hobbies with some details about it.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <form action = "Hobbie" method = "post" >
        <input type = "checkbox" name = "sing"> Singing<br>
        <input type = "checkbox" name = "dance">
Dancing<br>
        <input type = "checkbox" name = "cook"> Cooking<br>
        <input type = "checkbox" name = "read"> Reading<br>
        <input type = "submit" value = "Select Hobbie">
    </form>
```

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

## HobbiesServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;

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

public class HobbiesServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException
    {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        out.println("<html>\n" +"<body>\n" +"<ul>\n"
+"<li><b> SINGING : </b>:"+ request.getParameter("sing") +
"\n" + " <li><b> DANCING : </b>: "+
request.getParameter("dance") + "\n" + " <li><b> COOKING : </b>: "+ request.getParameter("cook") + "\n" + " <li><b> READING : </b>: "+ request.getParameter("read") + "\n"
+"</ul>\n" +"</body>"+"</html>");
    }
}
```

## Output:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Displays the project structure for "Registration1". It includes a Deployment Descriptor, JAX-WS Web Services, a JRE System Library (JavaSE-15), and a src/main/java directory containing a default package with HobbiesServlet.java and servletpage.java, and a main directory with HobbiesServlet.java and servletpage.java.
- Web Browser:** Shows the URL <http://localhost:8080/Registration1/Hobbies.html>. The page displays a list of hobbies with checkboxes:
  - Singing (unchecked)
  - Dancing (checked)
  - Cooking (checked)
  - Reading (checked)A dropdown menu labeled "Select Hobbie" is visible.
- Outline and Task List:** Both are empty.
- Servers:** Shows "Tomcat v10.0 Server at localhost [Started, Synchronized]".
- Bottom Bar:** Includes a search bar, a ribbon of icons, and system status information (23:28, ENG, 14-05-2021).

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Displays the project structure for "Registration1". It includes a Deployment Descriptor, JAX-WS Web Services, a JRE System Library (JavaSE-15), and a src/main/java directory containing a default package with HobbiesServlet.java and servletpage.java, and a main directory with HobbiesServlet.java and servletpage.java.
- Web Browser:** Shows the URL <http://localhost:8080/Registration1/Hobbies.html>. The page displays a list of hobbies with checkboxes:
  - SINGING : null
  - DANCING : :on
  - COOKING : :on
  - READING : :on
- Outline and Task List:** Both are empty.
- Servers:** Shows "Tomcat v10.0 Server at localhost [Started, Synchronized]".
- Bottom Bar:** Includes a search bar, a ribbon of icons, and system status information (23:27, ENG, 14-05-2021).

2. Write a servlet application for storing student's registration details in database.

### StoringStudentInfo.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="StudentFormServlet" method="POST">
Enter your Full Name : <input type="text" name="name"><br>
Enter your Address : <input type="text" name="address"><br>
Enter your E-mail ID : <input type="emailid" name="emailid"><br>
Enter your Username : <input type="text" name="username"><br>
Enter your Password : <input type="password" name="password"><br>
<input type="submit" value="Submit">

</body>
</html>
```

### StudentInfoServlet.java

```
import java.io.IOException;
import java.io.*;
import java.sql.DriverManager;

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

import java.sql.*;
```

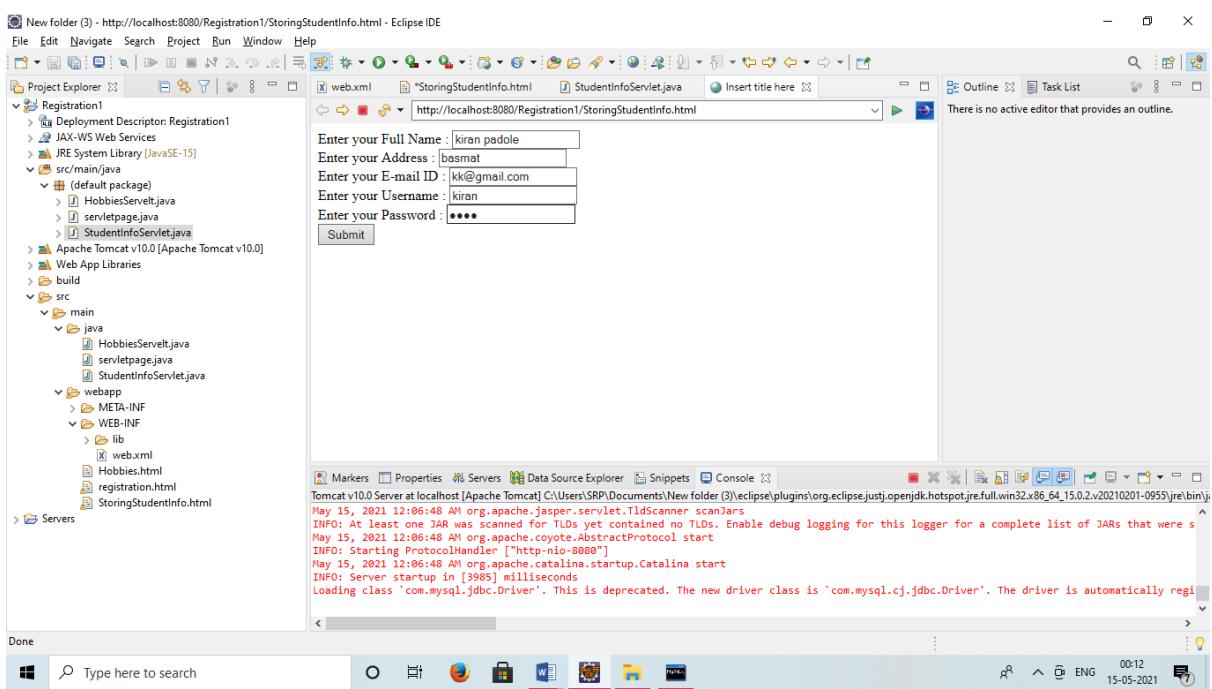
```
public class StudentInfoServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;

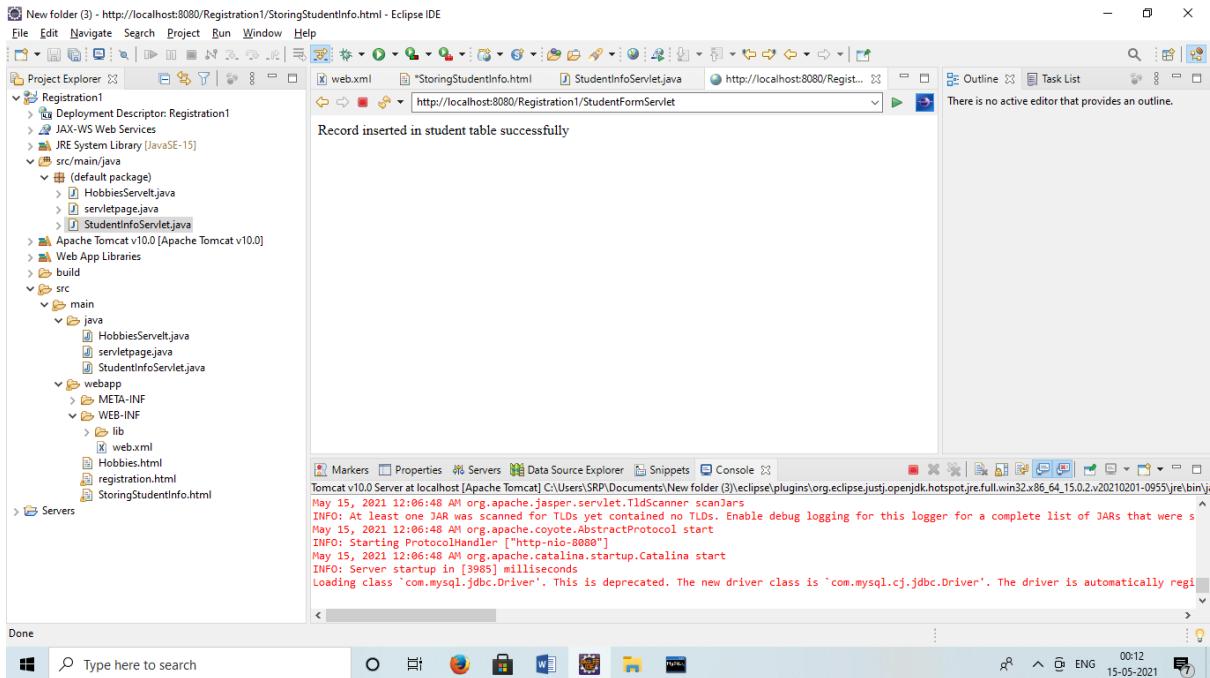
    protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException,
    IOException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306
/demo","root","root");
            Statement st=con.createStatement();
            String sname = request.getParameter("name");
            String saddress =
request.getParameter("address");
            String semailid =
request.getParameter("emailid");
            String susername =
request.getParameter("username");
            String spassword =
request.getParameter("password");
            String str = "insert into student
values('"+sname+"','"+saddress+"','"+semailid+"','"+susername
+"','"+spassword+"')";
            int k = st.executeUpdate(str);
            if(k>0)
            {
                out.println("Record inserted in student
table successfully");
            }
        }
    }
}
```

```

        catch(Exception e)
{
    out.println(e);
}
}
}

```





3. Write a Servlet application to create an html page for accepting employee details like: Name, password, email id, country and store it in database. Perform CRUD operation on that table like, edit, delete, retrieve, and update.

## CURD.HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="SaveServlet.java" method="post">
<table>
<tr><td>Employee Name:</td><td><input type="text" name="name"/></td></tr>
```

```

<tr><td>Employee Password:</td><td><input
type="password" name="password"/></td></tr>
<tr><td>Employee Email:</td><td><input type="email"
name="email"/></td></tr>
<tr><td>Employee Country:</td><td>
<select name="country" style="width:150px">
<option>India</option> <option>USA</option>
<option>UK</option> <option>Other</option>
</select> </td></tr> <br>
<br> <tr><td colspan="2"><input type="submit"
value="Save Employee"/></td></tr>
</table> </form> <br/>
<a href="ViewServlet">View Information of
Employees</a>
</body>

</html>

```

## SaveServlet.java

```

import java.io.IOException;
import java.io.PrintWriter;

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

public class SaveServlet extends HttpServlet
{
    /**
     *
     */
    private static final long serialVersionUID = 1L;

```

```
protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {
    response.setContentType("text/html");
    PrintWriter out=response.getWriter();

    String name=request.getParameter("name");
    String password=request.getParameter("password");
    String email=request.getParameter("email");
    String country=request.getParameter("country");

    Emp e=new Emp();
    e.setName(name);
    e.setPassword(password);
    e.setEmail(email);
    e.setCountry(country);

    int status=EmpDeo.save(e);
    if(status>0){
        out.print("<p>Record saved
successfully!</p>");
    }else{
        out.println("Sorry! unable to save record");
    }

    out.close();
}

import java.io.IOException;
import java.io.PrintWriter;

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

public class SaveServlet extends HttpServlet
{
    /**
     *
     */
    private static final long serialVersionUID = 1L;
```

```

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out=response.getWriter();

    String name=request.getParameter("name");
    String password=request.getParameter("password");
    String email=request.getParameter("email");
    String country=request.getParameter("country");

    Emp e=new Emp();
    e.setName(name);
    e.setPassword(password);
    e.setEmail(email);
    e.setCountry(country);

    int status=EmpDeo.save(e);
    if(status>0){
        out.print("<p>Record saved successfully!</p>");
    }else{
        out.println("Sorry! unable to save record");
    }
    out.close();
}

}

```

## EditServlet.java

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/EditServlet")
public class EditServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        out.println("<h1>Update Employee</h1>");
    }
}

```

```

String sid=request.getParameter("id");
int id=Integer.parseInt(sid);

Emp e=EmpDao.getEmployeeById(id);

out.print("<form action='EditServlet2' method='post'>");
out.print("<table>");
out.print("<tr><td></td><td><input type='hidden' name='id' value='"++e.getId()+"'/></td></tr>");
out.print("<tr><td>Name:</td><td><input type='text' name='name' value='"++e.getName()+"'/></td></tr>");
out.print("<tr><td>Password:</td><td><input type='password' name='password' value='"++e.getPassword()+"'/></td></tr>");
out.print("<tr><td>Email:</td><td><input type='email' name='email' value='"++e.getEmail()+"'/></td></tr>");
out.print("<tr><td>Country:</td><td>");
out.print("<select name='country' style='width:150px'>");
out.print("<option>India</option>");
out.print("<option>USA</option>");
out.print("<option>UK</option>");
out.print("<option>Other</option>");
out.print("</select>");
out.print("</td></tr>");
out.print("<tr><td colspan='2'><input type='submit' value='Edit & Save' /></td></tr>");
out.print("</table>");
out.print("</form>");

out.close();
}
}

```

## EditServlet2.java

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/EditServlet2")
public class EditServlet2 extends HttpServlet {
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();

        String sid=request.getParameter("id");
        int id=Integer.parseInt(sid);
        String name=request.getParameter("name");
        String password=request.getParameter("password");
        String email=request.getParameter("email");
        String country=request.getParameter("country");

        Emp e=new Emp();
        e.setId(id);
        e.setName(name);
        e.setPassword(password);
        e.setEmail(email);
        e.setCountry(country);

        int status=EmpDao.update(e);
        if(status>0){
            response.sendRedirect("ViewServlet");
        }else{
            out.println("Sorry! unable to update record");
        }
    }
}
```

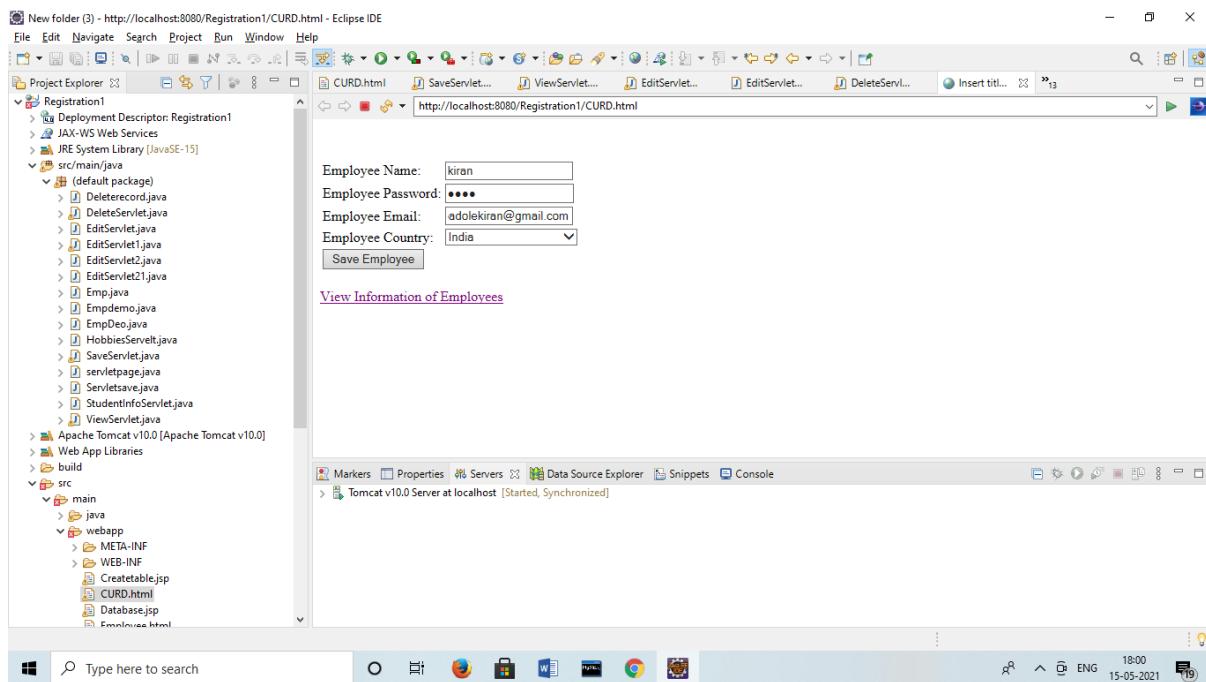
```
        out.close();
    }

}
```

## DeleteServlet.java

```
import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/DeleteServlet")
public class DeleteServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        String sid=request.getParameter("id");
        int id=Integer.parseInt(sid);
        EmpDao.delete(id);
        response.sendRedirect("ViewServlet");
    }
}
```



## 5. Write a JSP program to design an Employee registration form.

Following operation should happen:

- If employee is registered, he should be able to login.
- New employee should register first and then employee can register.
- Employee details should be stored in database. (at least 5 details)

### Registration.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<body bgcolor="skyblue"><center><h1> Register Now !! </h1>
<form action="servletpage.jsp" method="post"><br>
```

```

Enter First name <input type="text" name="fname" /><br><br>
Enter Last name <input type="text" name="lname" /><br><br>
Enter Email ID <input type="text" name="email" /><br><br>
Enter User name <input type="text" name="userid" /><br><br>
Enter Password <input type="password" name="password"
/><br><br><br>
<input type="submit" value="Register" />

</body>
</html>

```

## Servlefile.jsp

```

<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.* , java.util.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<%
String fname=request.getParameter("fname");
String lname=request.getParameter("lname");
String email=request.getParameter("email");
String userid=request.getParameter("userid");
String password=request.getParameter("password");
try
{

```

```
Class.forName("com.mysql.jdbc.Driver");
Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fym
ca", "root", "root");
Statement st=conn.createStatement();
int i=st.executeUpdate("insert into
Registration(Firstname,Lastname,emailid,Username,Password)va
lues('"+fname+"','"+lname+"','"+email+"','"+userid+"','"+pas
sword+"')");
out.println("Thank you for registering with us ! Please <a
href='Login.html'>Login</a>");
}
catch(Exception e)
{
System.out.print(e);
e.printStackTrace();
}
%>

</body>
</html>
```

## Login.html

```
<!DOCTYPE html>
```

```

<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<body bgcolor="blue">
<center>
<h1> Login !! </h1><br><br>
<form action="Login.jsp" method="post">
Enter User name <input type="text" name="usr" /><br><br><br>
Enter Password <input type="password" name="password"
/><br><br><br>
<input type="submit" value="Login"/> </form><br><br>
</center>

</body>
</html>

```

## Login.jsp

```

<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.* , java.util.*"%>

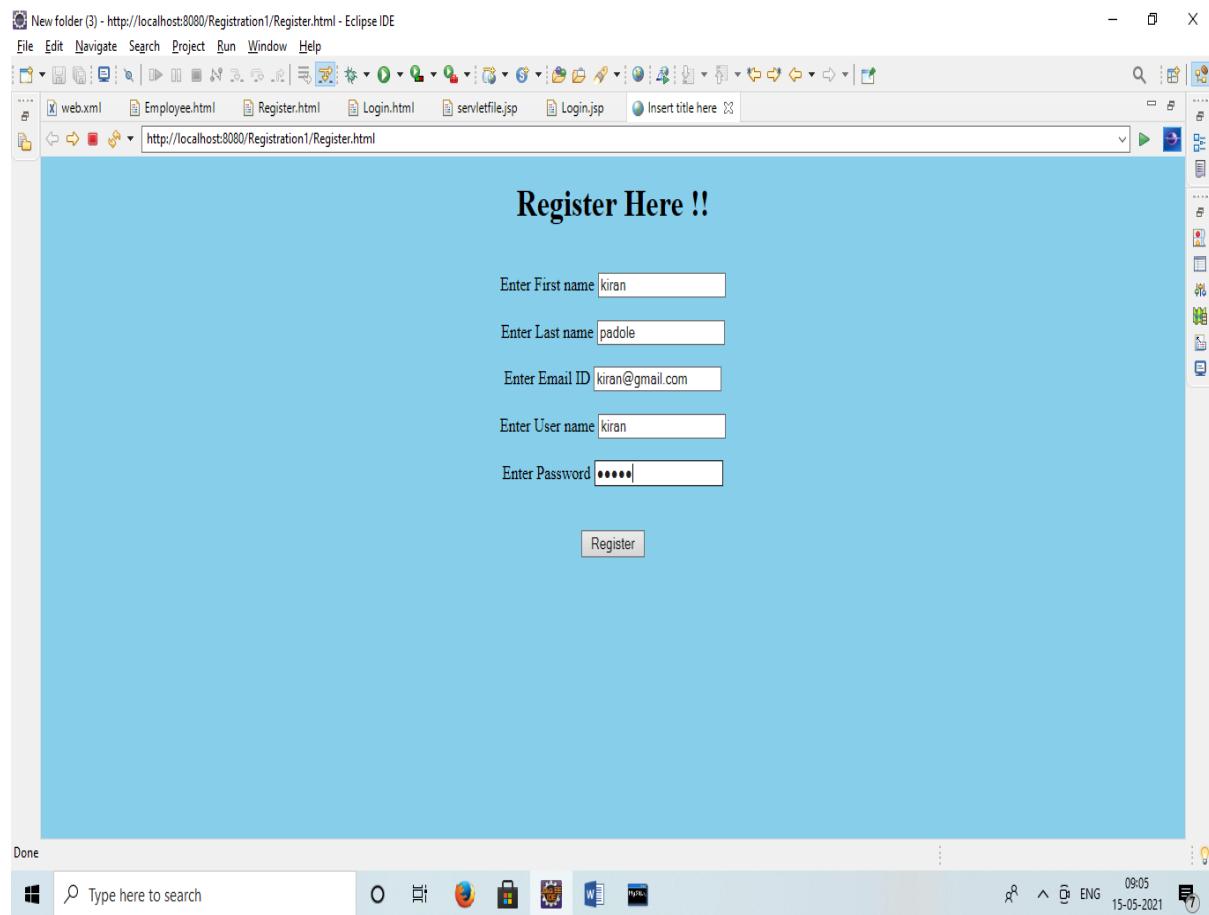
<%
String Username=request.getParameter("usr");
String Password=request.getParameter("password");
Class.forName("com.mysql.jdbc.Driver");
java.sql.Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fym
ca","root","root");
Statement st= con.createStatement();
ResultSet rs=st.executeQuery("select * from Registration
where Username='"+Username+"' and Password='"+Password+"'");
try{
rs.next();

