|  |
| --- |
| Exp.No:1 |
| Date:17/8/2022 |

**Basic Java Programs-I**

**Aim:**

To write java programs for the given basic problems.

**Programs:**

1.Fibonacci Series

import java.util.\*;  //used to include the all util packages includes Scanner for user input

public class fibonacci

{

      public static void main(String[] args)

      {

            System.out.println("Enter the value = :");

            Scanner obj=new Scanner(System.in);

            int n=obj.nextInt();

            System.out.println("Fibonacci Series:");

            int a=0,b=1,c;

            for(int i=0;i<n;++i)

            {

                  System.out.print(" "+a);

                  c=a+b;

                  a=b;

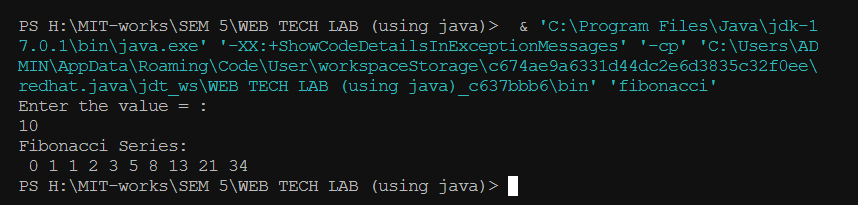
                  b=c;

            }

      }

}

Output:



2.Prime number:

import java.util.\*;

class prime

{

      int prime(int n)

      {

            for(int i=2;i<n/2;++i)

            { if(n%i==0){return 0;}}

            return 1;

      }

      public static void main(String[] args)

      {

            prime obj=new prime();

            Scanner in=new Scanner(System.in);

            System.out.print("Enter the number = ");

            int num=in.nextInt();

            int out=obj.prime(num);

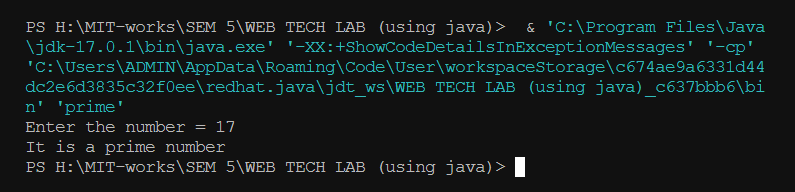
            if(out==0){System.out.println("Not a prime number");}

            else{System.out.println("It is a prime number");}

      }

}

Output:



3.Palindrome Check:

import java.util.\*;

public class palindrome

{

      boolean ispalin(int n){

            int rem,rev=0,i=0,temp=n;

            while(temp>0){

                  rem=temp%10;

                  rev=rev\*10+rem;

                  temp=temp/10;

            }

            boolean ans=(n==rev)?true:false;

            return ans;

      }

      public static void main(String[] args)

      {

            Scanner inp=new Scanner(System.in);

            palindrome obj= new palindrome();

            System.out.print("Enter the number = ");

            int n=inp.nextInt();

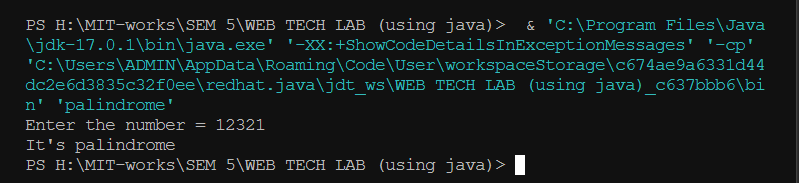
            if(obj.ispalin(n)){System.out.print("It's palindrome");}

            else{System.out.print("Not a palindrome");}

      }

}

Output:



4.Factorial:

import java.util.Scanner;

import java.util.\*;

class factorial

{

    int fact(int n)

    {

        int ans=1;

        for(int i=n;i>0;--i){ans\*=i;}

        return ans;

    }

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        factorial ob=new factorial();

        System.out.print("Enter the number = ");

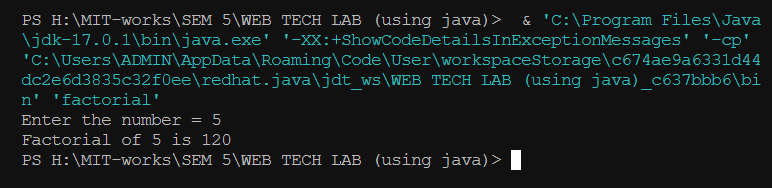
        int n=inp.nextInt();