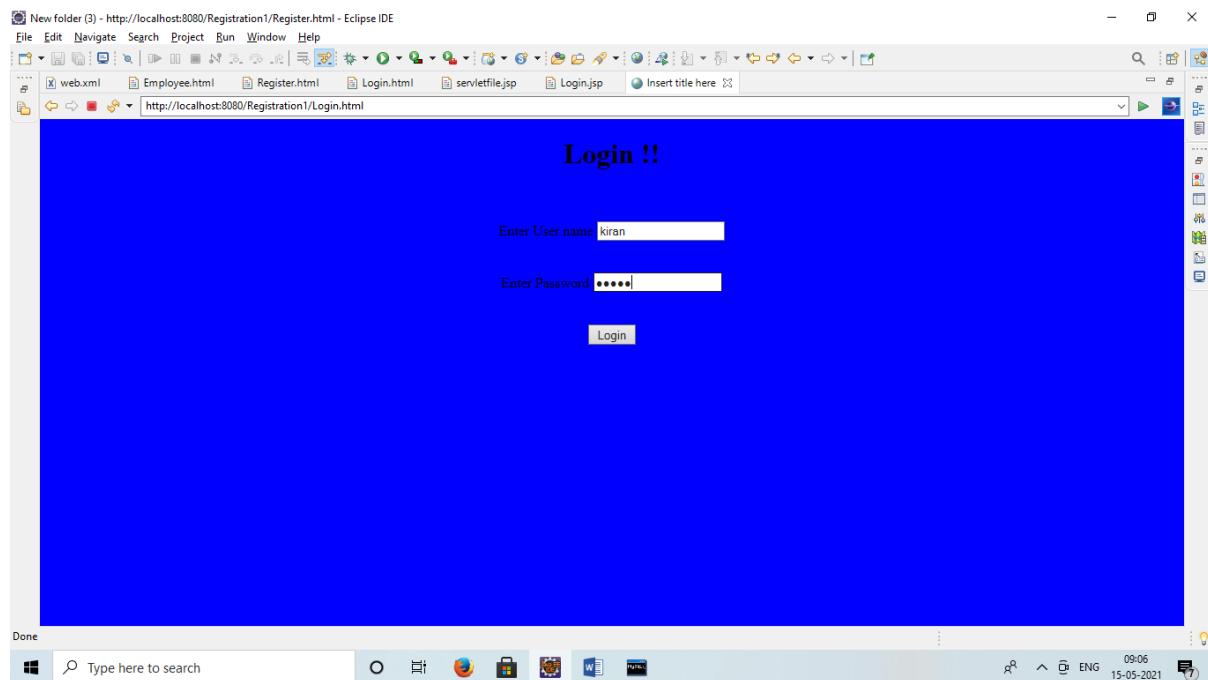
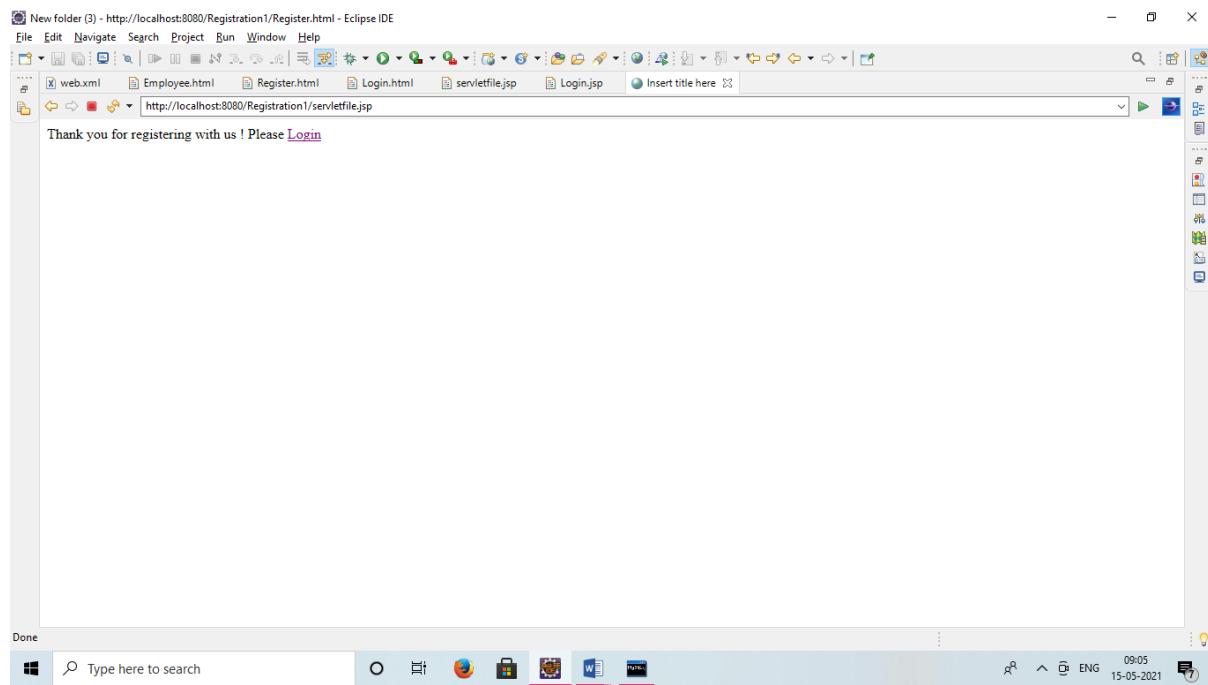
```

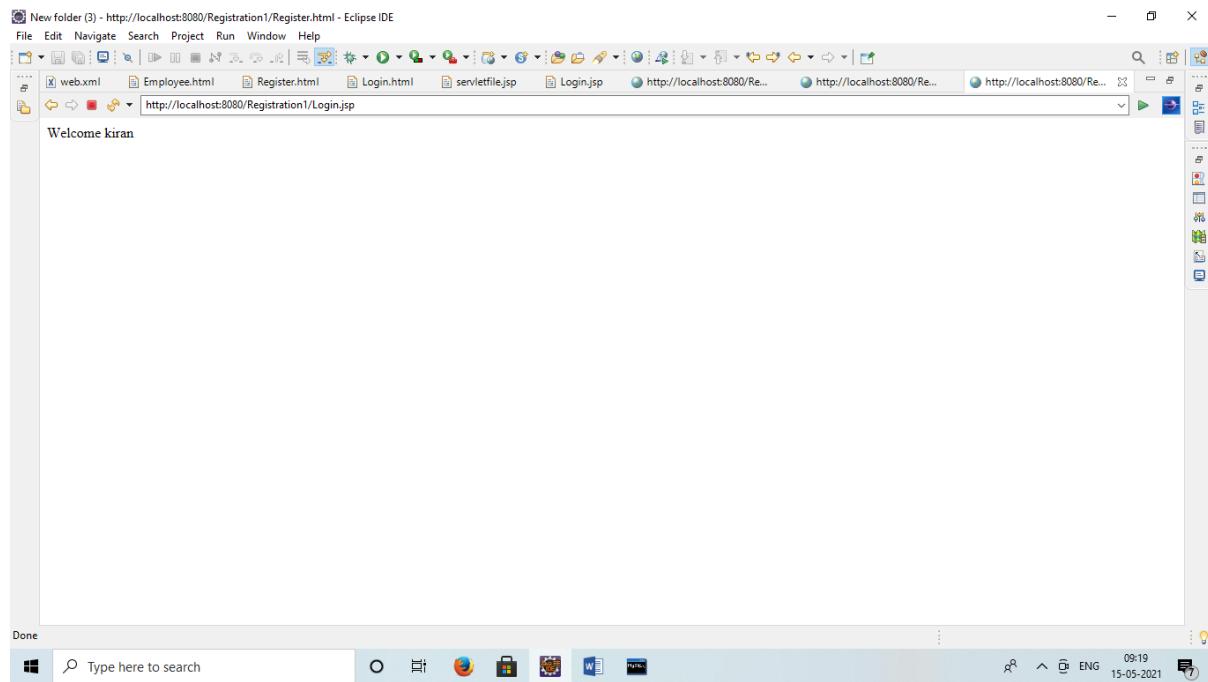
```

if(rs.getString("Password").equals(Password) &&
rs.getString("Usernamer").equals(username))
{
out.println("Welcome " +username);
}
else{
out.println("Invalid password or Username.");
}
}
catch (Exception e) {
e.printStackTrace();
}
%>

```







6. Write a JSP program for performing following operations on click of buttons (Take suitable input from user):
- a. Find square of entered number.
  - b. Check whether entered number is prime or not
  - c. Display current date.
  - d. Check whether entered number is Even number or not

PerformOperation.html

```
<!DOCTYPE html>
<html>
```

```

<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body bgcolor="pink">
<form action="Operation.jsp" method="post">
<center><h1> Perform Operations on JSP </h1> <br><br>
Enter any number <input type="text"
name="num"><br><br>
<input type="submit" value="Find Calculations">
</center>
</form>

</body>
</html>

```

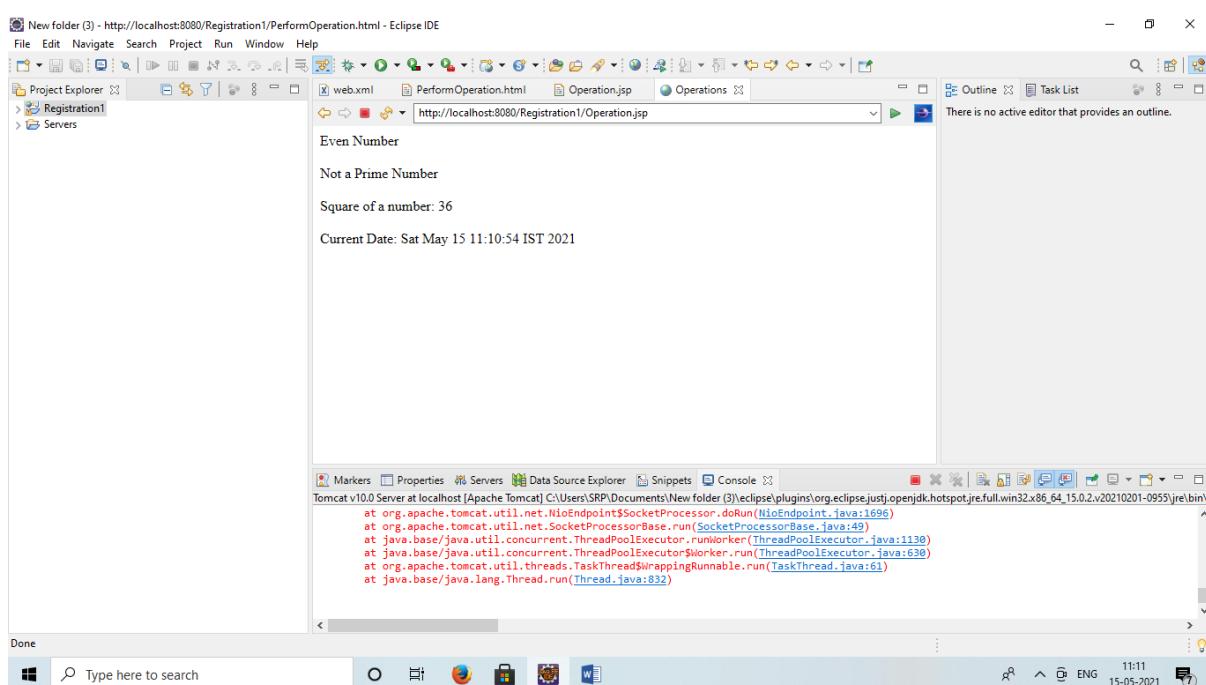
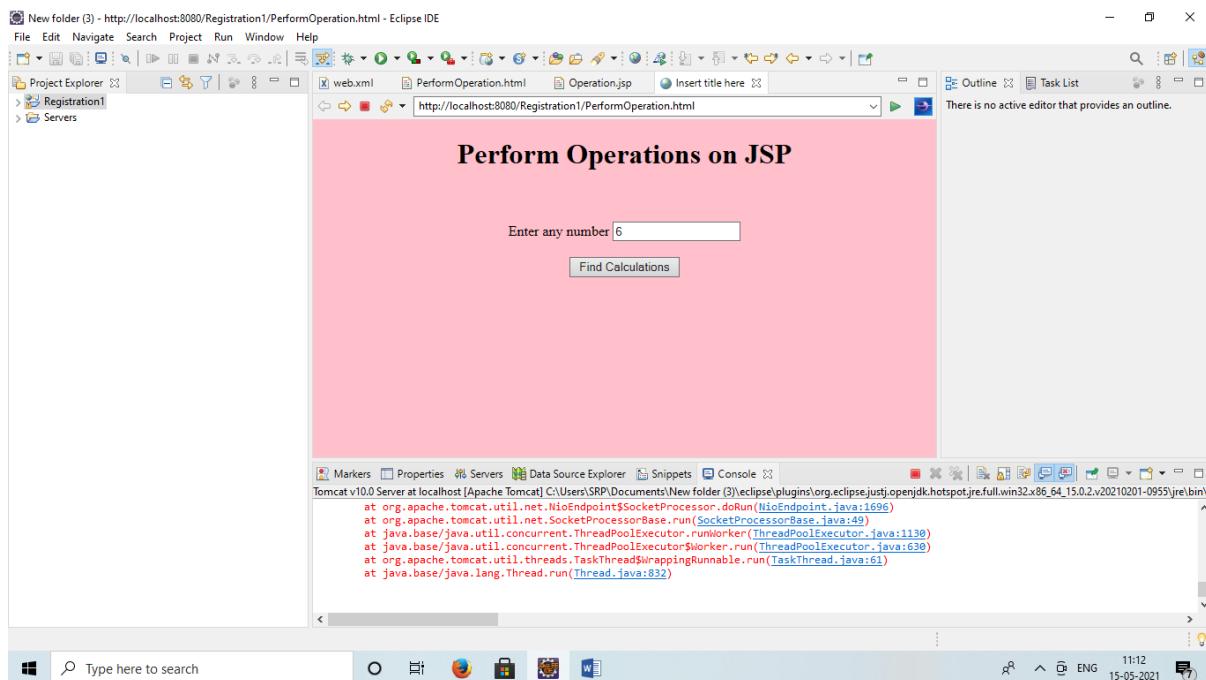
## Operation.jsp

```

<%@ page import = "java.io.* , java.util.*" language="java"
contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<html> <head>
<meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-1"> <title>Operations </title> </head>
<body>
<%
int number=Integer.parseInt(request.getParameter("num"));
if(number%2==0)
{
    out.println("Even Number");
}
else
{
    out.println("Odd Number");
}

```

```
%>
<br><br>
<%
int flag=0;
for(int i=2;i<=(number-1);i++)
{
    if(number%i==0)
    {
        flag=1;
        break;
    }
}
if(flag==0)
{
    out.println("Prime Number");
}
else
{
    out.println("Not a Prime Number");
}
%>
<br><br>
<%
int sq=number*number;
out.println("Square of a number: "+sq);
%>
<br><br>
<%
    Date date = new Date();
    out.println("Current Date: " +date.toString());
%>
```



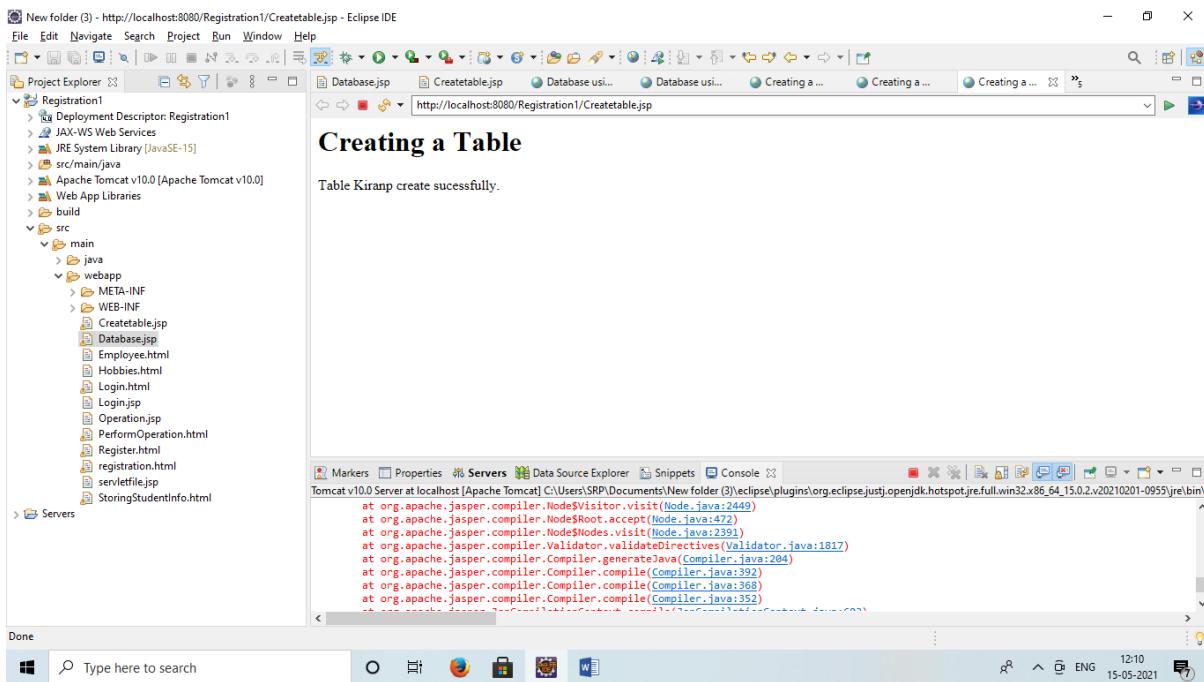
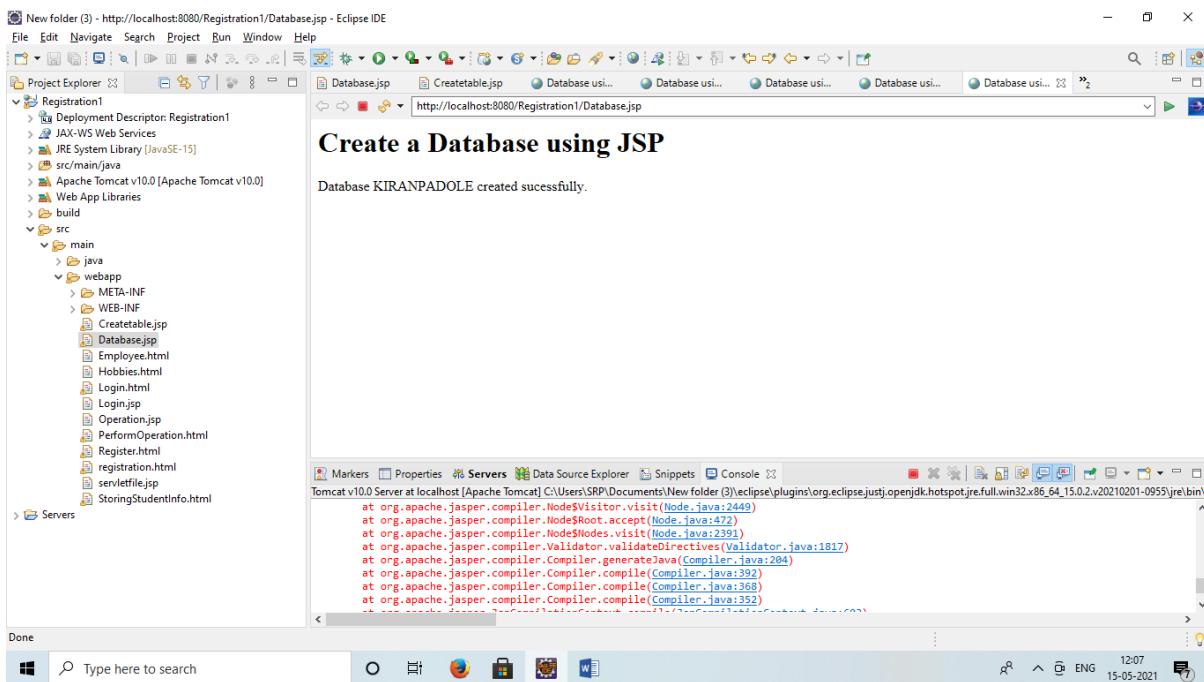
7. Write a JSP program to create database and table.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.*" %>
<html><head>
<title>Database using JSP</title></head>
<body><h1>Create a Database using JSP</h1>
<%
Connection connection = null;
try {
Class.forName("com.mysql.jdbc.Driver").newInstance();
connection = DriverManager.getConnection("jdbc:mysql://localhost:3306", "root", "root");
Statement statement = connection.createStatement();
String query = "CREATE DATABASE ";
statement.executeUpdate(query);
out.println("Database mca created sucessfully.");
}
catch (Exception e)
{
out.println("An error occurred.");
}
%>
</body></html>
```

## CreateTable.jsp

```
<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.*" %>
<html><head>
<title>Creating a Table</title></head>
```

```
<body><h1>Creating a Table</h1>
<%
Connection connection = null;
try {
Class.forName("com.mysql.jdbc.Driver").newInstance();
connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/KIR
ANP", "root", "root");
Statement statement = connection.createStatement();
String query = "CREATE TABLE Kiranp (Id int(20),FName
char(50), LName char(20));";
statement.executeUpdate(query);
out.println("Table Kiranp create sucessfully.");
}
catch (Exception e)
{
out.println("An error occurred.");
}
%>
</body></html>
```



8. Write a JSP application for accepting employee details like: First name, last name, city name and email id and save it in database. Perform following operations:

- a. Insert record in database
- b. Retrieve all records in database
- c. Update a record from database
- d. Delete a record from database

## HTML page

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1> Enter Employee Details </h1>
<form method="post" action="Insertdata.jsp">
Enter First name <input type="text"
name="first_name"><br><br>
```

```

Enter Last name <input type="text" name="Last_name"><br><br>
Enter City <input type="text" name="city_name"><br><br>
Enter Email Id <input type="email" name="email"><br><br><br>
<input type="submit" value="Proceed">