        System.out.println("Factorial of "+n+" is "+ob.fact(n));

    }

}

Output:



5.Sum of divisible number:

class sum\_of\_integers

{

    public static void main(String[] args)

    {

        int sum=0;

        for(int i=100;i<=200;++i)

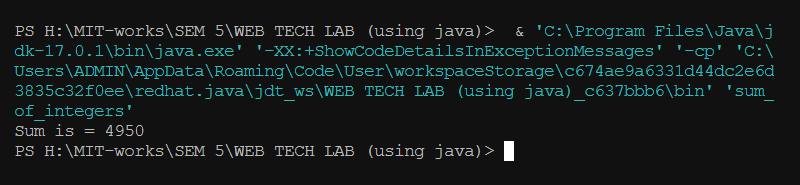
        {sum+=(i%3==0)?i:0;}

        System.out.println("Sum is = "+sum);

    }

}

Output:



6.Print Even number:

public class even\_number {

    public static void main(String[] args)

    {

        for(int i=0;i<=50;i=i+2)

        {

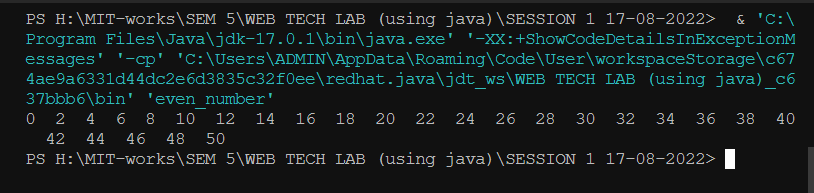
            System.out.println(i);

        }

    }

}

Output:



7.Student Details:

import java.util.Scanner;

public class student\_details {

    long roll\_no,mob;

    String name;

    void getdetails()

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the Name = ");

        name=inp.nextLine();

        System.out.print("Enter the Roll number = ");

        roll\_no=inp.nextLong();

        System.out.print("Enter the Mobile number = ");

        mob=inp.nextLong();

    }

    void display()

    {

        System.out.println("Name = "+name);

        System.out.println("Roll number = "+roll\_no);

        System.out.println("Mobile number = "+mob);

    }

    public static void main(String[] args)

    {

        student\_details obj=new student\_details();

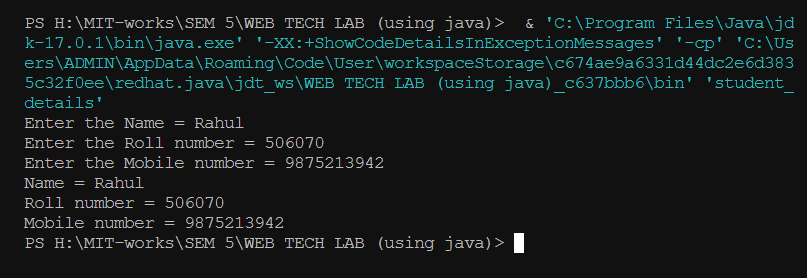
        obj.getdetails();

        obj.display();

    }

}

Output:



8.Bank Account:

import java.util.Scanner;

public class bank\_account{

    long acc\_no,mob;

    float balance=1000;

    String Name;

    Scanner inp=new Scanner(System.in);

    void getdetails()

    {

        System.out.print("Enter the Name = ");

        Name=inp.nextLine();

        System.out.print("Enter the Acc number = ");

        acc\_no=inp.nextLong();

        System.out.print("Enter the Mobile number = ");

        mob=inp.nextLong();

    }

    void display()

    {

        System.out.println("\n----------------------------");

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

        System.out.println("Account number = "+acc\_no);

        System.out.println("Mobile number = "+mob);

        System.out.println("Current balance = "+balance);

        System.out.println("----------------------------\n");

    }

    void credit()

    {

System.out.print("Enter the amount to deposit - ");

        balance+=inp.nextFloat();

}

    void debit()

    {

System.out.print("Enter the amount to withdraw - ");

        float amt=inp.nextFloat();

        if(amt<balance){ balance-=amt;}

        else{System.out.println("Insufficient Balance\n");}

}

    public static void main(String[] args)