</body>
</html>

```

## Insertdata.jsp

```

<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.* , java.util.*"%>
<html>
<body>
<%
String first_name=request.getParameter("first_name");
String last_name=request.getParameter("last_name");
String city_name=request.getParameter("city_name");
String email=request.getParameter("email");
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fym
ca", "root", "root");
Statement st=conn.createStatement();
int i=st.executeUpdate("insert into
emp(first_name, last_name, city_name, email)values('"+first_na
me+"', '"
+last_name+"', '"+city_name+"', '"+email+"')");
out.println("Data is successfully inserted!");
}
catch(Exception e)
{
System.out.print(e);
e.printStackTrace();
}

```

```
}

%>
<h5><a href="RetriveData.jsp">Click here to read the
data</a></h5>
</body>
</html>
```

## Retrievedata.jsp

```
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%
String id = request.getParameter("userid");
String driver = "com.mysql.jdbc.Driver";
String connectionUrl =
"jdbc:mysql://localhost:3306/";
String database = "fymca";
String userid = "root";
String password = "root";
try {
Class.forName(driver);
} catch (ClassNotFoundException e) {
e.printStackTrace();
}
Connection connection = null;
Statement statement = null;
ResultSet resultSet = null;
%>
<!DOCTYPE html>
<html>
<body>
<h1>Retrieve data from database in jsp</h1>
```

```

<table border="1">
<tr>
<td>first name</td>
<td>last name</td>
<td>City name</td>
<td>Email</td>
</tr>
<%
try{
connection =
DriverManager.getConnection(connectionUrl+database, userid,
password);
statement=connection.createStatement();
String sql ="select * from emp";
resultSet = statement.executeQuery(sql);
while(resultSet.next()){
%>
<tr>
<td><%=resultSet.getString("first_name") %></td>
<td><%=resultSet.getString("last_name") %></td>
<td><%=resultSet.getString("city_name") %></td>
<td><%=resultSet.getString("email") %></td>
</tr>
<%
}
connection.close();
} catch (Exception e) {
e.printStackTrace();
}
%>
<h5><a href="UpdateData.jsp">Click here to update the
data</a></h5>
</table></body></html>

```

## Update.jsp

```

<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>

```

```

<%@page import="java.sql.Connection"%>
<%
String id = request.getParameter("id");
String driver = "com.mysql.jdbc.Driver";
String connectionUrl = "jdbc:mysql://localhost:3306/";
String database = "fymca";
String userid = "root";
String password = "root";
try {
Class.forName(driver);
} catch (ClassNotFoundException e) {
e.printStackTrace();
}
Connection connection = null;
Statement statement = null;
ResultSet resultSet = null;
%>
<%
try{
connection = DriverManager.getConnection(connectionUrl+database, userid,
password);
statement=connection.createStatement();
String sql ="select * from emp where id="+id;
resultSet = statement.executeQuery(sql);
while(resultSet.next()){
%>
```

```

<!DOCTYPE html>
<html>
<body>
<h1>Update data from database in jsp</h1>
<form method="post" action="update-process.jsp">
<input type="hidden" name="id" value="<% =resultSet.getString("id") %>">
<input type="text" name="id" value="<% =resultSet.getString("id") %>">
<br>
First name:<br>
<input type="text" name="first_name" value="<% =resultSet.getString("first_name") %>">
<br>
Last name:<br>
<input type="text" name="last_name" value="<% =resultSet.getString("last_name") %>">
<br>
City name:<br>
```

```

<input type="text" name="city_name" value="<% =resultSet.getString("city_name") %>">
<br>
Email Id:<br>
<input type="email" name="email" value="<% =resultSet.getString("email") %>">
<br><br>
<input type="submit" value="submit">
</form>
<%
}
connection.close();
} catch (Exception e) {
e.printStackTrace();
}
%>
</body></html>

```

### UpdateProcess.jsp:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.*" %>
<%! String driverName = "com.mysql.jdbc.Driver";%>
<%!String url = "jdbc:mysql://localhost:3306/fymca";%>
<%!String user = "root";%>
<%!String psw = "root";%>
<%
String id = request.getParameter("id");
String first_name=request.getParameter("first_name");
String last_name=request.getParameter("last_name");
String city_name=request.getParameter("city_name");
String email=request.getParameter("email");
if(id != null)
{
Connection con = null;
PreparedStatement ps = null;
int personID = Integer.parseInt(id);
try
{
Class.forName(driverName);
con = DriverManager.getConnection(url,user,psw);
String sql="Update emp set id=?,first_name=?,last_name=?,city_name=?,email=?
where id="+id;

```

```

ps = con.prepareStatement(sql);
ps.setString(1,id);
ps.setString(2, first_name);
ps.setString(3, last_name);
ps.setString(4, city_name);
ps.setString(5, email);
int i = ps.executeUpdate();
if(i > 0)
{
    out.print("Record Updated Successfully");
}
else
{
    out.print("There is a problem in updating Record.");
}
}
catch(SQLException sql)
{
    request.setAttribute("error", sql);
    out.println(sql);
}
}
%>

```

## indexUpdate.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.*" %>
<%@ page import="java.io.*" %>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%
String id = request.getParameter("id");
String driver = "com.mysql.jdbc.Driver";
String connectionUrl = "jdbc:mysql://localhost:3306/";

```

```

String database = "fymca";
String userid = "root";
String password = "root";
try {
Class.forName(driver);
} catch (ClassNotFoundException e) {
e.printStackTrace();
}
Connection connection = null;
Statement statement = null;
ResultSet resultSet = null;
%>
<html>
<body>
<h1>Retrieve data from database in jsp</h1>
<table border="1">
<tr>
<td>id</td>
<td>first name</td>
<td>last name</td>
<td>City name</td>
<td>Email</td>
<td>update</td>
</tr>
<%
try{
connection = DriverManager.getConnection(connectionUrl+database, userid, password);
statement=connection.createStatement();
String sql ="select * from emp";
resultSet = statement.executeQuery(sql);
while(resultSet.next()){
%>
<tr>
<td><%=resultSet.getString("id") %></td>
<td><%=resultSet.getString("first_name") %></td>
<td><%=resultSet.getString("last_name") %></td>
<td><%=resultSet.getString("city_name") %></td>
<td><%=resultSet.getString("email") %></td>
<td><a href="update.jsp?id=<%=resultSet.getString("id")%>">update</a></td>
</tr>
<%
}
connection.close();
} catch (Exception e) {
e.printStackTrace();
}

```

```
%>
</table></body></html>
```

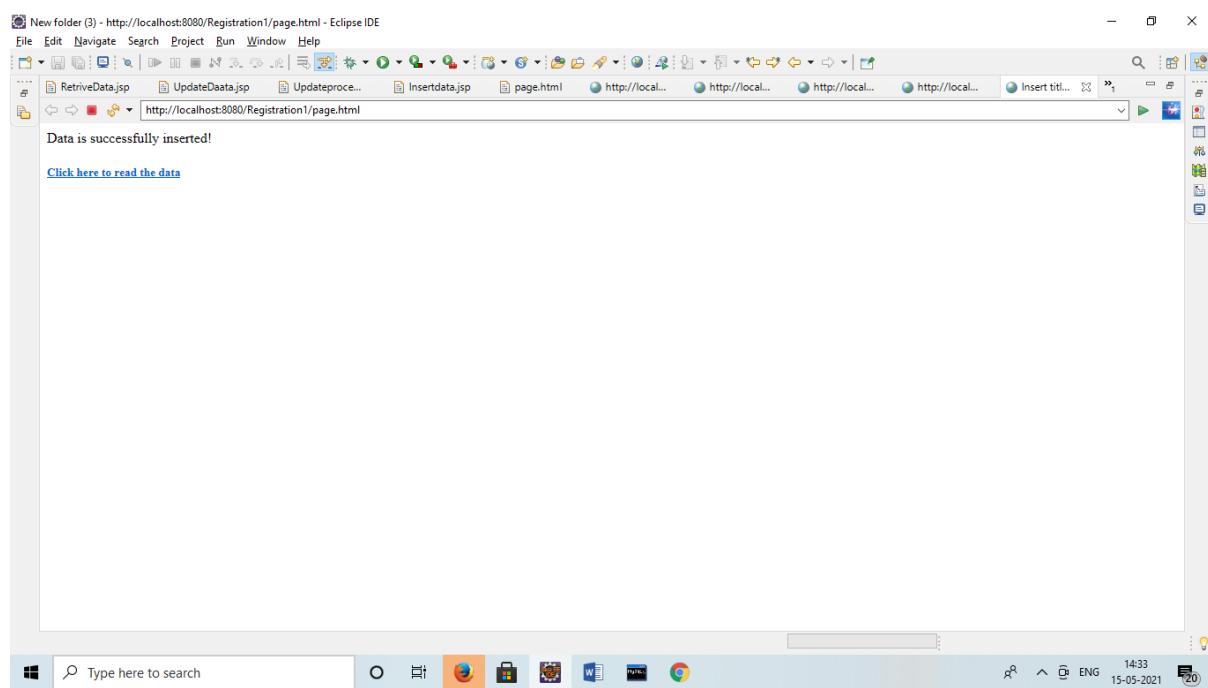
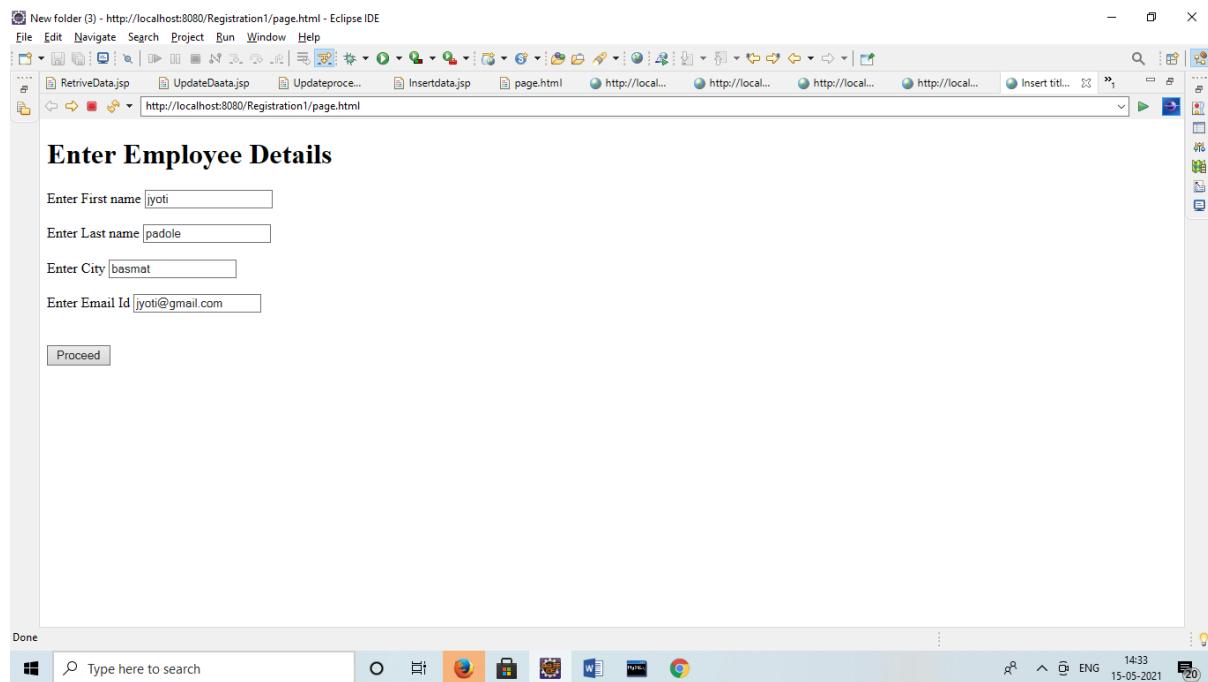
## indexDelete.jsp

```
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%
String driver = "com.mysql.jdbc.Driver";
String connectionUrl = "jdbc:mysql://localhost:3306/";
String database = "fymca";
String userid = "root";
String password = "root";
try {
Class.forName(driver);
} catch (ClassNotFoundException e) {
e.printStackTrace();
}
Connection connection = null;
Statement statement = null;
ResultSet resultSet = null;
%>
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<body>
<h1>Retrieve data from database in jsp</h1>
<table border="1">
<tr>
<td>first name</td>
<td>last name</td>
<td>City name</td>
<td>Email</td>
<td>Action</td>
```

```
</tr>
<%
try{
connection = DriverManager.getConnection(connectionUrl+database, userid,
password);
statement=connection.createStatement();
String sql ="select * from emp";
resultSet = statement.executeQuery(sql);
int i=0;
while(resultSet.next()){
%>
<tr>
<td><%=resultSet.getString("first_name") %></td>
<td><%=resultSet.getString("last_name") %></td>
<td><%=resultSet.getString("city_name") %></td>
<td><%=resultSet.getString("email") %></td>
<td><a href="delete.jsp?id=<%=resultSet.getString("id") %>">
<button type="button" class="delete">Delete</button></a></td>
</tr>
<%
i++;
}
connection.close();
} catch (Exception e) {
e.printStackTrace();
}
%>
</table>
</body>
</html>
```

## Delete.html

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.* , java.util.*" %>
<%
String id=request.getParameter("id");
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/fymca", "root",
"root");
Statement st=conn.createStatement();
int i=st.executeUpdate("DELETE FROM emp WHERE id="+id);
out.println("Data Deleted Successfully!");
}
catch(Exception e)
{
System.out.print(e);
e.printStackTrace();
}
%>
```



New folder (3) - http://localhost:8080/Registration1/page.html - Eclipse IDE

File Edit Navigate Search Project Run Window Help

RetriveData.jsp UpdateDaata.jsp Updateproce... Insertdata.jsp page.html http://local... http://local... http://local... http://local... http://local... 1

http://localhost:8080/Registration1/RetriveData.jsp

## Retrieve data from database in jsp

[Click here to update the data](#)

first name	last name	City name	Email
null	null	null	null
null	null	null	null
kiran	padole	basmat	padole@gmail.com
Ram	padole	basmat	dss@gmail.com
jyoti	padole	basmat	jyoti@gmail.com
jyoti	padole	basmat	jyoti@gmail.com

Done

Type here to search

14:33 15-05-2021

JavaServlet - http://localhost:8080/ExamplesServletJSP/IndexUpdate.jsp - Eclipse IDE

File Edit Navigate Search Project Run Window Help

http://localhost:8080/ExamplesServletJSP/update-process.jsp

http://localhost:8080/ExamplesServletJSP/update-process.jsp

Record Updated Successfully

Search for anything

17:44 14-05-2021

## DATA STRUCTURE AND ALGORITHMS:

### 1. WAP for implementation of Linear Search

```
<html>
<body>
<script>

var arr=new Array();
    var size=5;

    for(i=0;i<size;i++)
    {
        arr[i]=prompt("Enter Elements of Array:"+ (i+1));
    }
    document.write(arr);

    var n=prompt("Enter Number to search :");
    document.write("<br>");
    document.write("Number To be search :" +n);
    document.write("<br>");

    for(i=0;i<arr.length;i++)
    {
        if(n==arr[i])
        {
```

```

        document.write("Found the Number at Index :" + i);

    }

}

</script>

</body>

</html>

```



## 2. WAP for implementation of Binary Search

```
<html>
```

```
<script>
```

```
function binary()
{
    var n=parseInt(prompt("enter the size of an array: "));

    var a=new Array(n);

    var p=0;

    for(var i=0;i<a.length;i++)

    {
        a[i]=parseInt(prompt("enter array elements"));

    }

    for(var i=0;i<a.length;i++)

    {
        for(var j=i+1;j<a.length;j++)

        {
            if(a[i]>a[j])

            {
                var t=a[i];

                a[i]=a[j];

                a[j]=t;
            }
        }
    }

    var k=parseInt(prompt("enter the key element "));

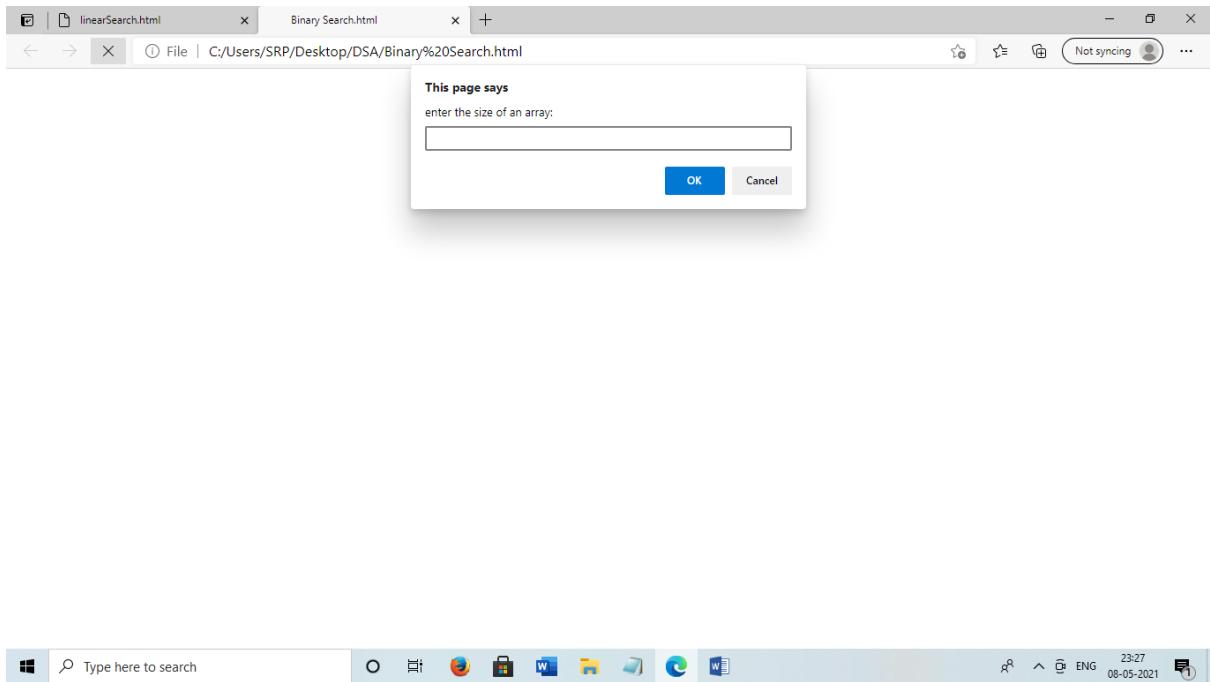
    var i=0;

    var u=parseInt(a.length-1);

    while(i<=u)
```

```
{  
    var m=parseInt((i+u)/2);  
    if(k==a[m])  
    {  
        p=1;  
        break;  
    }  
    else if(k>a[m])  
    {  
        i=m+1;  
    }  
    else if(k<a[m])  
    {  
        u=m-1;  
    }  
}  
if(p==1)  
    document.writeln("element found at :" + m);  
else  
    document.writeln("element not found");  
  
}  
</script>  
<body onLoad="binary()">  
</body>
```

```
</html>
```



### 3. WAP for implementation of Stack using array

```
<html>
```

```
<head></head>

<body>

<h1>Implementation of Stack using Array</h1>

<div class="result"></div>

<br />

<input type="text" class="stackPush" />

<button class="pushBtn">Push</button>

<button class="popBtn">Pop</button>
```

```
<button class="Btn">Display</button>

<script>

let resEle = document.querySelector(".result");

let BtnEle = document.querySelector(".Btn");

let pushBtnEle = document.querySelector(".pushBtn");

let popBtnEle = document.querySelector(".popBtn");

class Stack {

    constructor() {

        this.items = [];

        this.top = -1;

        this.length=0;

    }

}

Stack.prototype.push = function (ele) {

    if(this.length<=4){

        this.items[this.top] = ele;

        this.top = this.top+1;

        this.length=this.length+1;

        return ele +"is pushed to the stack at location"+ this.length + "and index is"+this.top;

    }

    else

    {

        return "Overflow condition";

    }

}
```

```

    }

};

Stack.prototype.pop = function () {
    if (this.top == -1) {
        return "Underflow: no more elements to delete";
    }
    else{
        let tempNum = this.items[this.top];
        this.top = this.top-1;
        this.length=this.length-1;
        return "Popped element";
    }
};

Stack.prototype.display = function () {
    if (this.top == -1) {
        return "Stack is empty";
    }
    for (let i = -1; i < this.top; i++) {
        resEle.innerHTML += this.items[i]+ "";
    }
};

let stack1 = new Stack();

BtnEle.addEventListener("click", () => {
    resEle.innerHTML = "";
    stack1.display();
}

```

```

});
```

```

pushBtnEle.addEventListener("click", () => {
    let ele = document.querySelector(".stackPush").value;
    resEle.innerHTML = stack1.push(ele);
});
```

```

popBtnEle.addEventListener("click", () => {
    resEle.innerHTML = stack1.pop();
});
```

```
</script>
```

```
</body>
```

```
</html>
```



## Implementation of Stack using Array

22 is pushed to the stack at location 2 and index is 1

Push Pop Display



## 4. WAP for implementation of Queue using array

```

<html>
<body>
<h1>Implementation of queue using Array</h1>
<div class="result"></div>
<br />
<input type="text" class="enqueueVal" /><button class="enqueueBtn">
Enqueue
</button>
<button class="dequeueBtn">Dequeue</button>
<button class="Btn">Display</button>
<script>
    let resEle = document.querySelector(".result");
    let BtnEle = document.querySelector(".Btn");
    let enqueueBtnEle = document.querySelector(".enqueueBtn");
    let dequeueBtnEle = document.querySelector(".dequeueBtn");
    class Queue {
        constructor() {
            this.items = [];
            this.length = 0;
            this.rear=-1;
            this.front=-1;
        }
    }
    Queue.prototype.enqueue = function (ele) {
        if(this.length<=4)
        {
            if(this.rear== -1)
            {
                this.front=0;
                this.rear=0;
                this.items[this.rear] = ele;
                this.length += 1;
                return ele +"is enqueued to the queue at front= " +this.front+ " and rear= "+this.rear;
            }
            else{
                this.rear=this.rear+1;
                this.items[this.rear] = ele;
            }
        }
    }

```

```

this.length += 1;

    return ele + "is enqueue to the queue at front= " +this.front+ " and rear=
"+this.rear;
}
}
else
{
    return "Overflow condition";
}
};

Queue.prototype.dequeue = function () {

if (this.front===-1) {
    return "Underflow: no more elements to remove";
}
else if(this.front==this.rear)
{
    let tempNum = this.items[this.front];

    this.front=-1;
    this.rear=-1;
    this.length=0;
    this.items=[];
    return tempNum + " is the element Deleted and now front is " +this.front+ " and
rear is " + this.rear + " and length is "+this.length+ " and array is empty"+ this.items;
}

else{
    let tempNum = this.items[this.front];
    this.front=this.front+1;
    return tempNum + " is the element Deleted and now front is " +this.front+ " and
rear is " + this.rear;
}

};

Queue.prototype.display = function () {

if (this.front===-1) {

```

```

        return "Queue is empty";
    }
    for (let i = this.front; i < this.length; i++) {
        resEle.innerHTML += this.items[i] + ", ";
    }
};

let queue1 = new Queue();
BtnEle.addEventListener("click", () => {
    resEle.innerHTML = "";
    queue1.display();
});

enqueueBtnEle.addEventListener("click", () => {
    let ele = document.querySelector(".enqueueVal").value;
    resEle.innerHTML = queue1.enqueue(ele);
});

dequeueBtnEle.addEventListener("click", () => {
    resEle.innerHTML = queue1.dequeue();
});
</script>
</body>
</html>

```



## Implementation of queue using Array

23 is enqueued to the queue at front= 0 and rear= 1

23	<input type="button" value="Enqueue"/>	<input type="button" value="Dequeue"/>	<input type="button" value="Display"/>
----	--	--	--



## 5. WAP for implementation of Circular Queue using Array

```
<html>
<body>
<h1>Implementation of Circular Queue using Array</h1>
<div class="result"></div>
<br/>
<input type="text" class="enqueueVal" />
<button class="enqueueBtn">Enqueue</button>
<button class="dequeueBtn">Dequeue</button>
<button class="frontBtn">Front</button>
<button class="rearBtn">Rear</button>
<button class="Btn">Display</button>

<script>
let resEle=document.querySelector(".result");
let BtnEle=document.querySelector(".Btn");
let enqueueBtnEle=document.querySelector(".enqueueBtn");
let dequeueBtnEle=document.querySelector(".dequeueBtn");
let rearBtnEle=document.querySelector(".rearBtn");
let frontBtnEle=document.querySelector(".frontBtn");
class Queue{
    constructor(){
        this.size = 5;
        this.items= new Array(this.size);
        this.rear=-1;
        this.front=-1;
    }
}
Queue.prototype.enqueue=function(ele){
    if((this.front == -1)&&(this.rear == -1)){
        this.front = 0;
        this.rear = 0;
        this.items[this.rear] = ele;
        return ele + " is the element added. Now front is " +this.front+",";
        rear is " + this.rear;
    }else if((this.rear+1)%this.size == this.front){
        return "Overflow Condition.";
    }
}
```

```

}else{
    this.rear = (this.rear+1)%this.size;
    this.items[this.rear] = ele;
    return ele + " is the element added. Now front is " +this.front+",";
rear is " + this.rear;
}
}

Queue.prototype.dequeue=function(){
if((this.front == -1)&&(this.rear == -1)){
    return "Underflow Condition.";
}else if(this.front == this.rear){
    let temp = this.items[this.front];
    this.front = -1;
    this.rear = -1;
    return temp + " is the element deleted. Now front is "
+this.front+, rear is " + this.rear;
}else{
    let temp = this.items[this.front];
    this.front = (this.front+1)%this.size;
    return temp + " is the element deleted. Now front is "
+this.front+, rear is " + this.rear;
}
}

Queue.prototype.rearfun=function(){
    return "Rear : "+this.rear;
}

Queue.prototype.frontfun=function(){
    return "Front : "+this.front;
}

Queue.prototype.display=function(){
    if(this.front ===-1 && this.rear===-1){
        return "Queue is empty."
}else{
    for(let i = 0; i<this.size; i++){
        if(this.items[i] !== undefined){
            resEle.innerHTML +=this.items[i]+ " ";
        }
    }
}
}

```

```
}

let queue1=new Queue();
BtnEle.addEventListener("click", () => {
    resEle.innerHTML="";
    queue1.display();
});
enqueueBtnEle.addEventListener("click", () => {
    let ele=document.querySelector(".enqueueVal").value;
    resEle.innerHTML=queue1.enqueue(ele);
});
dequeueBtnEle.addEventListener("click", () => {
    resEle.innerHTML=queue1.dequeue();
});
rearBtnEle.addEventListener("click", () => {
    let ele=document.querySelector(".enqueueVal").value;
    resEle.innerHTML=queue1.rearfun();
});
frontBtnEle.addEventListener("click", () => {
    let ele=document.querySelector(".enqueueVal").value;
    resEle.innerHTML=queue1.frontfun();
});
</script>
</body>
</html>
```



## Implementation of Circular Queue using Array

100 is the element added. Now front is 1, rear is 2

Enqueue Dequeue Front Rear Display



## 6. WAP for implementation of Priority Queue using Array

```
<html>

</script>
class QElement {
    constructor(element, priority)
    {
        this.element = element;
        this.priority = priority;
    }
}
class PriorityQueue {
    constructor()
    {
        this.items = [];
    }
}
```

```

        }

enqueue(element, priority)
{
    var qElement = new QElement(element, priority);
    var contain = false;
    for (var i = 0; i < this.items.length; i++) {
        if (this.items[i].priority > qElement.priority) {
            this.items.splice(i, 0, qElement);
            contain = true;
            break;
        }
    }
    if (!contain) {
        this.items.push(qElement);
    }
}

dequeue()
{
    if (this.isEmpty())
        return "Underflow";
    return this.items.shift();
}

{
    if (this.isEmpty())
        return "No elements in Queue";
    return this.items[0];
}

rear()
{
    if (this.isEmpty())
        return "No elements in Queue";
    return this.items[this.items.length - 1];
}

isEmpty()
{
    // return true if the queue is empty.
    return this.items.length == 0;
}

// printPQueue()
printPQueue()

```

```

{
    var str = "";
    for (var i = 0; i < this.items.length; i++)
        str += this.items[i].element + " ";
    return str;
}
}

document.write(priorityQueue.isEmpty()+"<br/>");
document.write(priorityQueue.front()+"<br/>");
priorityQueue.enqueue("Sumit", 2);
priorityQueue.enqueue("Gourav", 1);
priorityQueue.enqueue("Piyush", 1);
priorityQueue.enqueue("Sunny", 2);
priorityQueue.enqueue("Sheru", 3);

// prints [Gourav Piyush Sumit Sunny Sheru]
document.write(priorityQueue.printPQueue()+"<br/>");

// prints Gourav
document.write(priorityQueue.front().element+"<br/>");

// prints Sheru
document.write(priorityQueue.rear().element+"<br/>");

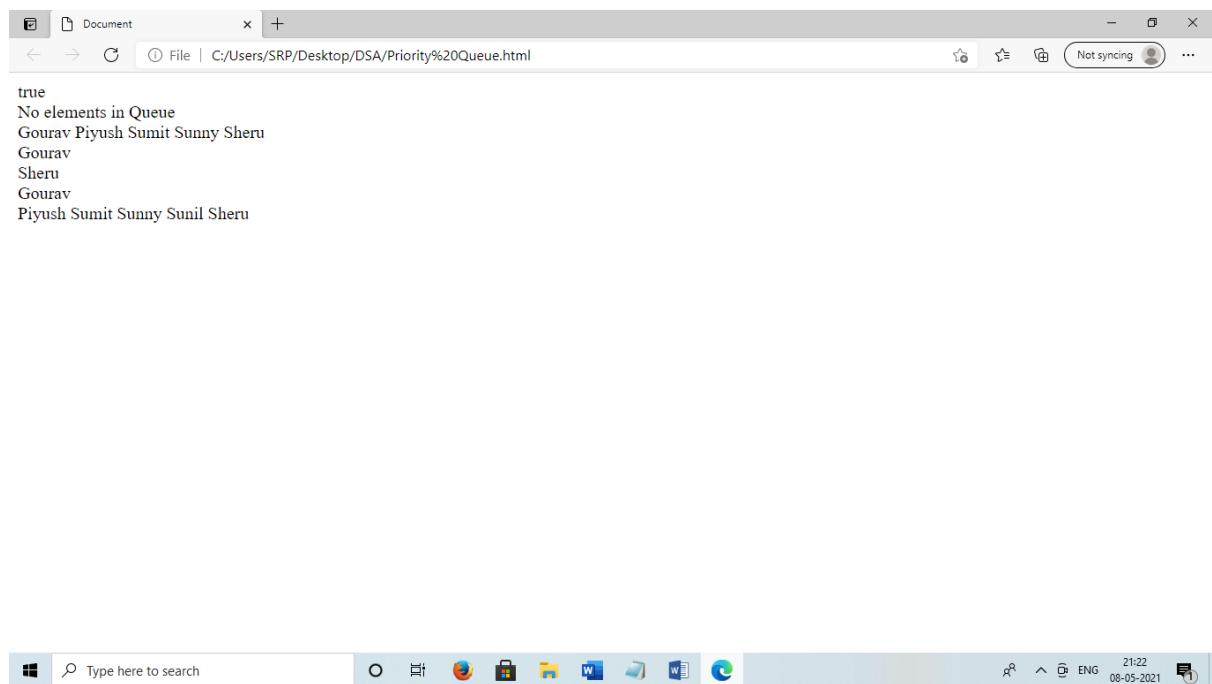
// removes Gouurav
// priorityQueue contains
// [Piyush Sumit Sunny Sheru]
document.write(priorityQueue.dequeue().element+"<br/>");

// Adding another element to the queue
priorityQueue.enqueue("Sunil", 2);

// prints [Piyush Sumit Sunny Sunil Sheru]
document.write(priorityQueue.printPQueue()+"<br/>");

</script>
</body>
</html>

```



## 1. WAP for implementation of Stack to Reverse a String

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Document</title>

<body>

<script>

class Queue {
```

```

this.elements1 = [];

    this.front=-1;

    this.rear=-1;

}

enqueue(element){

    if(this.elements1.length<=4)

    {

        if(this.isEmpty())

        {

            this.front=0;

            this.rear=0;

            this.elements1[this.rear]=element;

            //document.write("Front:" +

this.elements1[this.front]+"Rear:"+this.elements1[this.rear]);

        }

        else if(this.front>-1)

        {

            this.rear=this.rear+1;

            this.elements1[this.rear]=element;

            //document.write("Front:" +

this.elements1[this.front]+"Rear:"+this.elements1[this.rear]);

        }

    }

}

}else

{

```

```

        return 'Overflow situation';

    }

}

dequeue() {
    let tempNum1;

    if(!this.isEmpty()) {
        if(this.front==this.rear)

        {

            tempNum1 = this.elements1[this.front];

            this.front=-1;

            this.rear=-1;

            this.length=0;

            this.elements1=[];

        }

        else{

            tempNum1 = this.elements1[this.front];

            this.front=this.front+1;

        }

        return tempNum1;

    }

    else {

        return 'Underflow situation';

    }
}

```

```
}

isEmpty() {
    return this.elements1.length == 0;
}

print(){
    return this.elements1;
}

}

class Stack {
    constructor(){
        this.elements = [];
        this.top=-1;
    }

    push(element){
        if(this.elements.length<=4)
        {
            if(this.isEmpty())
            {
                this.top=0;
                this.elements[this.top]=element;
                //document.write("Front:" +
                this.elements1[this.front]+"Rear:"+this.elements1[this.rear]);
            }
        }
    }
}
```

```

        }

        else if(this.top>-1)

        {

            this.top=this.top+1;

            this.elements[this.top]=element;

            //document.write("Front:" +
            this.elements1[this.front]+"Rear:"+this.elements1[this.rear]);

        }

    }

}

return 'Overflow situation';

}

}

pop(){

let tempNum;

if(!this.isEmpty()) {

    if(this.top==0)

    {

        tempNum = this.elements[this.top];

        this.top=-1;

        this.elements=[];

    }

    else{


```

```

        tempNum = this.elements[this.top];
        this.top=this.top-1;
    }

    return tempNum;
}

else {
    return 'Underflow situation';
}

}

isEmpty(){
    return this.elements.length == 0;
}

print(){
    return this.elements;
}

}

//Creates a new stack.

let stack = new Stack();

let queue = new Queue(); //Creating a new queue

function reverse(stack){

```

```
while(!stack.isEmpty()){ // Looping until the stack is empty.

    queue.enqueue(stack.pop()); //Enqueue the popped stack element.

}

document.write("Printing queue before reversal: "+ queue.print()+"<br/>");

while(!queue.isEmpty()){ //Looping until the queue is empty.

    stack.push(queue.dequeue()); //Pushed the dequeued queue element.

}

}

//Pushes 3 strings onto the stack

stack.push(10);

stack.push(20);

stack.push(30);

stack.push(40);

//Prints the stack before the reversal.

document.write("Printing stack before reversal: "+ stack.print()+"<br/>");



//Reverses the stack.

reverse(stack);




//Prints the stack after the reversal.
```

```
document.write("Printing stack after reversal: "+ stack.print());  
</script>  
</body>  
</html>
```



## 1. WAP to check balanced parentheses using Stacks

```
<!DOCTYPE html>  
<html lang="en">  
<head>
```

```
<title>Document</title>

<script>

let isMatchingBrackets = function (str) {

    let stack = [];

    let map = {
        '(': ')',
        '[': ']',
        '{': '}'
    }

    for (let i = 0; i < str.length; i++) {

        // If character is an opening brace add it to a stack

        if (str[i] === '(' || str[i] === '{' || str[i] === '[') {

            stack.push(str[i]);

        }

        // If that character is a closing brace, pop from the stack, which will also reduce the length of the
        // stack each time a closing bracket is encountered.

        else {

            let last = stack.pop();

            //If the popped element from the stack, which is the last opening brace doesn't match the
            //corresponding closing brace in the map, then return false

            if (str[i] !== map[last]) {return false};

        }

    }

}
```

```

}

// By the completion of the for loop after checking all the brackets of the str, at the end, if the stack is
not empty then fail

if (stack.length !== 0) {return false};

return true;

}

document.write(isMatchingBrackets("(){}") + "<br/>"); // returns true

document.write(isMatchingBrackets("[{{()}}({[]})]{{[{}]}((((([]){}))[]{{{({{{{{}}}})}}))}))[][" + "<br
/>"); // returns true

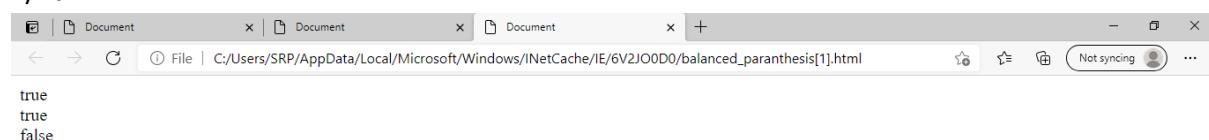
document.write(isMatchingBrackets("{{()}}}})); // returns false

</script>

</body>

</html>

```



## 9. WAP to reverse stack using queue

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Document</title>

<script>

class Queue {

    constructor(){

        this.elements1 = [];

        this.front=-1;

        this.rear=-1;

    }

}
```

```
enqueue(element){  
    if(this.elements1.length<=4)  
    {  
        if(this.isEmpty())  
        {  
            this.front=0;  
            this.rear=0;  
            this.elements1[this.rear]=element;  
            //document.write("Front:" +  
this.elements1[this.front]+ "Rear:" +this.elements1[this.rear]);  
        }  
        else if(this.front>-1)  
        {  
            this.rear=this.rear+1;  
            this.elements1[this.rear]=element;  
            //document.write("Front:" +  
this.elements1[this.front]+ "Rear:" +this.elements1[this.rear]);  
        }  
    }  
}
```

```
    }else

    {
        return 'Overflow situation';
    }

}

dequeue() {
    let tempNum1;

    if(!this.isEmpty()) {

        if(this.front==this.rear)

        {
            tempNum1 = this.elements1[this.front];

            this.front=-1;

            this.rear=-1;

            this.length=0;

            this.elements1=[];

        }
    }
}
```

```
        else{  
            tempNum1 = this.elements1[this.front];  
            this.front=this.front+1;  
        }  
        return tempNum1;  
    }  
    else {  
        return 'Underflow situation';  
    }  
}  
  
isEmpty() {  
    return this.elements1.length == 0;  
}  
  
print(){  
    return this.elements1;  
}
```

```
}
```

```
class Stack {
```

```
    constructor(){
```

```
        this.elements = [];
```

```
        this.top=-1;
```

```
}
```

```
    push(element){
```

```
        if(this.elements.length<=4)
```

```
{
```

```
        if(this.isEmpty())
```

```
{
```

```
            this.top=0;
```

```
            this.elements[this.top]=element;
```

```
            //document.write("Front:" +
```

```
this.elements1[this.front]+"Rear:"+this.elements1[this.rear]);
```

```
}
```

```
        else if(this.top>-1)
```

```
    {  
  
        this.top=this.top+1;  
  
        this.elements[this.top]=element;  
  
        //document.write("Front:" +  
this.elements1[this.front]+"Rear:"+this.elements1[this.rear]);  
  
    }  
  
}
```

```
}else  
{  
  
    return 'Overflow situation';  
  
}  
  
}
```

```
pop(){  
  
let tempNum;  
  
if(!this.isEmpty()) {  
  
    if(this.top==0)  
  
    {  
  
        tempNum = this.elements[this.top];
```

```
        this.top=-1;  
  
        this.elements=[];  
  
    }  
  
    else{  
  
        tempNum = this.elements[this.top];  
  
        this.top=this.top-1;  
  
    }  
  
    return tempNum;  
  
}  
  
else {  
  
    return 'Underflow situation';  
  
}  
  
}  
  
  
  
isEmpty(){  
  
    return this.elements.length == 0;  
  
}
```

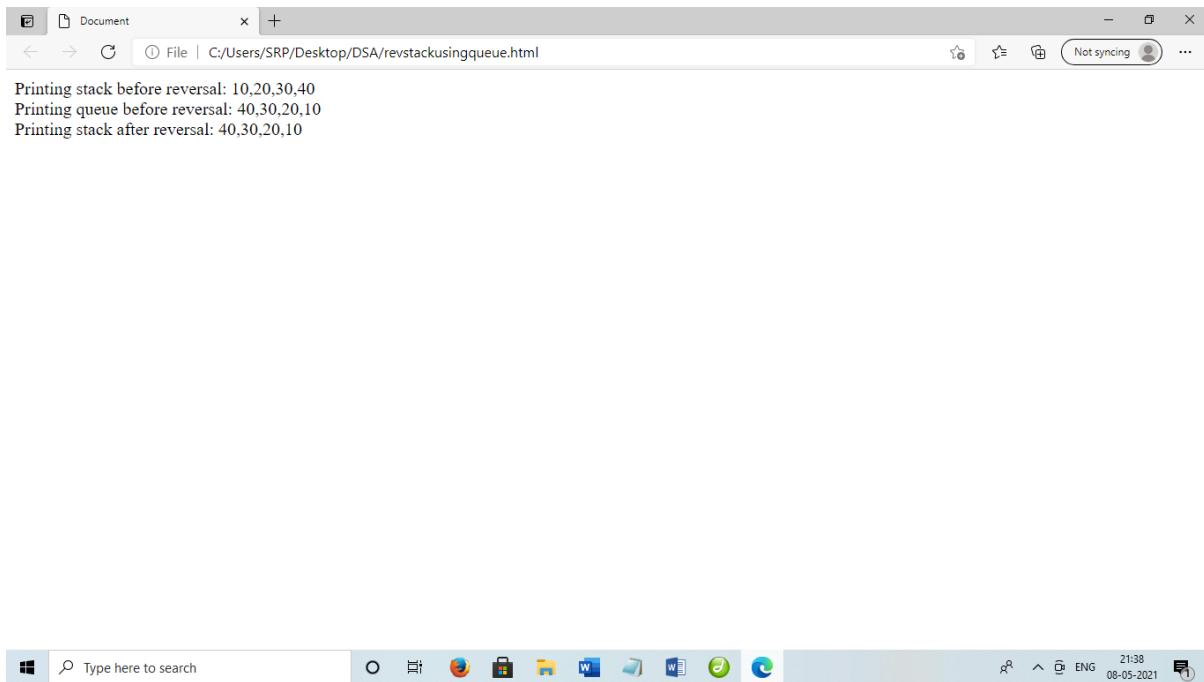
```
print(){  
    return this.elements;  
}  
  
}  
  
//Creates a new stack.  
let stack = new Stack();  
  
let queue = new Queue(); //Creating a new queue  
function reverse(stack){  
  
    while(!stack.isEmpty()){ // Looping until the stack is empty.  
        queue.enqueue(stack.pop()); //Enqueue the popped stack  
        element.  
    }  
}
```

```
document.write("Printing queue before reversal: "+  
queue.print()+"<br/>");  
  
while(!queue.isEmpty()){ //Looping until the queue is empty.  
    stack.push(queue.dequeue()); //Pushed the dequeued  
queue element.  
}  
}  
  
//Pushes 3 strings onto the stack
```

```
stack.push(10);  
stack.push(20);  
stack.push(30);  
stack.push(40);  
  
//Prints the stack before the reversal.
```

```
document.write("Printing stack before reversal: "+  
stack.print()+"<br/>");
```

```
//Reverses the stack.  
  
reverse(stack);  
  
  
  
//Prints the stack after the reversal.  
  
document.write("Printing stack after reversal: "+ stack.print());  
  
</script>  
  
</body>  
  
</html>
```