    {

bank\_account obj=new bank\_account();

        obj.getdetails();

        while(true){

            System.out.print("Enter the choice = ");

            int ch=obj.inp.nextInt();

            switch(ch)

            {case 1:

                    obj.display();

                    break;

                case 2:

                    obj.credit();

                    break;

                case 3:

                    obj.debit();

                    break;

            }

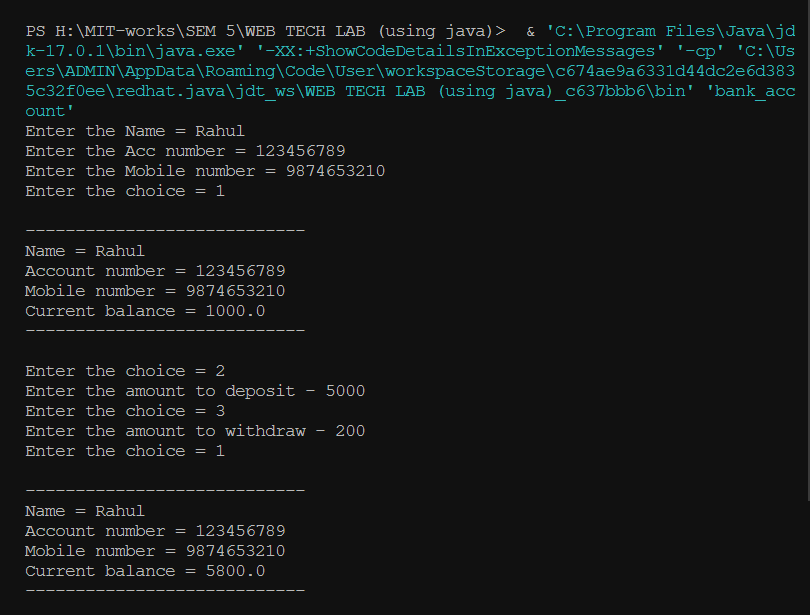
            if(ch==0) break;

        }

    }

}

Output:



Result:

Thus, the java programs for the given problems is completed and the output is displayed and verified.

|  |  |
| --- | --- |
| Exp.No:2 |  |
| Date:24/8/2022 |

|  |
| --- |
| **Basic Java Programs-II** |
|  |

Aim:

To write java programs for the given basic problems.

Programs:

1.Calculator:

import java.util.\*;

public class Calculator {

    public static void main(String[] args) {

        while(true)

        {

            Scanner inp=new Scanner(System.in);

            System.out.print("Enter the choice - ");

            int ch=inp.nextInt();

            if(ch==111) break;

            System.out.print("Enter the numebers = ");

            int a = inp.nextInt();

            int b=inp.nextInt();

            switch(ch)

            {

                case 1:

                    System.out.println("Addition of a b = "+(a+b));

                    break;

                case 2:

                    System.out.println("Subtraction of a b = "+(a-b));

                    break;

                case 3:

                    System.out.println("Multiplication of a b = "+(a\*b));

                    break;

                case 4:

                    System.out.println("Division of a b = "+(a/b));

                    break;

                case 5:

                    System.out.println("Square root of a and b = "+(Math.sqrt(a))+" and "+Math.sqrt(b));

                    break;

                case 6:

                    System.out.println("Power = "+Math.pow(a, b));

                    break;

                default:

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

                    break;