## 1. WAP for implementation of Singly Linked List

```
<html>
<body>
<script>

class Linkedlist {

constructor() {
    this.head = null;
    this.tail=null;
    this.length=0;
}

addlast(element)
{
    var node=new this.Node(element);
    var current;

    if(this.head==null)
    {
        this.head=node;
        this.tail=node;
    }
    else
    {
        current=this.head;
        while(current.next!=null)
            current=current.next;
        current.next=node;
        node.prev=current;
        this.tail=node;
    }
}
gethead()
{
    return this.head;
}
gettail()
{
    return this.tail;
}
getlength()
{
    return this.length;
}
toString()
{
    var str="";
    var current=this.head;
    while(current!=null)
    {
        str+=current.value+",";
        current=current.next;
    }
    return str;
}
}
```

```

        this.length++;

        document.write(node.element+" element is added at location " +this.length+"head
is"+headval+"and tail is"+tailval+"<br/>");

    }

    else

    { current=this.head;

        while(current.next)

        {

            current=current.next;

        }

        current.next=node;

        this.tail=node;

        this.length++;

        document.write(node.element+" element is added at location " +this.length+
" that is after "+current.element+"head is"+this.head.element+"and tail
is"+this.tail.element+"<br/>");

    }

}

addfirst(element)

{

var node=new this.Node(element);

var current;

```

```

if(this.head==null)

{
    this.head=node;

    this.tail=node;

    var headval=JSON.stringify(this.head);

    var tailval=JSON.stringify(this.tail);

    this.length++;

    document.write(node.element+" element is added and head is"+headval+"and tail
is"+tailval+"<br/>");

}

else

{
    node.next=this.head;

    this.head=node;

    current=this.head;

    while(current.next)

    {
        current=current.next;

    }

    this.tail=current;

    this.length++;

    document.write(node.element+" element is added and head
is"+this.head.element+"and tail is"+this.tail.element+"<br/>");

}

```

```
}
```

```
deletelast()
```

```
{
```

```
var current;
```

```
if(this.head==null)
```

```
{
```

```
document.write("No element to delete<br/>");
```

```
}
```

```
else if(this.head==this.tail)
```

```
{
```

```
document.write(this.head.element+" element is deleted <br/>");
```

```
this.head=null;
```

```
this.tail=null;
```

```
this.length--;
```

```
}
```

```
else
```

```
{ let prev=this.head;
```

```
current=this.head.next;
```

```
        while(current.next!=null)

        {   prev=current;

            current=current.next;

        }

        document.write(current.element+" element is deleted <br/>");

        prev.next=null;

        this.tail=prev;

        this.length--;

    }

}

deletefirst()

{

var current;

if(this.head==null)

{

    document.write("No element to delete<br/>");

}

else if(this.head==this.tail)

{
```

```
document.write(this.head.element+" element is deleted <br/>");

this.head=null;

this.tail=null;

this.length--;

}

else

{

var temp=this.head;

this.head=this.head.next;

current=this.head;

while(current.next)

{ current=current.next;

}

this.tail=current;

this.length--;

document.write(temp.element+" element is deleted and head is

"+this.head.element+" <br/>");

}

}
```

```
display()

{
    if(this.head==null)

    {
        document.write("No element to display<br/>");

    }

    else

    {
        var curr=this.head;

        var it=1;

        while(it<=this.length)

        {

            document.write(curr.element+"->");

            curr=curr.next;

            it++;

        }

        document.write("<br/>");

    }

}

}

}
```

```
Linkedlist.prototype.Node = class {
    constructor(element) {
        this.element = element; this.next = null;
    }
};
```

```
let list1 = new Linkedlist();
```

```
list1.addlast(35);
```

```
list1.addlast(44);
```

```
list1.display();
```

```
list1.addlast(56);
```

```
list1.display();
```

```
list1.deletelast();
```

```
list1.display();
```

```
list1.addfirst(23);
```

```
list1.display();
```

```
list1.deletefirst();
```

```
list1.display();
```

```
list1.deletelast();
```

```
list1.display();
```

```
list1.deletefirst();
```

```

list1.display();

list1.deletelast();

</script>

</body>

</html>

```

35 element is added at location 1 head is {"element":35,"next":null} and tail is {"element":35,"next":null}  
 44 element is added at location 2 that is after 35 head is 35 and tail is 44  
 35->44->  
 56 element is added at location 3 that is after 44 head is 35 and tail is 56  
 35->44->56->  
 56 element is deleted  
 35->44->  
 23 element is added and head is 23 and tail is 44  
 23->35->44->  
 23 element is deleted and head is 35  
 35->44->  
 44 element is deleted  
 35->  
 35 element is deleted  
 No element to display  
 No element to delete

## 11. WAP for implementation of Doubly Linked List

```

<!DOCTYPE html>

<html lang="en">

<head>

```

```
<title>Document</title>

<script>

class DLinkedlist {

    constructor() {
        this.head = null;
        this.tail=null;
        this.length=0;
    }

    addlast(element)
    {
        var node=new this.Node(element);
        var current;

        if(this.head==null)
        {
            this.head=node;
            this.tail=node;
            var headval=JSON.stringify(this.head);
            var tailval=JSON.stringify(this.tail);
            this.length++;
            document.write(node.element+" element is added at location " +this.length+"head
is"+headval+"and tail is"+tailval+"<br/>");
        }
    }
}
```

```

else
{
    current=this.head;
    while(current.next)
    {
        current=current.next;
    }

    current.next=node;
    node.prev=current;//add this statement
    this.tail=node;
    this.length++;

    document.write(node.element+" element is added at location " +this.length+
that is after "+current.element+"head is"+this.head.element+"and tail
is"+this.tail.element+"<br/>");

}
}

addfirst(element)
{
var node=new this.Node(element);
var current;

if(this.head==null)
{
    this.head=node;
}

```

```

        this.tail=node;

        var headval=JSON.stringify(this.head);

        var tailval=JSON.stringify(this.tail);

        this.length++;

        document.write(node.element+" element is added and head is"+headval+"and tail
is"+tailval+"<br/>");

    }

else

{

node.next=this.head;

    this.head.prev=node;//add this statement

    this.head=node;

current=this.head;

while(current.next)

{

    current=current.next;

}

this.tail=current;

    this.length++;

    document.write(node.element+" element is added and head
is"+this.head.element+"and tail is"+this.tail.element+"<br/>");

}

}

```

```
deletelast()

{
    var current;

    if(this.head==null)
    {
        document.write("No element to delete<br/>");

    }

else if(this.head==this.tail)
{
    document.write(this.head.element+" element is deleted <br/>");

    this.head=null;

    this.tail=null;

    this.length--;

}

else
{
    current=this.head;

    while(current.next!=null)
    {
        current=current.next;
    }

    let temp = current.element

    document.write(temp+" element is deleted <br/>");

    this.tail= current.prev;
}
```

```
this.tail.next=null;  
this.length--;  
}  
}
```

}

**deletefirst()**

{

```
var current;
```

```
if(this.head==null)
```

{

```
document.write("No element to delete<br/>");
```

}

```
else if(this.head==this.tail)
```

{

```
document.write(this.head.element+" element is deleted <br/>");
```

```
this.head=null;
```

```
this.tail=null;
```

this.length--;

}

else

```
{  
    var temp=this.head;  
  
    this.head=this.head.next;  
  
    this.head.prev=null;//add this statement  
  
    current=this.head;  
  
    while(current.next)  
    {    current=current.next;  
  
    }  
  
    this.tail=current;  
  
    this.length--;  
  
    document.write(temp.element+" element is deleted and head is  
"+this.head.element+" <br/>");
```

```
}
```

```
}
```

```
display()  
{  
    if(this.head==null)  
    {  
        document.write("No element to display<br/>");  
    }
```

```
    }

else

{

    var curr=this.head;

    var it=1;

    while(it<=this.length)

    {

        document.write("<-"+curr.element+"->");

        curr=curr.next;

        it++;

    }

    document.write("<br/>");

}

}

}

DLinkedlist.prototype.Node = class {

constructor(element) {

    this.element = element;

    this.next = null;

    this.prev = null;

}

};

};
```

```
let list1 = new DLinkedlist();
```

```
list1.addlast(35);
```

```
list1.addlast(44);
```

```
list1.addlast(50);
```

```
list1.addlast(46);
```

```
list1.display();
```

```
list1.deletefirst();
```

```
list1.deletelast();
```

```
list1.display();
```

```
list1.deletefirst();
```

```
list1.deletelast();
```

```
list1.display();
```

```
list1.deletelast();
```

```
</script>
```

```
</body>
```

```
</html>
```



## 12. WAP for implementation of Stack using Linked List.

```
<!DOCTYPE html>

<html lang="en">
<head>

<title>Document</title>

<script>
```

```
class Linkedlist {  
  
    constructor() {  
        this.head = null;  
        this.tail=null;  
        this.top=null;  
        this.length=0;  
    }  
  
    push(element)  
    {  
        var node=new this.Node(element);  
        var current;  
  
        if(this.head==null)  
        {  
            this.head=node;  
            this.tail=node;  
            this.top=node;  
            var headval=JSON.stringify(this.head);  
            var tailval=JSON.stringify(this.tail);  
            var topval=JSON.stringify(this.top);  
            this.length++;  
  
            document.write(node.element+" element is added at location " +this.length+"head  
is"+headval+"and tail is"+tailval+ "and top is"+topval+"<br/>");  
        }  
    }  
}
```

```

else
{
    current=this.head;
    while(current.next)
    {
        current=current.next;
    }

    current.next=node;
    this.tail=node;
    this.top=node;
    this.length++;

    document.write(node.element+" element is added at location " +this.length+
that is after "+current.element+"head is "+this.head.element+"and tail is"+this.tail.element+"and
top is"+this.top.element+"<br/>");

}
}

pop()
{
var current;
if(this.head==null)
{
    document.write("No element to delete<br/>");
}

```

```
}

else if(this.head==this.tail)

{

    document.write(this.head.element+" element is deleted <br/>");

    this.head=null;

    this.tail=null;

    this.top=null;

    this.length--;

}

else

{

    let prev=this.head;

    current=this.head.next;

    while(current.next!=null)

    {

        prev=current;

        current=current.next;

    }

    document.write(current.element+" element is deleted <br/>");

    prev.next=null;

    this.tail=prev;

    this.top=prev;

    this.length--;

}
```

```
}
```

```
display()
{
    if(this.head==null)
    {
        document.write("No element to display<br/>");
    }
    else
    {
        var curr=this.head;
        var it=1;
        while(it<=this.length)
        {
            document.write(curr.element+"->");
            curr=curr.next;
            it++;
        }
        document.write("<br/>");
    }
}
```

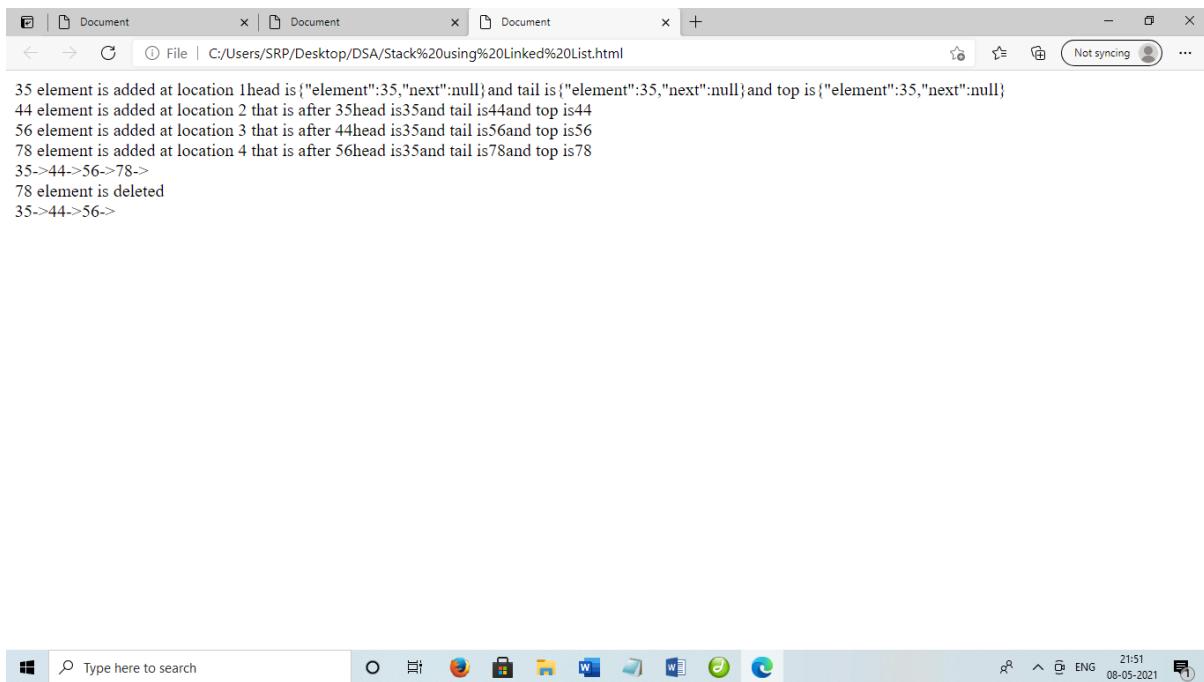
```
    }
}

}

Linkedlist.prototype.Node = class {
  constructor(element) {
    this.element = element; this.next = null;
  }
};

let list1 = new Linkedlist();
list1.push(35);
list1.push(44);
list1.push(56);
list1.push(78);
list1.display();
list1.pop();
list1.display();

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



### 13. WAP for implementation of Queue using Linked List.

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Queue End Term</title>

<script>

//Queue using linkedlist

function queueUsingLL()
```

```
let Node = function(elm){  
    this.element = elm;  
    this.next = null;  
}  
  
let length = 0;  
  
let head = null;  
  
this.enqueue = function(elm){  
    let node = new Node(elm),  
        current;  
  
    if(head === null){  
        head = node;  
    }else{  
        current = head;  
  
        while(current.next){  
            current = current.next;  
        }  
  
        current.next = node;  
    }  
  
    length++;  
}  
  
this.dequeue = function(){  
    let current = head;  
  
    if(current){  
        let elm = current.element;  
        current = current.next;  
    }  
}
```

```
    head = current;
    length--;
    return elm;
}
return null;
}

this.front = function(){
if(head){
    return head.element;
}
return null;
}

this.rear = function(){
let current = head;
if(current === null){
    return null;
}
while(current.next){
    current = current.next;
}
return current.element;
}

this.toArray = function(){
let arr = [];
let current = head;
```

```
while(current){  
    arr.push(current.element);  
    current = current.next;  
}  
  
return arr;  
}  
  
this.isEmpty = function(){  
    return length === 0;  
}  
  
this.size = function(){  
    return length;  
}  
  
this.clear = function(){  
    head = null;  
    length = 0;  
}  
  
}  
  
let queue = new queueUsingLL();  
  
document.write(queue.isEmpty()+"<br>");  
  
queue.enqueue('kiran');  
  
queue.enqueue('Ramkishan');  
  
queue.enqueue('Padole');  
  
document.write(queue.toArray()+"<br>");  
  
queue.dequeue('kiran');  
  
queue.dequeue('Ramkishan');
```

```

document.write(queue.toArray()+"<br>");

queue.enqueue('FYMCA');

queue.enqueue('B');

queue.dequeue();

document.write(queue.toArray()+"<br>")+"<br>";

document.write(queue.size()+"<br>")+"<br>";

document.write(queue.front()+"<br>")+"<br>";

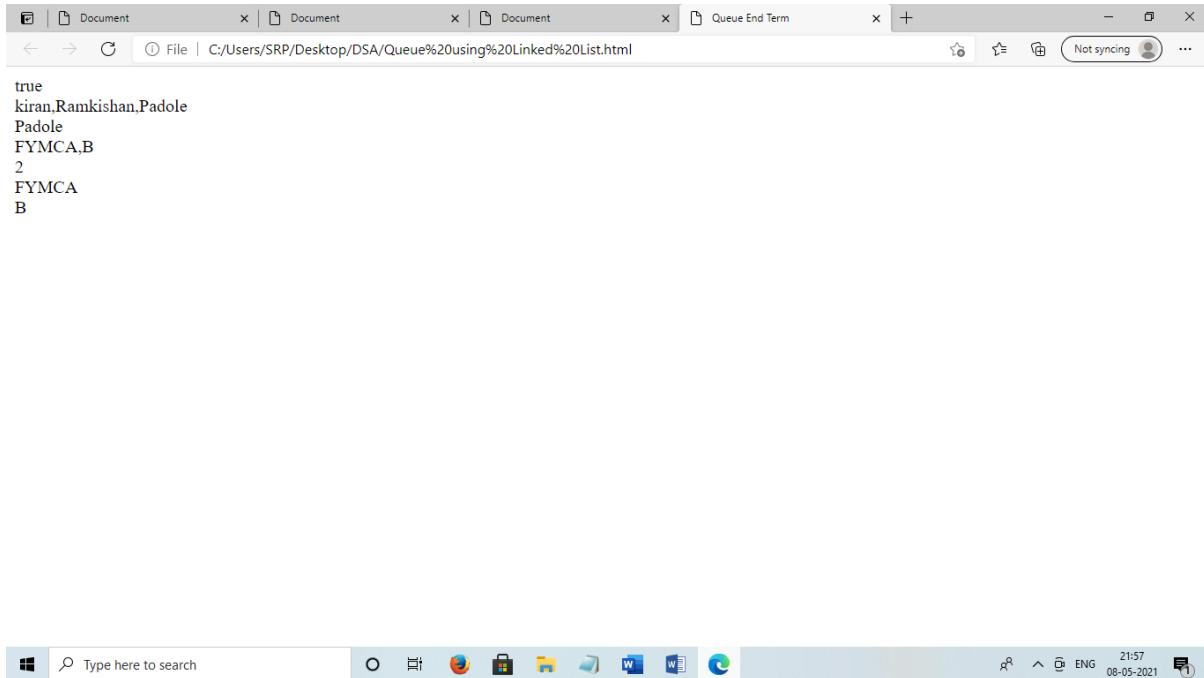
document.write(queue.rear()+"<br>")+"<br>";

</script>

</body>

</html>

```



#### 14. WAP for implementation of Graph and Adjacency List

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Document</title>

<script>

class Graph {

    // defining vertex array and
    // adjacent list
    constructor(noOfVertices)

    {
        this.noOfVertices = noOfVertices;
        this.AdjList = new Map();
    }

    // functions to be implemented

    // addVertex(v)
    addVertex(v)
```

```
{  
    // initialize the adjacent list with a  
    // null array  
    this.AdjList.set(v, []);  
  
}  
  
// addEdge(v, w)  
  
addEdge(v, w)  
  
{  
    // get the list for vertex v and put the  
    // vertex w denoting edge between v and w  
    this.AdjList.get(v).push(w);  
  
  
    // Since graph is undirected,  
    // add an edge from w to v also  
    this.AdjList.get(w).push(v);  
  
}  
  
// printGraph()  
printGraph()  
  
{  
    // get all the vertices  
    var get_keys = this.AdjList.keys();
```

```
// iterate over the vertices
for (var i of get_keys)
{
    // great the corresponding adjacency list
    // for the vertex
    var get_values = this.AdjList.get(i);
    var conc = "";

    // iterate over the adjacency list
    // concatenate the values into a string
    for (var j of get_values)
        conc += j + " ";

    // print the vertex and its adjacency list
    document.write(i + " -> " + conc + "<br/>");

}
// bfs(v)
// dfs(v)
}
```

```
// Using the above implemented graph class

var g = new Graph(6);

var vertices = [ 'A', 'B', 'C', 'D', 'E', 'F' ];

// adding vertices

for (var i = 0; i < vertices.length; i++) {

    g.addVertex(vertices[i]);

}

// adding edges

g.addEdge('A', 'B');

g.addEdge('A', 'D');

g.addEdge('A', 'E');

g.addEdge('B', 'C');

g.addEdge('D', 'E');

g.addEdge('E', 'F');

g.addEdge('E', 'C');

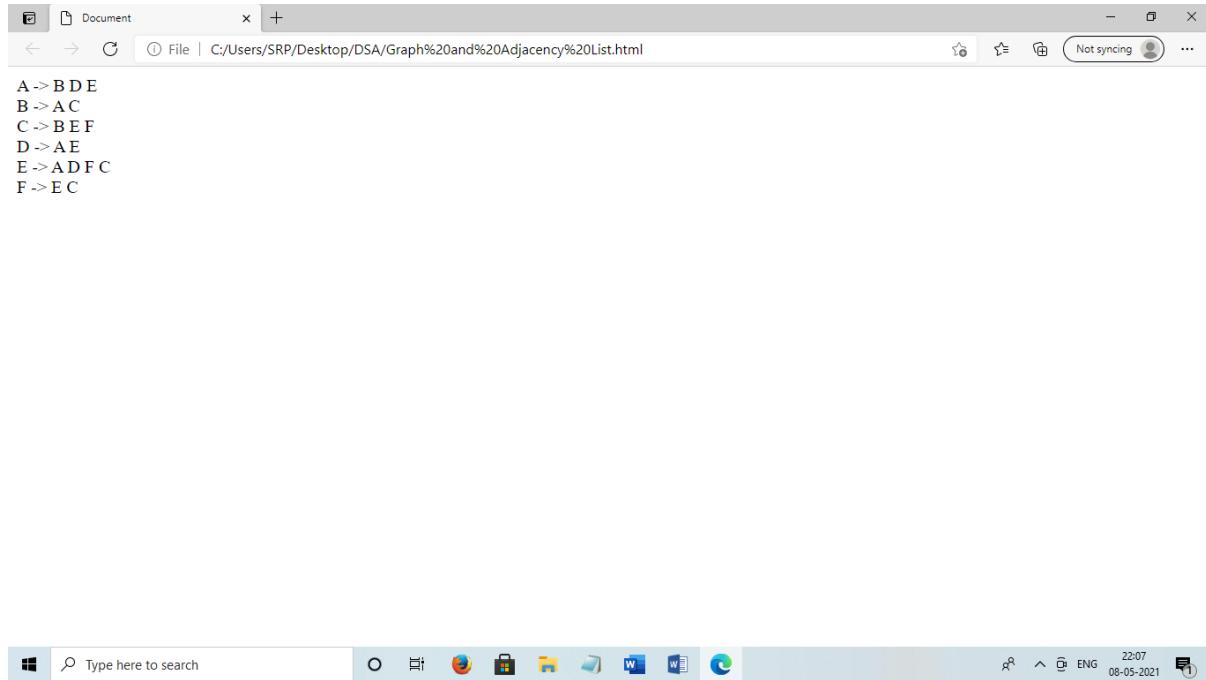
g.addEdge('C', 'F');

g.printGraph();
```

```
</script>
```

```
</body>
```

```
</html>
```



## 15. WAP for implementation of BFS and DFS of Graph

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>Document</title>
```

```
<script>

class Graph{

    // defining vertex array and
    // adjacent list

    constructor(noOfVertices)

    {

        this.noOfVertices = noOfVertices;
        this.AdjList = new Map();

    }

    // functions to be implemented

    // addVertex(v)
    addVertex(v)

    {

        // initialize the adjacent list with a
        // null array

        this.AdjList.set(v, []);

    }

    // addEdge(v, w)
    addEdge(v, w)

    {

        // get the list for vertex v and put the
        // vertex w denoting edge between v and w

    }
}
```

```

this.AdjList.get(v).push(w);

// Since graph is undirected,
// add an edge from w to v also
this.AdjList.get(w).push(v);

}

// printGraph()
printGraph()

{

// get all the vertices
var get_keys = this.AdjList.keys();

// iterate over the vertices
for (var i of get_keys)

{

// great the corresponding adjacency list
// for the vertex
var get_values = this.AdjList.get(i);
var conc = "";

// iterate over the adjacency list
// concatenate the values into a string
for (var j of get_values)
    conc += j + " ";
}

```

```

// print the vertex and its adjacency list

document.write(i + " -> " + conc + "<br/>");

}

}

// bfs(v)

bfs(v)

{

let q=[];

q.push(v);

let visited=[];

let keys=this.AdjList.keys();

for(let v of keys)

{

visited[v]=false;

}

while(q.length>0)

{

let element =q.shift();

visited[element]=true;

document.write("Popped element is :" +element+ "<br/>");

let eList=this.AdjList.get(element);

for(let eData in eList)

{

let e=eList[eData];

if(!visited[e])

```

```
{  
    q.push(e);  
    document.write("Added element is:"+e+"<br/>");  
    visited[e]=true;  
}  
  
}  
  
}
```

```
// dfs(v)  
dfs(v)  
{  
    let visited=[];  
    let keys=this.AdjList.keys();  
    for(let v of keys)  
    {  
        visited[v]=false;  
    }  
    this.dfsRecu(v,visited);  
}  
  
dfsRecu(v,visited)  
{
```

```

let eList=this.AdjList.get(v);

visited[v]=true;

document.write("Popped element:"+v+"<br/>");

for(let eData in eList)

{

    let e=eList[eData];

    if(!visited[e])

    {

        visited[e]=true;

        document.write("Added element:"+e+"<br/>");

        this.dfsRecu(e,visited);

    }

}

}

}

// Using the above implemented graph class

var g = new Graph(6);

var vertices = [ 'A', 'B', 'C', 'D', 'E', 'F' ];



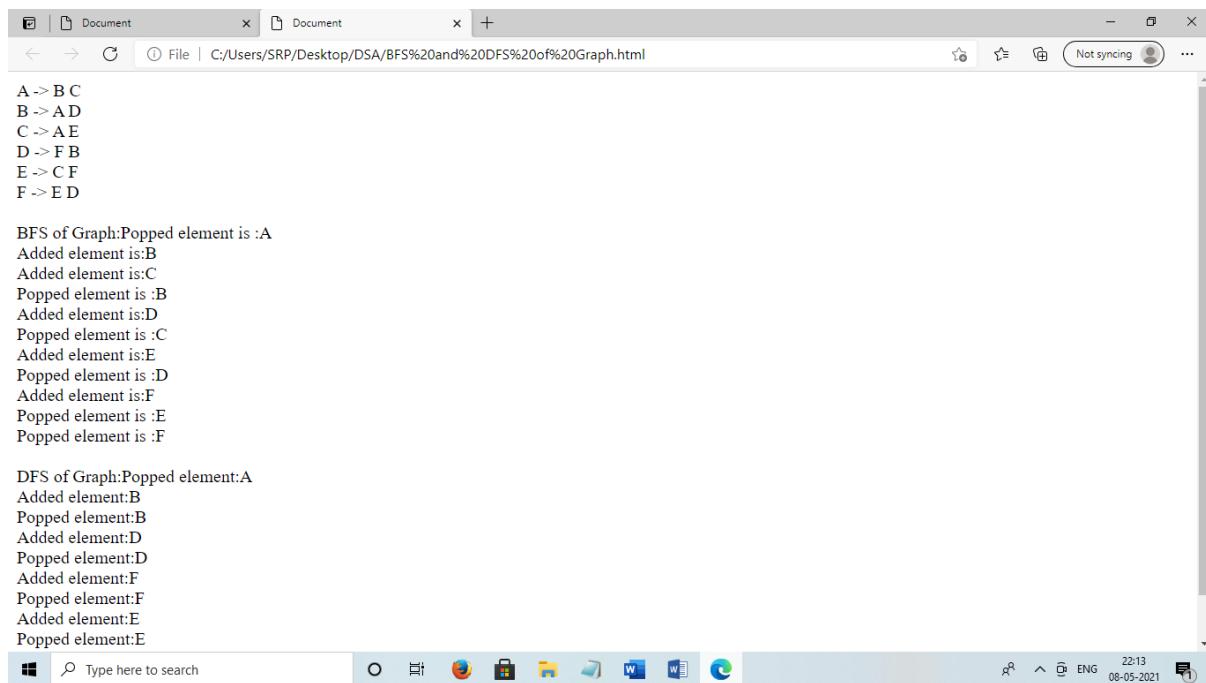
// adding vertices

for (var i = 0; i < vertices.length; i++) {

```

```
g.addVertex(vertices[i]);  
}  
  
// adding edges  
g.addEdge('A', 'B');  
g.addEdge('A', 'C');  
g.addEdge('C', 'E');  
g.addEdge('E', 'F');  
g.addEdge('D', 'F');  
g.addEdge('B', 'D');  
  
g.printGraph();  
document.write("<br/> BFS of Graph:");  
g.bfs('A');  
document.write("<br/> DFS of Graph:");  
g.dfs('A');  
  
</script>  
</body>
```

```
</html>
```



A -> B C  
B -> A D  
C -> A E  
D -> F B  
E -> C F  
F -> E D

BFS of Graph:Popped element is :A  
Added element is:B  
Added element is:C  
Popped element is :B  
Added element is:D  
Popped element is :C  
Added element is:E  
Popped element is :D  
Added element is:F  
Popped element is :E  
Popped element is :F

DFS of Graph:Popped element:A  
Added element:B  
Popped element:B  
Added element:D  
Popped element:D  
Added element:F  
Popped element:F  
Added element:E  
Popped element:E

## 16.WAP for implementation of Binary Search Tree Operations

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>Document</title>
```

```
<script>
```

```
class Node {
```

```
    constructor(value) {
```

```

        this.val = value;
        this.leftChild = null;
        this.rightChild = null;
    }

}

class BinarySearchTree {
    constructor(rootValue) {
        this.root = new Node(rootValue);
    }

    insert(currentNode, newValue) {
        if (currentNode === null) {
            currentNode = new Node(newValue);
        } else if (newValue < currentNode.val) {
            currentNode.leftChild = this.insert(currentNode.leftChild, newValue);
        } else {
            currentNode.rightChild = this.insert(currentNode.rightChild, newValue);
        }
        return currentNode;
    }

    insertBST(newValue) {
        if(this.root==null){
            this.root=new Node(newValue);
        }
    }
}

```

```

        return;
    }

    this.insert(this.root, newValue);

}

removeBST(data)
{
    // root is re-initialized with
    // root of a modified tree.

    this.root = this.removeNode(this.root, data);

}

removeNode(node, data)
{
    // if the root is null then tree is
    // empty
    if(node === null)

        return null;

    // if data to be delete is less than
    // roots data then move to left subtree
    else if(data < node.val)

    {
        node.leftChild = this.removeNode(node.leftChild, data);

        return node;
    }
}

```

```
}

// if data to be delete is greater than
// roots data then move to right subtree
else if(data > node.val)
{
    node.rightChild = this.removeNode(node.rightChild, data);
    return node;
}

// if data is similar to the root's data
// then delete this node
else
{
    // deleting node with no children
    if(node.leftChild === null && node.rightChild === null)
    {
        node = null;
        return node;
    }

    // deleting node with one children
    if(node.leftChild === null)
    {
        node = node.rightChild;
```

```

        return node;

    }

else if(node.rightChild === null)

{
    node = node.leftChild;

    return node;
}

// Deleting node with two children

// minimum node of the right subtree

// is stored in aux

var aux = this.findMinNode(node.rightChild);

node.val = aux.val;

node.rightChild = this.removeNode(node.rightChild, aux.val);

return node;

}

findMinNode(node)

{
    // if left of a node is null

    // then it must be minimum node

    if(node.leftChild === null)

```

```
    return node;  
  }  
  
  else  
    return this.findMinNode(node.leftChild);  
}
```

```
preOrderPrint(currentNode) {  
  if (currentNode!==null) {  
    document.write(currentNode.val);  
    this.preOrderPrint(currentNode.leftChild);  
    this.preOrderPrint(currentNode.rightChild);  
  }  
}
```

```
inOrderPrint(currentNode) {  
  if (currentNode!==null) {  
    this.inOrderPrint(currentNode.leftChild);  
    document.write(currentNode.val);  
    this.inOrderPrint(currentNode.rightChild);  
  }  
}
```

```
}
```

```
postOrderPrint(currentNode) {  
    if (currentNode !== null) {  
        this.postOrderPrint(currentNode.leftChild);  
        this.postOrderPrint(currentNode.rightChild);  
        document.write(currentNode.val);  
    }  
}
```

```
search(data)  
{  
    let currNode = this.root;  
    while (currNode !== null) {  
        if (currNode.val === data) {  
            // Found the element!  
            return true;  
        } else if (data < currNode.val) {  
            // Go Left as data is smaller than parent  
            currNode = currNode.leftChild;  
        } else {  
            // Go right as data is greater than parent  
            currNode = currNode.rightChild;  
        }  
    }  
}
```

```
    return false;

}

}

var BST = new BinarySearchTree(6);

document.write("The root val for BST : "+BST.root.val+"<br/>")

BST.insertBST(4);

BST.insertBST(9);

BST.insertBST(5);

BST.insertBST(2);

BST.insertBST(8);

BST.insertBST(12);

document.write("<br/> Preorder of BST : ");

BST.preOrderPrint(BST.root);

document.write("<br/> Inorder of BST : ");

BST.inOrderPrint(BST.root);

document.write("<br/> Postorder of BST : ");

BST.postOrderPrint(BST.root);

BST.removeBST(8);

document.write("<br/> after removal Inorder of BST : ");

BST.inOrderPrint(BST.root);
```

```

document.write("<br/>Value found:"+BST.search(15));

BST.removeBST(9);

document.write("<br/> after removal Inorder of BST : ");

BST.inOrderPrint(BST.root);

var bst1=JSON.stringify(BST);

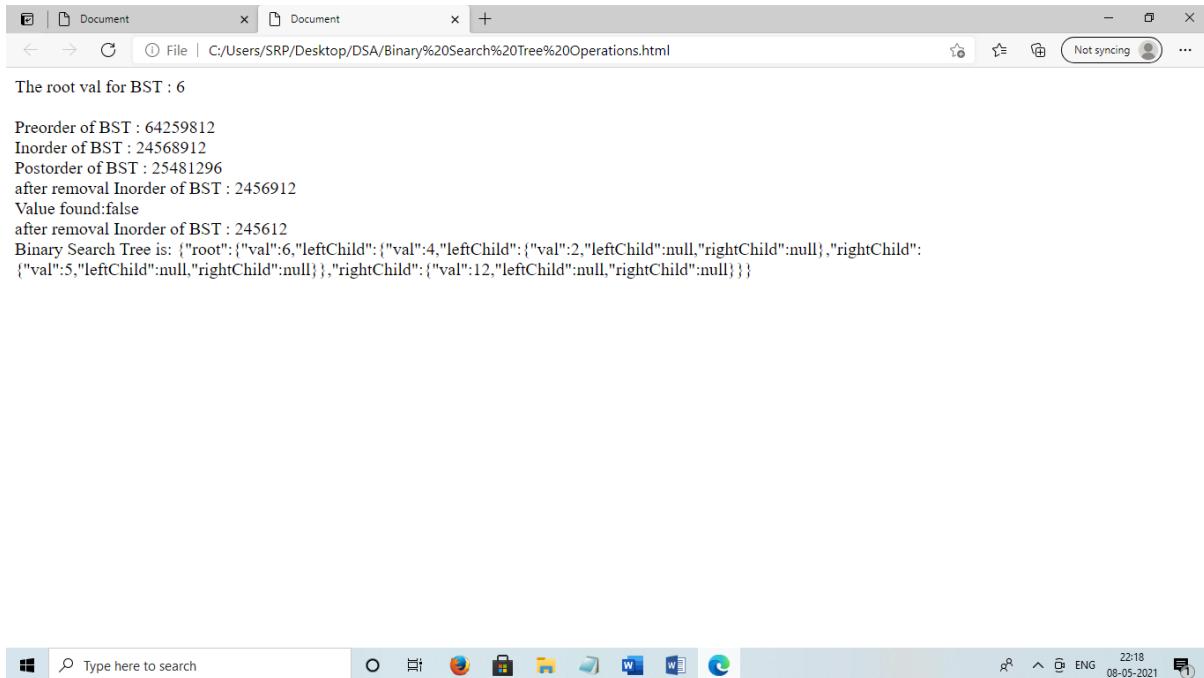
document.write("<br/> Binary Search Tree is: "+ bst1+"<br/>");

</script>

</body>

</html>

```



## 17. WAP for implementation of Simple hash function

```
<!DOCTYPE html>
<html lang="en">
<head>

<title>Document</title>

<script>

function Hashtable()
{
    this.table= new Array(15);
    this.simpleHash=simpleHash;
    this.showDistro=showDistro;
    this.put=put;
}

function simpleHash(data)
{
    var total=0;
    /*for(var i=0;i<data.length;++i)
    {
        total+=data.charCodeAt(i);
    }*/
}
```

```
total =data % this.table.length;

return total;

}

function put(data)

{

var pos=this.simpleHash(data);

this.table[pos]=data;

}

function showDistro()

{

var n=0;

for(var i=0;i<this.table.length;++i)

{

if(this.table[i]!=undefined)

{

document.write(i+":"+this.table[i]+<br/>);

}

}
```

```
}
```

```
}
```

```
var input=[8,3,13,6,4,10,24,72];
```

```
console.log(input);
```

```
let hTable= new Hashtable();
```

```
for(var i=0;i<input.length;i++)
```

```
{
```

```
hTable.put(input[i]);
```

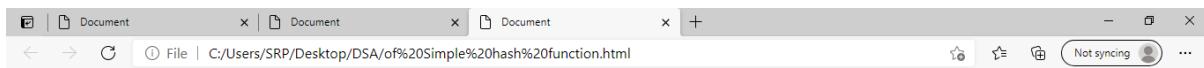
```
}
```

```
hTable.showDistro();
```

```
</script>
```

```
</body>
```

```
</html>
```



3:3  
4:4  
6:6  
8:8  
9:24  
10:10  
12:72  
13:13



## 1. WAP for implementation of Separate chaining hashing

```
<!DOCTYPE html>

<html lang="en">

<head>
```

```
<title>Document</title>
```

```
<script>

  class HashTable {

    constructor() {

      this.table = new Array(10);

      this.values = [];
    }
  }
}
```

```
}
```

```
// Defining the hashing function which allows a sting to be used as a key
```

```
hash(string) {
```

```
    const H = 37;
```

```
    let total = 0;
```

```
/*for (var i = 0; i < string.length; i++) {
```

```
    total += H * total + string.charCodeAt(i);
```

```
}*/
```

```
total = string % this.table.length;
```

```
if (total < 1) {
```

```
    this.table.length -1
```

```
}
```

```
return parseInt(total);
```

```
}
```

```
showDistro() {
```

```
    for (const key in this.table) {
```

```
        if(this.table[key] !== undefined) {
```

```
            console.log(key, ':', this.table[key]);
```

```
        }
```

```
}
```

```
}
```

```
put(data) {  
    const pos = this.hash(data);  
    this.table[pos] = data;  
}  
  
get(key) {  
    return this.table[this.hash(key)];  
}  
}  
  
// HashTable with Build Chains technique of collision-resolution.  
class HashTableChains extends HashTable {  
    constructor() {  
        super();  
        this.buildChains();  
    }  
    buildChains() {  
        for (var i = 0; i < this.table.length; i++) {  
            this.table[i] = new Array();  
        }  
    }  
    showDistro() {
```

```
for (const key in this.table) {  
  if(this.table[key][0] !== undefined) {  
    document.write(key, ':', this.table[key]+<br/>);  
  }  
}  
}
```

```
put(key, data) {  
  const pos = this.hash(key);  
  let index = 0;  
  if(this.table[pos][index] === undefined) {  
    this.table[pos][index] = data;  
  } else {  
    ++index;  
    while (this.table[pos][index] !== undefined ) {  
      index++;  
    }  
    this.table[pos][index] = data;  
  }  
}
```

```
get(key) {  
  const pos = this.hash(key);  
  let index = 0;
```

```

        while (this.table[pos][index] != key) {
            if(this.table[pos][index] !== undefined) {
                return this.table[pos][index]
            } else {
                return undefined;
            }
            index++;
        }
    }

}

// HashTable with Linear Probing technique of collision-resolution.

/*class HashTableLinearP extends HashTable {

constructor() {
    super();
    this.values = new Array();
}

put(key, data) {
    const pos = this.hash(key);
    if(this.table[pos] === undefined) {
        this.table[pos] = key;
        this.values[pos] = data;
    } else {

```

```
        while(this.table[pos] !== undefined) {  
            pos++;  
        }  
  
        this.table[pos] = key;  
        this.values[pos] = data;  
    }  
  
}  
  
  
get(key) {  
    const hash = this.hash(key);  
    if (hash > -1) {  
        for (let i = hash; this.table[i] !== undefined; i++) {  
            if (this.table[i] === key) {  
                return this.values[i];  
            }  
        }  
    }  
    return undefined;  
}  
  
  
showDistro() {  
    for (const key in this.table) {  
        if(this.table[key] !== undefined) {  
            console.log(key, ':', this.values[key]);  
        }  
    }  
}
```

```
    }
}

}

}*/



var input=[8,3,13,6,4,10];

let hTable= new HashTable();

let hTable1= new HashTableChains();

//let hTable2= new HashTableLinearP();



for(var i=0;i<input.length;i++)

{

    hTable.put(input[i]);


}

hTable.showDistro();



var input1=[10,22,86,12,42,20];

for(var i=0;i<input1.length;i++)

{

    hTable1.put(input1[i],input1[i]);

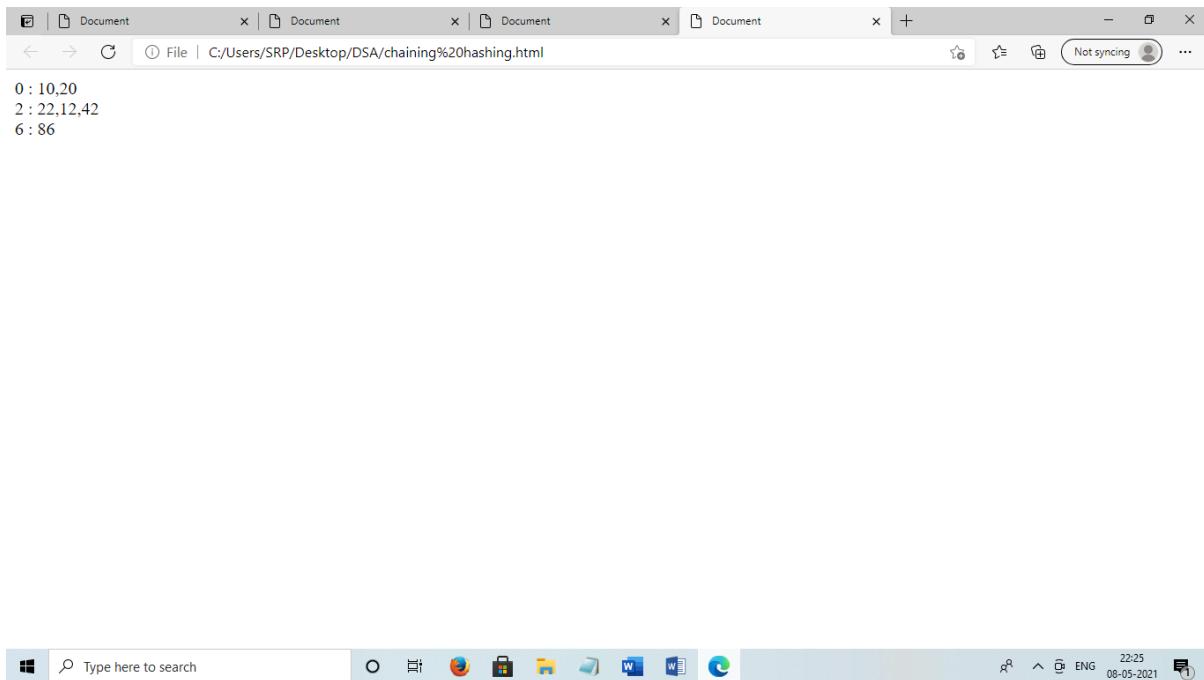

}

hTable1.showDistro();
```

```
</script>
```

```
</body>
```

```
</html>
```



## 18. WAP for implementation of Linear probing hashing

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>Document</title>
```

```
<script>
```

```
function HashTable(size) {  
    this.size = size;  
    this.keys = this.initArray(size);  
    this.values = this.initArray(size);  
    this.limit = 0;  
}
```

```
HashTable.prototype.put = function (key, value) {  
    if (this.limit >= this.size) throw 'hash table is full'  
    var hashedIndex = this.hash(key);  
    while (this.keys[hashedIndex] != null) {  
        hashedIndex++;  
        hashedIndex = hashedIndex % this.size;  
    }  
    this.keys[hashedIndex] = key;  
    this.values[hashedIndex] = value;  
    this.limit++;
```

```
        console.log(this.keys, this.values);  
    }  
  
}
```

```
HashTable.prototype.get = function (key) {  
    var hashedIndex = this.hash(key);
```

```
    while (this.keys[hashedIndex] != key) {  
        hashedIndex++;  
        hashedIndex = hashedIndex % this.size;  
    }
```

```
    return this.values[hashedIndex]  
}
```

```
HashTable.prototype.hash = function (key) {  
    if (!Number.isInteger(key)) throw 'must be int';
```

```
    return key % this.size;  
}
```

```
HashTable.prototype.initArray = function (size) {  
    var array = [];  
  
    for (var i = 0; i < size; i++) {  
        array.push(null);  
    }  
  
    return array;  
}
```