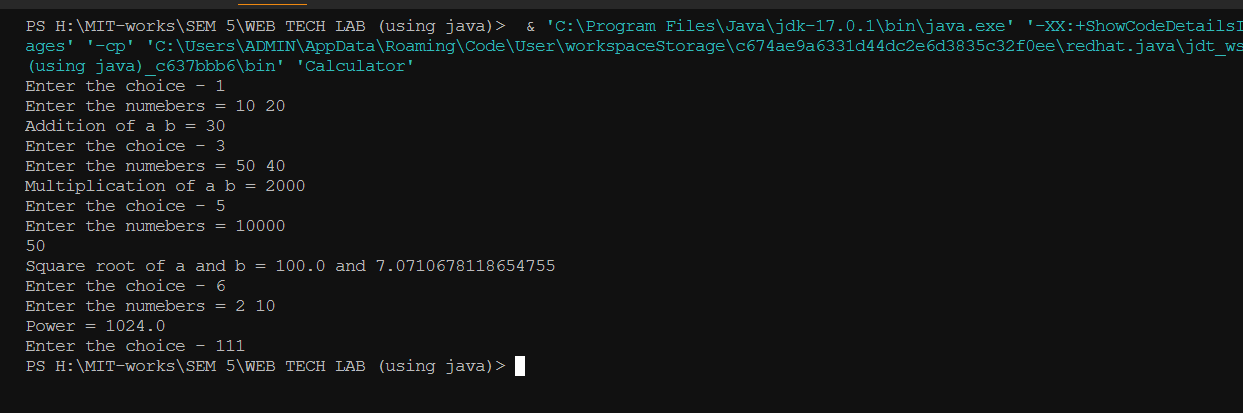
            }

        }

    }

}

Output:



2.Reversing Number:

import java.util.\*;

public class Reverse {

    int reverse(int n)

    {

        int rev=0;

        while(n>0)

        {

            rev=rev\*10+n%10;

            n/=10;

        }

        return rev;

    }

    public static void main(String[] args){

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the number - ");

        int num=inp.nextInt();

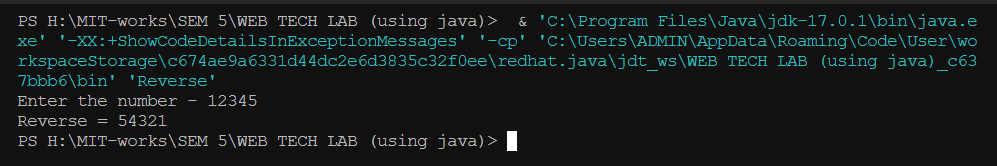
        Reverse obj=new Reverse();

        System.out.println("Reverse = "+obj.reverse(num));

    }

}

Output:



3.Factorial using Recursion:

import java.util.\*;

public class Factorial {

    int fact(int n)

    {

        if(n<2) return 1;

        else return n\*fact(n-1);

    }

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the number - ");

        int num=inp.nextInt();

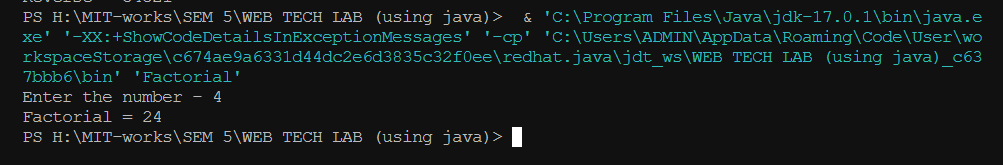
        Factorial obj=new Factorial();

        System.out.println("Factorial = "+obj.fact(num));

    }

}

Output:



4.Area Calculation:

import java.util.\*;

class calarea

{

    calarea(int a)

    {

        System.out.println("Area of a square = "+Math.pow(a, 2));

    }

    calarea(int l, int b)

    {

        System.out.println("Area of a rectangle = "+(l\*b));

    }

}

public class Area {

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the side - ");

        int a=inp.nextInt();

        calarea obj1 = new calarea(a);

        System.out.print("Enter the lenght and breadth - ");

        int l=inp.nextInt();

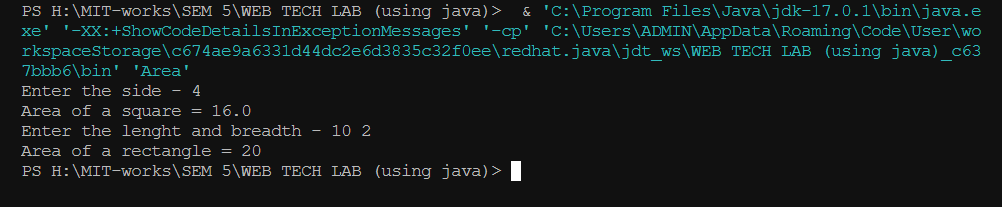
        int b=inp.nextInt();

        calarea obj2 = new calarea(l,b);

    }

}

Ouput:



5.Largest and Smallest:

import java.util.\*;

public class Largest\_and\_Smallest {

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        int arr[];

        System.out.println("Enter the number of elements - ");

        int n=inp.nextInt();

        arr=new int[n];

        for(int i=0;i<n;++i)