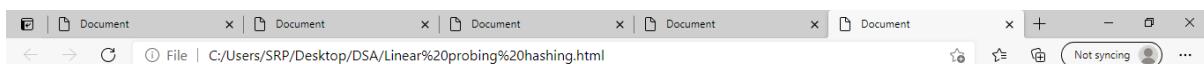
```
HashTable.prototype.showDistro=function() {  
    for (const key in this.keys) {  
        if(this.keys[key] !== undefined) {  
            document.write(key, ': ', this.values[key]+<br/>);  
        }  
    }  
  
    var exampletable = new HashTable(10);  
    exampletable.put(38,38);  
    exampletable.put(19,19);  
    exampletable.put(8,8);
```

```
exampletable.put(79,79);  
exampletable.put(10,10);  
exampletable.showDistro();
```

```
</script>
```

```
</body>
```

```
</html>
```



```
0 : 8  
1 : 79  
2 : 10  
3 : null  
4 : null  
5 : null  
6 : null  
7 : null  
8 : 38  
9 : 19
```



## 19. WAP for implementation of Linear probing hashing

```
<!DOCTYPE html>
<html lang="en">
<head>

<title>Document</title>

<script>

function HashTable(size) {
    this.size = size;
    this.keys = this.initArray(size);
    this.values = this.initArray(size);
    this.limit = 0;
}

HashTable.prototype.put = function (key, value) {
    if (this.limit >= this.size) throw 'hash table is full'
    var hashedIndex = this.hash(key);
    while (this.keys[hashedIndex] != null) {
        var step=1
        while (this.keys[hashedIndex]) {
            hashedIndex=(hashedIndex+(step*step));
            hashedIndex = hashedIndex % this.size;
            step++;
        }
    }
    this.keys[hashedIndex] = key;
    this.values[hashedIndex] = value;
    this.limit++;
}

console.log(this.keys, this.values);
}

HashTable.prototype.get = function (key) {
    var hashedIndex = this.hash(key);

    while (this.keys[hashedIndex] != key) {
        hashedIndex++;
    }
}
```

```
        hashedIndex = hashedIndex % this.size;
    }

    return this.values[hashedIndex]
}

HashTable.prototype.hash = function (key) {
    if (!Number.isInteger(key)) throw 'must be int';

    return key % this.size;
}

HashTable.prototype.initArray = function (size) {
    var array = [];

    for (var i = 0; i < size; i++) {
        array.push(null);
    }

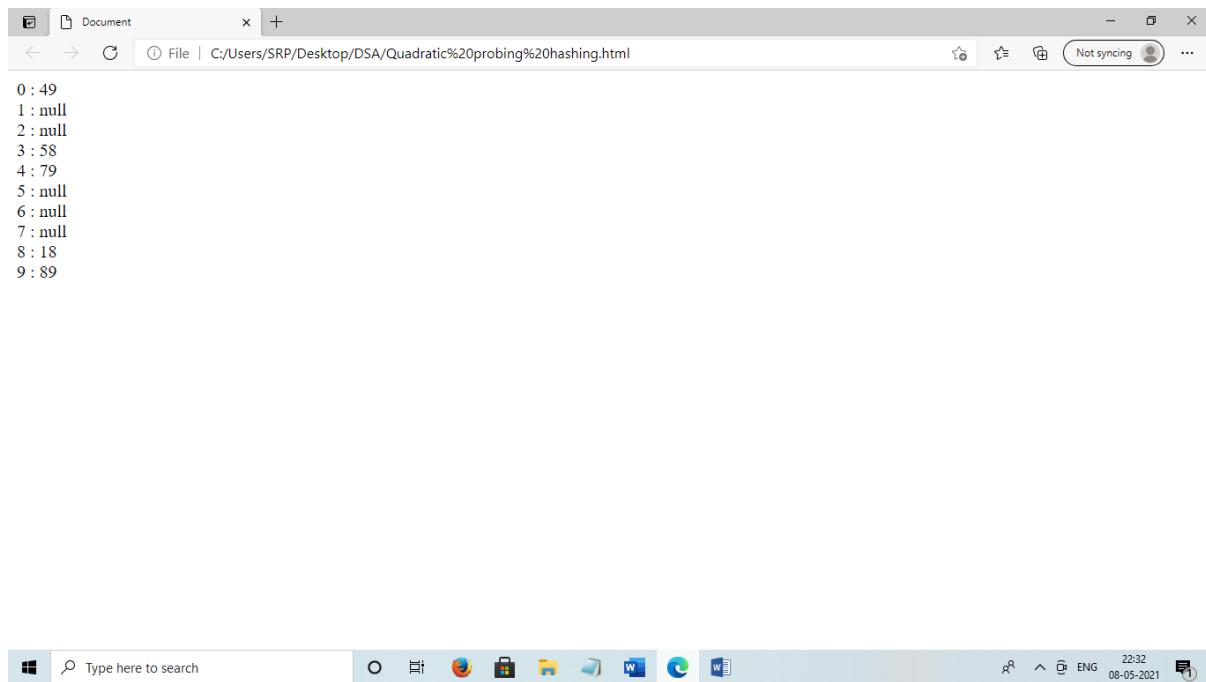
    return array;
}

HashTable.prototype.showDistro=function() {
    for (const key in this.keys) {
        if(this.keys[key] !== undefined) {
            document.write(key, ':', this.values[key]+<br/>);
        }
    }
}

var exampletable = new HashTable(10);
exampletable.put(89,89);
exampletable.put(18,18);
exampletable.put(49,49);
exampletable.put(58,58);
exampletable.put(79,79);
exampletable.showDistro();

</script>
</body>
```

</html>



## 20. WAP for implementation of Quadratic probing hashing

<!DOCTYPE html>

<html lang="en">

```
<head>
```

```
<title>Document</title>
```

```
<script>
```

```
function HashTable(size) {  
    this.size = size;  
    this.keys = this.initArray(size);  
    this.values = this.initArray(size);  
    this.limit = 0;  
}
```

```
HashTable.prototype.put = function (key, value) {  
    if (this.limit >= this.size) throw 'hash table is full'  
    var hashedIndex = this.hash(key);  
    while (this.keys[hashedIndex] != null) {  
        var step=1  
        var secondary=1+key%(this.keys.length-2);  
        while (this.keys[hashedIndex]) {  
            hashedIndex=(hashedIndex+(step*secondary));
```

```
        hashedIndex = hashedIndex % this.size;  
        step++;  
    }  
}  
  
this.keys[hashedIndex] = key;  
this.values[hashedIndex] = value;  
this.limit++;  
  
console.log(this.keys, this.values);  
}
```

```
HashTable.prototype.get = function (key) {  
    var hashedIndex = this.hash(key);  
  
    while (this.keys[hashedIndex] != key) {  
        hashedIndex++;  
        hashedIndex = hashedIndex % this.size;  
    }  
  
    return this.values[hashedIndex]  
}
```

```
HashTable.prototype.hash = function (key) {  
    if (!Number.isInteger(key)) throw 'must be int';  
  
    return key % this.size;  
}
```

```
HashTable.prototype.initArray = function (size) {  
    var array = [];  
  
    for (var i = 0; i < size; i++) {  
        array.push(null);  
    }  
  
    return array;  
}
```

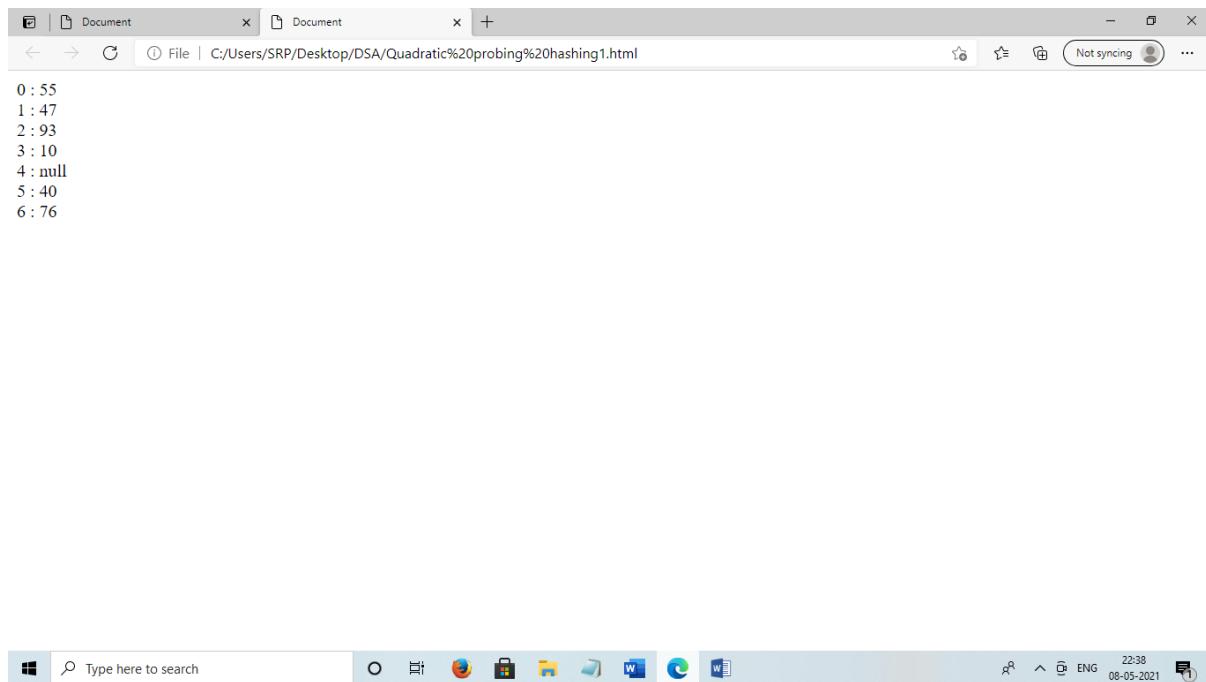
```
HashTable.prototype.showDistro=function() {  
    for (const key in this.keys) {  
        if(this.keys[key] !== undefined) {  
            document.write(key, ' : ', this.values[key]+<br/>);  
        }  
    }  
}
```

```
    }  
}  
}  
  
var exampletable = new HashTable(7);  
exampletable.put(76,76);  
exampletable.put(93,93);  
exampletable.put(40,40);  
exampletable.put(47,47);  
exampletable.put(10,10);  
exampletable.put(55,55);  
exampletable.showDistro();
```

```
</script>
```

```
</body>
```

```
</html>
```



## 21. WAP for implementation of Double Hashing

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>Document</title>
```

```
<script>
```

```
function HashTable(size) {
```

```
    this.size = size;
```

```

this.keys = this.initArray(size);

this.values = this.initArray(size);

this.limit = 0;

}

HashTable.prototype.put = function (key, value) {

if (this.limit >= this.size) throw 'hash table is full'

var hashedIndex = this.hash(key);

while (this.keys[hashedIndex] != null) {

var step=1

var secondary=1+key%(this.keys.length-2);

while (this.keys[hashedIndex]) {

hashedIndex=(hashedIndex+(step*secondary));

hashedIndex = hashedIndex % this.size;

step++;

}

this.keys[hashedIndex] = key;

this.values[hashedIndex] = value;

this.limit++;




console.log(this.keys, this.values);

}

HashTable.prototype.get = function (key) {

```

```
var hashedIndex = this.hash(key);

while (this.keys[hashedIndex] != key) {
    hashedIndex++;
    hashedIndex = hashedIndex % this.size;
}

return this.values[hashedIndex]
```

```
HashTable.prototype.hash = function (key) {
    if (!Number.isInteger(key)) throw 'must be int';

    return key % this.size;
}
```

```
HashTable.prototype.initArray = function (size) {
    var array = [];

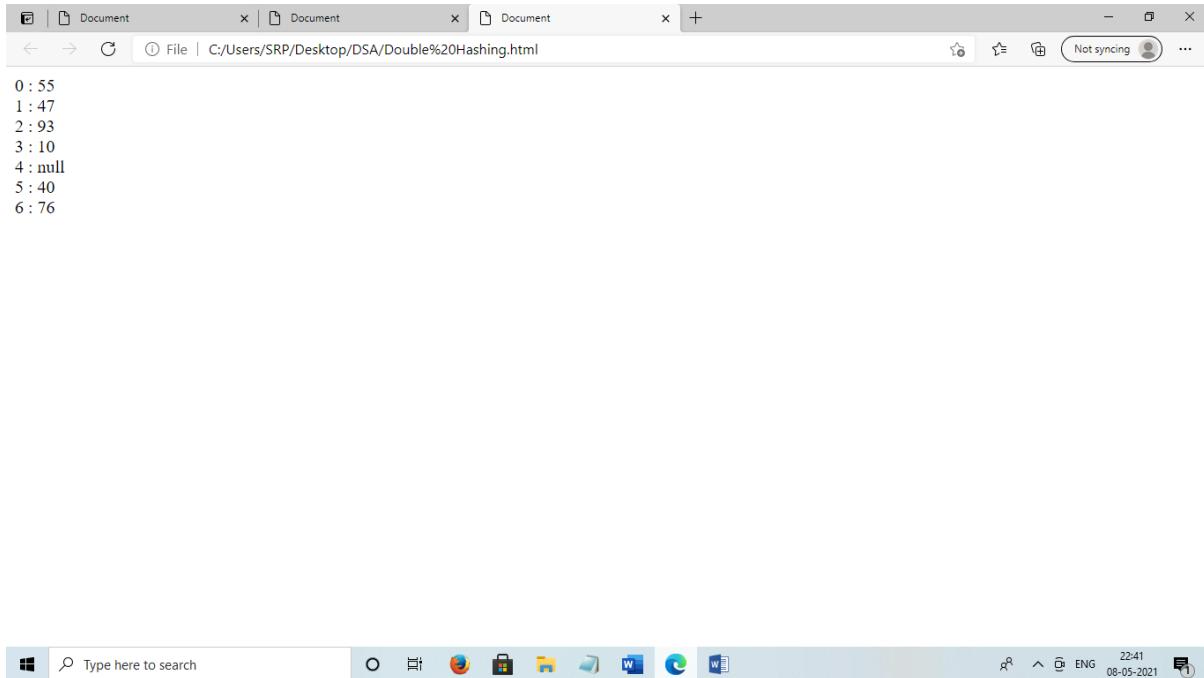
    for (var i = 0; i < size; i++) {
        array.push(null);
    }

    return array;
}
```

```
HashTable.prototype.showDistro=function() {
    for (const key in this.keys) {
        if(this.keys[key] !== undefined) {
            document.write(key, ': ', this.values[key]+<br/>);
        }
    }
}

var exampletable = new HashTable(7);
exampletable.put(76,76);
exampletable.put(93,93);
exampletable.put(40,40);
exampletable.put(47,47);
exampletable.put(10,10);
exampletable.put(55,55);
exampletable.showDistro();

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



## 22. WAP for implementation of Prim's Algorithm

```
<!DOCTYPE html>
<html lang="en">
<head>

<title>Prim's Algorithm</title>

<script>
class MST {
    // Number of vertices in the graph

    // A utility function to find the vertex with minimum key
    // value, from the set of vertices not yet included in MST
    minKey(key,mstSet,MAX_VALUE)
    {
        // Initialize min value
        let min = MAX_VALUE, min_index = -1;
```

```

for (let v = 0; v < V; v++) {
    if (mstSet[v] == false && key[v] < min) {
        min = key[v];
        min_index = v;
    }
}

return min_index;
}

// A utility function to print the constructed MST stored in
// parent[]
printMST(parent,graph)
{
    console.log("Edge Weight");
    for (let i = 1; i < V; i++)
        document.write(parent[i] + " - " + i + ":" + graph[i][parent[i]]+<br/>);
}

// Function to construct and print MST for a graph represented
// using adjacency matrix representation
primMST(graph,V)
{
    // Array to store constructed MST
    let parent = new Array(V);

    // Key values used to pick minimum weight edge in cut
    let key = new Array(V);

    // To represent set of vertices included in MST
    let mstSet = new Array(V);

    // Initialize all keys as INFINITE
    for (let i = 0; i < V; i++) {
        key[i] = MAX_VALUE;
        mstSet[i] = false;
    }

    // Always include first 1st vertex in MST.
    key[0] = 0; // Make key 0 so that this vertex is
    // picked as first vertex
}

```

```

parent[0] = -1; // First node is always root of MST

// The MST will have V vertices
for (let count = 0; count < V - 1; count++) {
    // Pick thd minimum key vertex from the set of vertices
    // not yet included in MST
    let u = this.minKey(key, mstSet,MAX_VALUE);

    // Add the picked vertex to the MST Set
    mstSet[u] = true;

    // Update key value and parent index of the adjacent
    // vertices of the picked vertex. Consider only those
    // vertices which are not yet included in MST
    for (let z = 0; z < V; z++)
        // let ans=graph[u][v];
        // console.log(ans.charCodeAt());
        // graph[u][v] is non zero only for adjacent vertices of m
        // mstSet[v] is false for vertices not yet included in MST
        // Update the key only if graph[u][v] is smaller than key[v]
        if (graph[u][z] != 0 && mstSet[z] == false && graph[u][z] < key[z])
        {
            parent[z] = u;
            key[z] = graph[u][z];
        }
    }

    // print the constructed MST
    this.printMST(parent, graph);
}

let t = new MST();
let graph = [ [ 0, 2, 5, 3],
              [ 2, 0, 2, 0],
              [ 5, 2, 0, 3],
              [ 3, 0, 3, 0]];

```

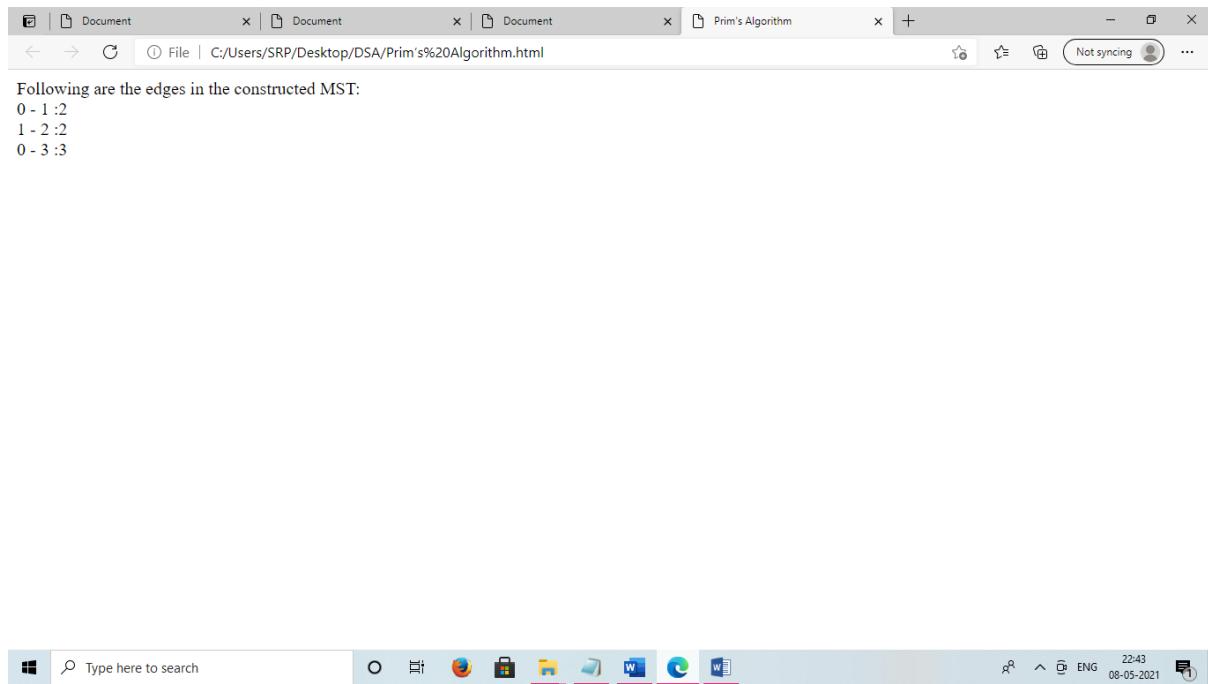
```

let V=4, MAX_VALUE=99999;

// Print the solution
    document.write("Following are the edges in the constructed MST:<br/>");
t.primMST(graph,V,MAX_VALUE);

</script>
</body>
</html>

```



### 23. WAP for implementation of Kruskal's Algorithm

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```
<title>Tower of Hanoi</title>
```

```
<script>

const MAX_INTEGER = Number.MAX_SAFE_INTEGER; // No edge

const MIN_INTEGER = Number.MIN_SAFE_INTEGER; // No self-loop

const matrix= [
    [MIN_INTEGER, 2, 5, 3],
    [2, MIN_INTEGER, 2, MIN_INTEGER],
    [5, 2, MIN_INTEGER, 3],
    [3, MIN_INTEGER, 3, MIN_INTEGER]
];
```

```
function Edge(begin, end, weight) {
    this.begin = begin;
    this.end = end;
    this.weight = weight;
}
```

```
Edge.prototype.getBegin = function () {
    return this.begin;
};
```

```
Edge.prototype.getEnd = function () {
    return this.end;
};

Edge.prototype.getWeight = function () {
    return this.weight;
};

function changeMatrixToEdgeArray(matrix) {
    const rows = matrix.length,
        cols = rows,
        result = [];
    for (let i = 0; i < rows; i++) {
        const row = matrix[i];
        for(let j = 0 ; j < cols; j++) {
            if(row[j] !== MIN_INTEGER && row[j] !== MAX_INTEGER) {
                result.push(new Edge(i, j, row[j]));
                matrix[i][j] = MAX_INTEGER;
                matrix[j][i] = MAX_INTEGER;
            }
        }
    }
    result.sort((a, b) => a.getWeight() - b.getWeight());
    return result;
}
```

```
}
```

```
function kruskal(matrix) {
```

```
    const edgeArray = changeMatrixToEdgeArray(matrix),
```

```
    result = [],
```

```
        // Use an array to save the end point of the edge of the current vertex, 0 means that  
no edge has been added as the starting point
```

```
    savedEdge = new Array(matrix.length).fill(0);
```

```
    for (let i = 0, len = edgeArray.length-2; i < len; i++) {
```

```
        const edge = edgeArray[i];
```

```
        const n = findEnd(savedEdge, edge.getBegin());
```

```
        const m = findEnd(savedEdge, edge.getEnd());
```

```
        document.write(savedEdge+"-----"+n+"--"+m+"---"+edge.getWeight());
```

```
        document.write("<br/>>");
```

```
        // Unequal means that this edge does not form a loop with the existing spanning tree
```

```
        if (n !== m) {
```

```
            result.push(edge);
```

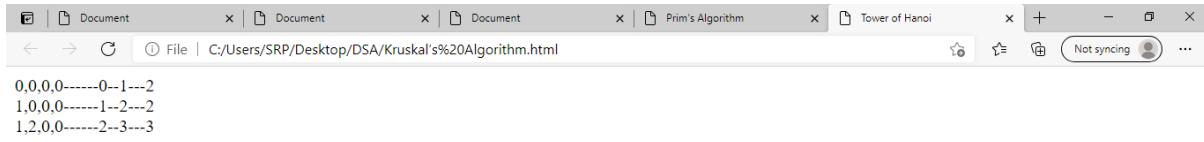
```
            // Add the ending vertex of this edge to the array, indicating that the vertex is  
already in the spanning tree
```

```
            savedEdge[n] = m;
```

```
        }
```

```
}
```

```
return result;  
}  
  
function findEnd(arr, start) {  
    // It just keeps looping until it finds the end point, if there is no connection, it returns 0  
    while (arr[start] > 0) {  
        start = arr[start];  
    }  
    return start;  
}  
  
var grp=new kruskal(matrix);  
</script>  
</body>  
</html>
```



The screenshot shows a browser window with four tabs open. The active tab is titled "Tower of Hanoi" and contains the following text:  
0,0,0-----0--1--2  
1,0,0,0-----1--2--2  
1,2,0,0-----2--3---3



## 24. WAP for implementation of Tower of Hanoi Algorithm

```
<!DOCTYPE html>
<html lang="en">
<head>

<title>Document</title>

<script>
function towerOfHanoi(n, from_rod, to_rod, aux_rod)
{
    if (n == 1)
    {
        document.write("Move disk 1 from rod " + from_rod + " to rod " +
to_rod+"<br/>");
        return;
    }
    towerOfHanoi(n - 1, from_rod, aux_rod, to_rod);
}
```

```

        document.write("Move disk " + n + " from rod " + from_rod + " to rod " +
to_rod+"<br/>");
        towerOfHanoi(n - 1, aux_rod, to_rod, from_rod);
    }

// Driver code
var n = 3; // Number of disks
towerOfHanoi(n, 'A', 'C', 'B'); // A, B and C are names of rods

</script>
</body>
</html>

```



## 25.WAP for implementation of Longest Common Subsequence

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<title>Document</title>

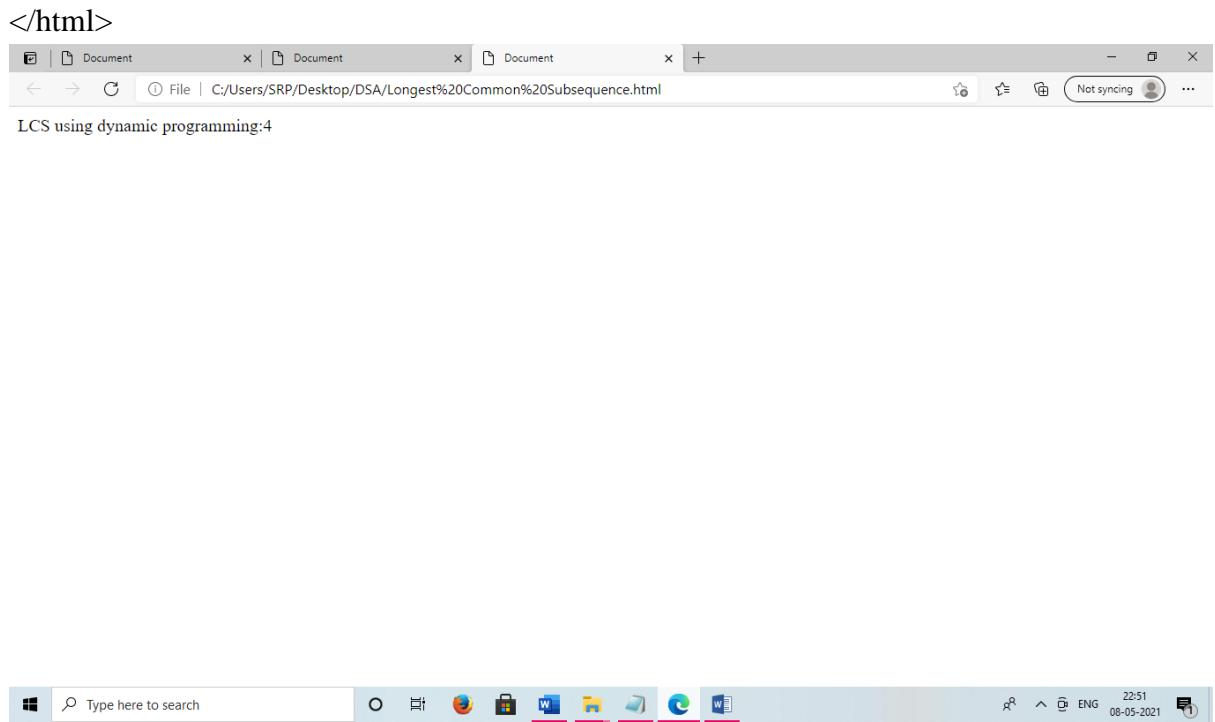
<script>

function LCS(str1, str2){
    var m = str1.length
    var n = str2.length
    var dp = [new Array(n+1).fill(0)] //The first line is all 0
    for(var i = 1; i <= m; i++){ //A total of m+1 lines
        dp[i] = [0] //The first column is all 0
        for(var j = 1; j <= n; j++){//A total of n+1 columns
            if(str1[i-1] === str2[j-1]){
                //Note here, the first character of str1 is in the second column, so you have to
                subtract 1, and str2 is the same
                dp[i][j] = dp[i-1][j-1] + 1 //Diagonal +
            } else {
                dp[i][j] = Math.max( dp[i-1][j], dp[i][j-1])
            }
        }
    }
    return dp[m][n];
}

var str1 = "ABACBDAB"; //BATD
var str2 = "BDCADB"; //ABACD
var lcs = LCS(str1,str2);
document.write("LCS using dynamic programming:"+lcs);

</script>
</body>

```



## 25. WAP for implementation of Longest Common Substring

```
<!DOCTYPE html>
<html lang="en">
<head>

<title>Substring</title>

<script>
() => {
  'use strict';

  // longestCommon :: String -> String -> String
  const longestCommon = (s1, s2) => maximumBy(
    comparing(length),
    intersect(...apList(
      [s => map(
        concat,
        concatMap(tails, compose(tail, inits))(s))
    ]),
    [s1, s2]
  )
}
```

```

        ))
);

// main :: IO ()
const main = () =>
    document.write(
        longestCommon(
            "ABCDXYZAY",
            "XYZABCDB"
        )
    );

```

#### // GENERIC FUNCTIONS -----

```

// Each member of a list of functions applied to each
// of a list of arguments, deriving a list of new values.

```

```

// apList (<*>) :: [(a -> b)] -> [a] -> [b]
const apList = (fs, xs) => //
    fs.reduce((a, f) => a.concat(
        xs.reduce((a, x) => a.concat([f(x)]), [])
    ), []);

```

```

// comparing :: (a -> b) -> (a -> a -> Ordering)
const comparing = f =>
    (x, y) => {
        const
            a = f(x),
            b = f(y);
        return a < b ? -1 : (a > b ? 1 : 0);
    };

```

```

// compose (<<<) :: (b -> c) -> (a -> b) -> a -> c
const compose = (f, g) => x => f(g(x));

```

```

// concat :: [[a]] -> [a]
// concat :: [String] -> String
const concat = xs =>

```

```

0 < xs.length ? () => {
  const unit = 'string' !== typeof xs[0] ? (
    []
  ) : '';
  return unit.concat.apply(unit, xs);
})() : [];

// concatMap :: (a -> [b]) -> [a] -> [b]
const concatMap = (f, xs) =>
  xs.reduce((a, x) => a.concat(f(x)), []);

// inits([1, 2, 3]) -> [[], [1], [1, 2], [1, 2, 3]]
// inits('abc') -> ["", "a", "ab", "abc"]

// inits :: [a] -> [[a]]
// inits :: String -> [String]
const inits = xs => [
  []
]
.concat('string' === typeof xs ? xs.split("") : xs)
.map((_, i, lst) => lst.slice(0, i + 1));

// intersect :: (Eq a) -> [a] -> [a]
const intersect = (xs, ys) =>
  xs.filter(x => -1 !== ys.indexOf(x));

// Returns Infinity over objects without finite length.
// This enables zip and zipWith to choose the shorter
// argument when one is non-finite, like cycle, repeat etc

// length :: [a] -> Int
const length = xs =>
  (Array.isArray(xs) || 'string' === typeof xs) ? (
    xs.length
  ) : Infinity;

// map :: (a -> b) -> [a] -> [b]
const map = (f, xs) => xs.map(f);

// maximumBy :: (a -> a -> Ordering) -> [a] -> a

```

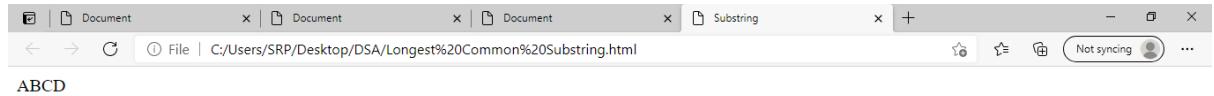
```
const maximumBy = (f, xs) =>
  0 < xs.length ? (
    xs.slice(1)
    .reduce((a, x) => 0 < f(x, a) ? x : a, xs[0])
  ) : undefined;

// tail :: [a] -> [a]
const tail = xs => 0 < xs.length ? xs.slice(1) : [];

// tails :: [a] -> [[a]]
const tails = xs => {
  const
    es = ('string' === typeof xs) ? (
      xs.split('')
    ) : xs;
  return es.map((_, i) => es.slice(i))
    .concat([
      []
    ]);
};

// MAIN ---
return main();
})();

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



## 27. WAP for implementation of Regular Expression Matching.

```
<!DOCTYPE html>

<html lang="en">
<head>

<title>Regex</title>

<script>

const regexMatching = (str, p) => {
    const ZERO_OR_MORE_CHARS = '*';
    const ANY_CHAR = '.';
    ...
}
```

```
const match = Array(str.length + 1).fill(null).map(() => {
    return Array(p.length + 1).fill(null);
});

match[0][0] = true;

for (let col = 1; col <= p.length; col += 1) {
    const patternIndex = col - 1;
    if (p[patternIndex] === ZERO_OR_MORE_CHARS) {
        match[0][col] = match[0][col - 2];
    } else {
        match[0][col] = false;
    }
}

for (let row = 1; row <= str.length; row += 1) {
    match[row][0] = false;
}

for (let row = 1; row <= str.length; row += 1) {
    for (let col = 1; col <= p.length; col += 1) {
        const stringIndex = row - 1;
        const patternIndex = col - 1;
        if (p[patternIndex] === ZERO_OR_MORE_CHARS) {
            if (match[row][col - 2] === true) {
                match[row][col] = true;
            } else if (
                p[patternIndex - 1] === str[stringIndex]
            ) {
                match[row][col] = true;
            }
        }
    }
}
```

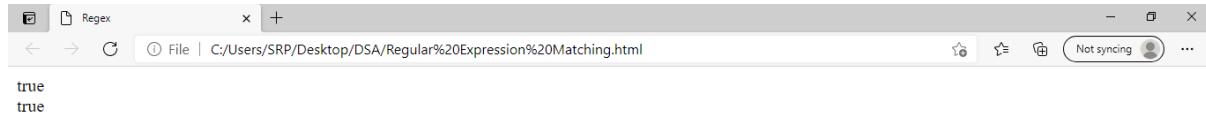
```

    || p[patternIndex - 1] === ANY_CHAR
    )
    && match[row - 1][col] === true
) {
    match[row][col] = true;
} else {
    match[row][col] = false;
}
} else if (
    p[patternIndex] === str[stringIndex]
    || p[patternIndex] === ANY_CHAR
) {
    match[row][col] = match[row - 1][col - 1];
} else {
    match[row][col] = false;
}
}

return match[str.length][p.length];
};

document.write(regexMatching('aab', 'c*a*b') + "<br/>");
document.write(regexMatching('axyb', 'a.*b'));
</script>
</body>
</html>

```



```
true
true
```



## 28. WAP for implementation of N Queen's Problem

```
var Queen = function (N) {
```

```
    if (N <= 0) {
```

```
        document.write('N is invalid');
```

```
        return null;
```

```
}
```

```
    return new Queen.fn.init(N);
```

```
}
```

```
Queen.fn = Queen.prototype = {
```

```
    constructor: Queen,
```

```
    init: function (N){
```

```
        var c = [];
```

```

this.N = N;
this.res = [];

// Recursive solution
this.traverse(c, 0);

// To support chained operations
return this;
}

};

// The prototype object of init points to the prototype of Queen, so that the object constructed by
init can use the method hanging on the Queen prototype.

Queen.fn.init.prototype = Queen.fn;

Queen.fn.traverse =function (c, row) {
if (row === this.N) {
    // The last layer is recursive, can be executed here to explain the solution is successful, save
    the solution results
    this.res.push(c.slice(0));
} else {
    // loop each column for detection
    for (let col = 0; col < this.N; col++) {
        // mark the column
        c[row] = col;
        // Check if the rules are met
        if (this.check(c, row)) {

```

```

    // Meet the rules and continue recursively to the next level
    this.traverse(c, row + 1);
}

}

}

}

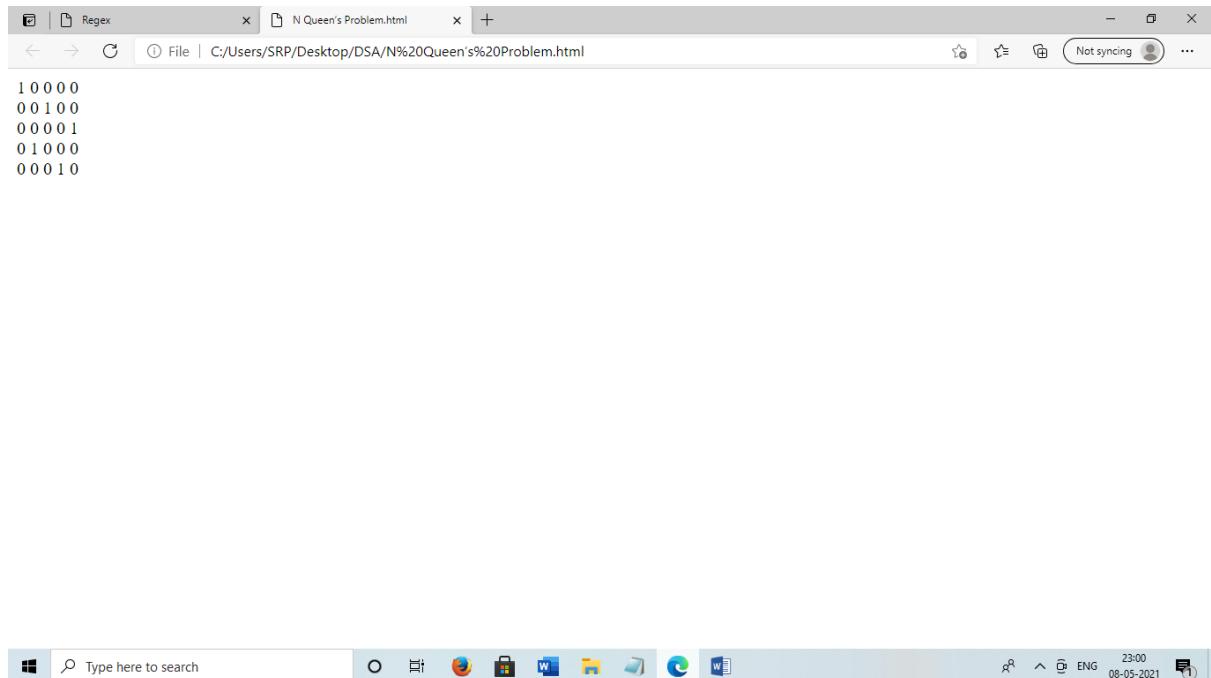
Queen.fn.check = function (c, row) {
    for (let i = 0; i < row; i++) {
        if (c[row] === c[i] || row + c[row] === i + c[i] || row - c[row] === i - c[i]) {
            return false;
        }
    }
    return true;
}

Queen.fn.printByIndex = function (index) {
    if (index < 1 || index > this.N) return;
    index--;
    let arr = [];
    for (let i = 0; i < this.N; i++) {
        for (let j = 0; j < this.N; j++) {
            arr[j] = (j === this.res[index][i]) ? '1': '0';
        }
    }
    document.write(arr.join('') + "<br/>");
}

```

```
let q = Queen(5);

q.printByIndex(1);</script></body></html>
```



## 29. WAP for implementation of Rain Terraces Problem

```
<!DOCTYPE html><html lang="en">

<head><title>Rain Terraces</title><script>

class RainTerraces

{

trap(height)

{
```

```

if(height.length==0)

{

return 0;

}

let ans = 0;

let size = height.length;

document.write("Size of array:"+size+"<br/>");

let left_max = [];

let right_max = [];


left_max[0] = height[0];

right_max[size - 1] = height[size - 1];

document.write("Left Max of array: "+left_max[0]+"<br/>");

document.write("Right Max of array: "+right_max[size - 1]+"<br/>");


for (let i = 1; i < size; i++) {

    left_max[i] = Math.max(height[i], left_max[i - 1]);

}

for (let i = size - 2; i >= 0; i--) {

    right_max[i] = Math.max(height[i], right_max[i + 1]);

}

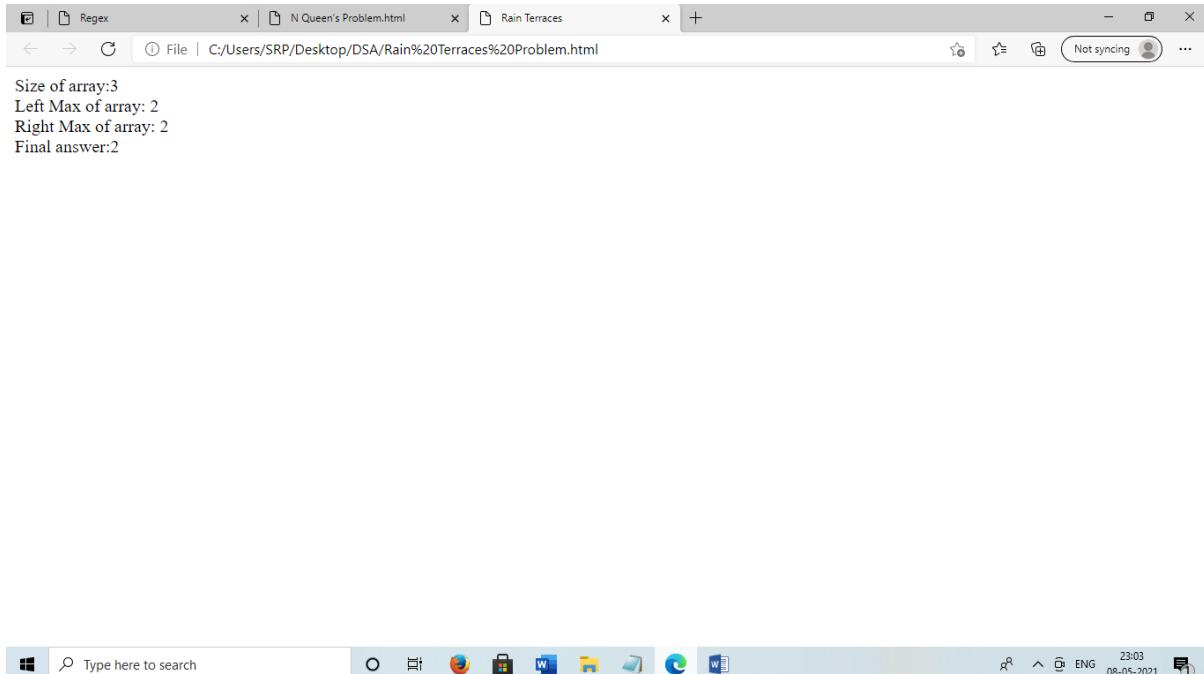
for (let i = 1; i < size - 1; i++) {

    ans = ans + Math.min(left_max[i], right_max[i]) - height[i];

}

```

```
document.write("Final answer:"+ans);  
}  
}  
  
let rain=new RainTerraces();  
  
let height=[2,0,2];  
  
rain.trap(height);  
</script></body></html>
```



### 30. WAP for implementation of Recursive Staircase Problem

```
<!DOCTYPE html><html lang="en"><head>
<title>Recursive Staircase</title>
<script>const recursiveStaircase = (num) => {
    if (num <= 0) {
        return 0;
    }
    const steps = [1, 2];
    if (num <= 2) {
        return steps[num - 1];
    }
    for (let currentStep = 3; currentStep <= num; currentStep += 1) {
        [steps[0], steps[1]] = [steps[1], steps[0] + steps[1]];
    }
    return steps[1];
};
document.write(recursiveStaircase(10)+"<br/>");
document.write(recursiveStaircase(4)+"<br/>");

document.write(recursiveStaircase(13)+"<br/>");
</script>
</body>
</html>
```