        {

            System.out.print("Enter the number - ");

            int num=inp.nextInt();

            arr[i]=num;

        }

        System.out.println("Array = "+Arrays.toString(arr));

        Arrays.sort(arr);

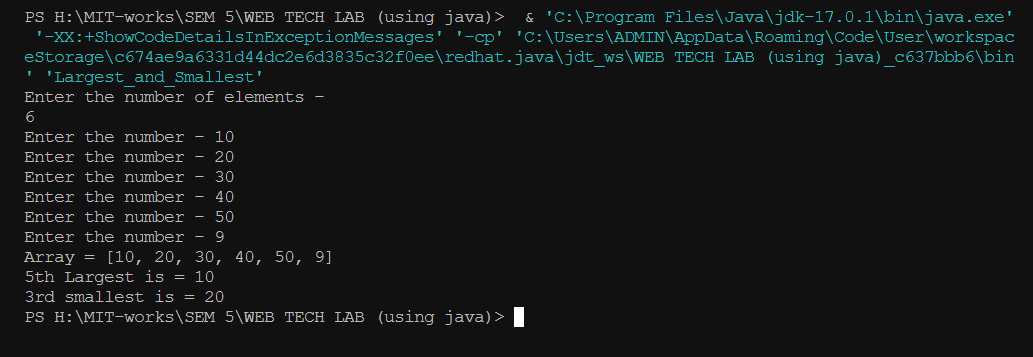
        System.out.println("5th Largest is = "+arr[n-5]);

        System.out.println("3rd smallest is = "+arr[2]);

    }

}

Output:



6.Matrix Operations:

import java.util.\*;

public class Matrix\_operations {

    void display(int A[][],int m,int n)

    {

        for(int i=0;i<m;++i)

        {

            for(int j=0;j<n;++j)

            {

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

            }

            System.out.println();

        }

    }

    void addition(int A[][],int B[][],int m,int n)

    {

        int ans[][]=new int[m][n];

        for(int i=0;i<m;++i)

        {

            for(int j=0;j<n;++j)

            {

                ans[i][j]=A[i][j]+B[i][j];

            }

        }

        Matrix\_operations obj=new Matrix\_operations();

        this.display(ans, m, n);

    }

    void multiplication(int A[][],int B[][],int m,int n)

    {

        int ans[][]=new int[m][n];

        for(int i=0;i<m;++i)

        {

            for(int j=0;j<n;++j)

            {

                for (int k=0;k<n;++k)

                {ans[i][j]+=A[i][k]\*B[k][j];}

            }

        }

        Matrix\_operations obj=new Matrix\_operations();

        this.display(ans, m, n);

    }

    void transpose(int A[][],int m,int n)

    {

        int ans[][]=new int[m][n];

        for(int i=0;i<m;++i)

        {

            for(int j=0;j<n;++j)

            {

                ans[i][j]=A[j][i];

            }

        }

        Matrix\_operations obj=new Matrix\_operations();

        this.display(ans, m, n);

    }

    int[][] getmatrix(int m,int n)

    {

        int ans[][]=new int[m][n];

        Scanner inp=new Scanner(System.in);

        for(int i=0;i<m;++i)

        {

            for(int j=0;j<n;++j)

            {

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

                int x=inp.nextInt();

                ans[i][j]=x;

            }

        }

        return ans;

    }

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        System.out.println("Enter the dimension of the matrix = ");

        int m=inp.nextInt();

        int n=inp.nextInt();

        Matrix\_operations obj=new Matrix\_operations();

        System.out.println("Matrx 1:");

        int A[][]=obj.getmatrix(m, n);

        System.out.println("Matrx 2:");

        int B[][]=obj.getmatrix(m, n);

        System.out.println("Addition = ");

        obj.addition(A, B, m, n);

        System.out.println("Multiplication = ");

        obj.multiplication(A, B, m, n);

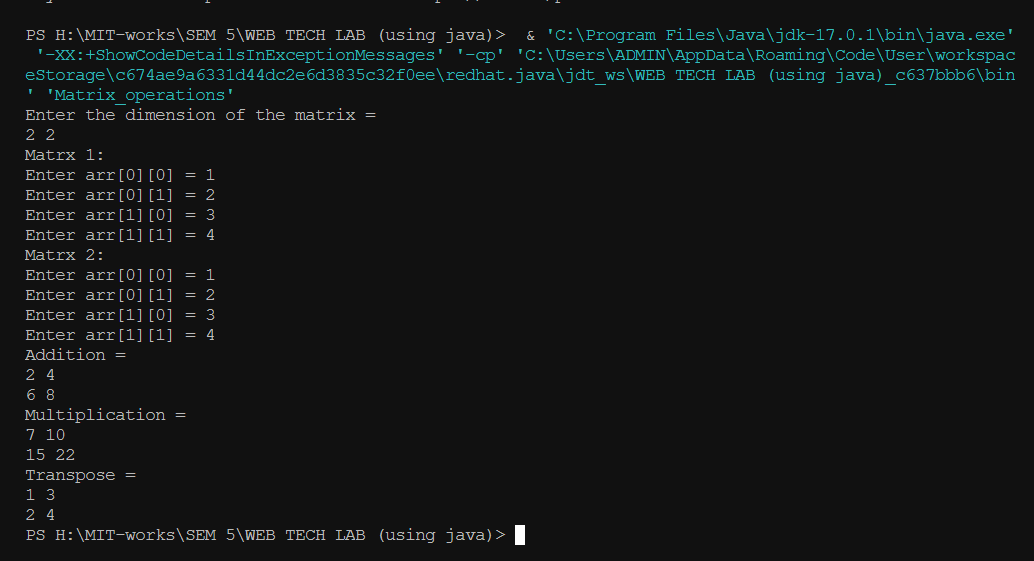
        System.out.println("Transpose = ");

        obj.transpose(A,m, n);

    }

}

Output:



7.Subset of an Array:

import java.util.\*;

public class Copyarray {

    int[] arrgen()

    {

        Scanner inp=new Scanner(System.in);

        int arr[];

        System.out.println("Enter the number of elements - ");

        int n=inp.nextInt();

        arr=new int[n];

        for(int i=0;i<n;++i)

        {

            System.out.print("Enter the number - ");

            int num=inp.nextInt();

            arr[i]=num;

        }

        return arr;

    }

    public static void main(String[] args)

    {

        Copyarray obj=new Copyarray();

        System.out.println("Array 1 = ");

        int arr1[]=obj.arrgen();

        System.out.println("Array 2 = ");

        int arr2[]=obj.arrgen();

        System.arraycopy(arr1, 1, arr2, 3, 3);

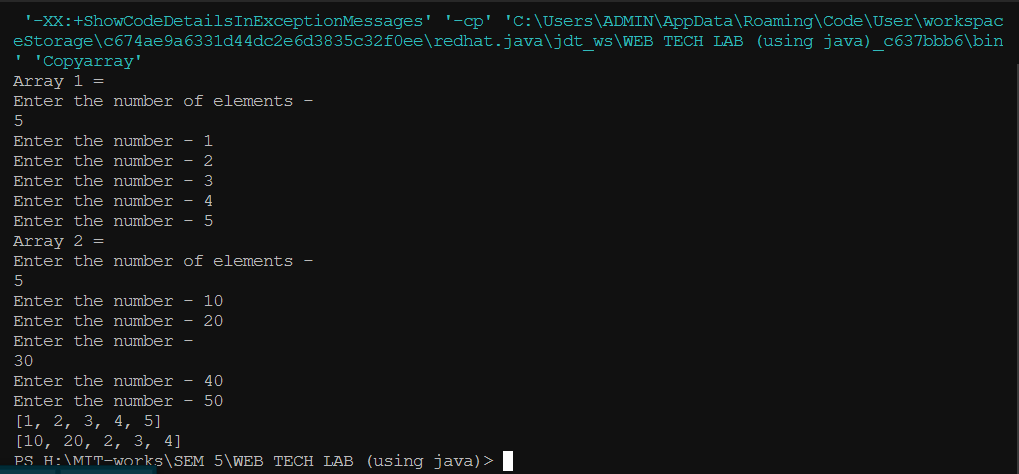
        System.out.println(Arrays.toString(arr1));

        System.out.println(Arrays.toString(arr2));

    }

}

Output:



8. Permutation and Combination

import java.util.\*;

public class Perm\_Comb {

    int fact(int n)

    {

        if(n<2) return 1;

        else return n\*fact(n-1);

    }

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("n= ");

        int n=inp.nextInt();

        System.out.print("r= ");

        int r=inp.nextInt();

        Perm\_Comb obj=new Perm\_Comb();

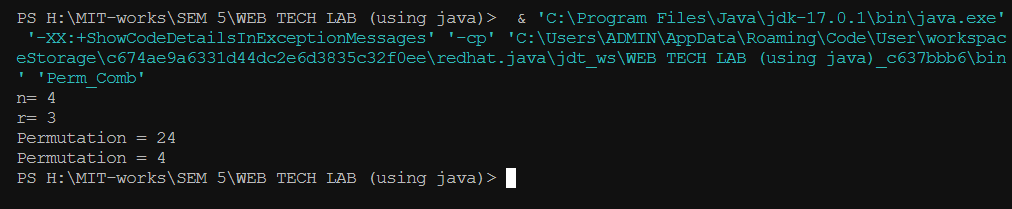
        System.out.println("Permutation = "+(obj.fact(n)/(obj.fact(n-r))));

        System.out.println("Permutation = "+(obj.fact(n)/(obj.fact(r)\*(obj.fact(n-r)))));

    }

}

Output:



9. Conditional operators

import java.util.\*;

public class Condition {

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the numbers - ");

        int a=inp.nextInt();

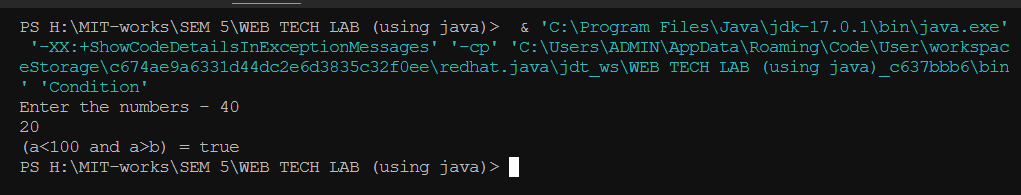
        int b=inp.nextInt();

        System.out.println("(a<100 and a>b) = "+((a<100 && a>b)==true));

    }

}

Output:



10.Library books:

import java.lang.reflect.Array;

import java.util.\*;

import javax.sound.sampled.SourceDataLine;

class Details {

      String name, author;

      int borrow;

      Details(String name, String author, int borrow) {

            this.name = name;

            this.author = author;

            this.borrow = borrow;

      }

      void display() {

            System.out.println("Name = " + name);

            System.out.println("Author = " + author);

            System.out.println("Borrow status = "+borrow);

      }

}

public class Library {

      void update(ArrayList<Details> arr,String name)

      {

            int siz=arr.size();

            for(int i=0;i<siz;++i)

            {

                  System.out.println(arr.get(i).name);

                  if((arr.get(i).name).equals(name)){

                        if(arr.get(i).borrow==0) arr.get(i).borrow=1;

                        else System.out.println("Already Book is borrowed");

                        return;

                  }

                  else{

                        System.out.println("Book is not available");

                        return;

                  }

            }

      }

      void List(ArrayList<Details> arr)

      {

            System.out.println("\n\nBorrowed Books:");

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

            {

                  if(arr.get(i).borrow==1) arr.get(i).display();

            }

            System.out.println("\n\nAvailable Books:");

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

                  if (arr.get(i).borrow == 0)

                        arr.get(i).display();

            }

      }

      public static void main(String[] args)

      {

            ArrayList<Details> arr = new ArrayList<Details>();

            System.out.println("Choice list: ");

            System.out.println("1.Insert a book \n2.Borrow a book\n3.Display");

            Library obj=new Library();

            while(true)

            {

                  Scanner inp = new Scanner(System.in);

                  System.out.print("\n\nEnter the choice - ");

                  int ch=inp.nextInt();

                  if(ch==111) break;

                  String name,author;

                  switch(ch)

                  {

                        case 1:

                              System.out.print("Name of the book = ");

                              name=inp.next();

                              System.out.print("Name of the author = ");

                              author=inp.next();

                              Details det=new Details(name,author,0);

                              arr.add(det);

                              System.out.println("New book added");

                              break;

                        case 2:

                              System.out.print("Name of the book = ");

                              String Name=inp.next();

                              obj.update(arr, Name);

                              break;

                        case 3:

                              obj.List(arr);

                              break;

                        default:

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

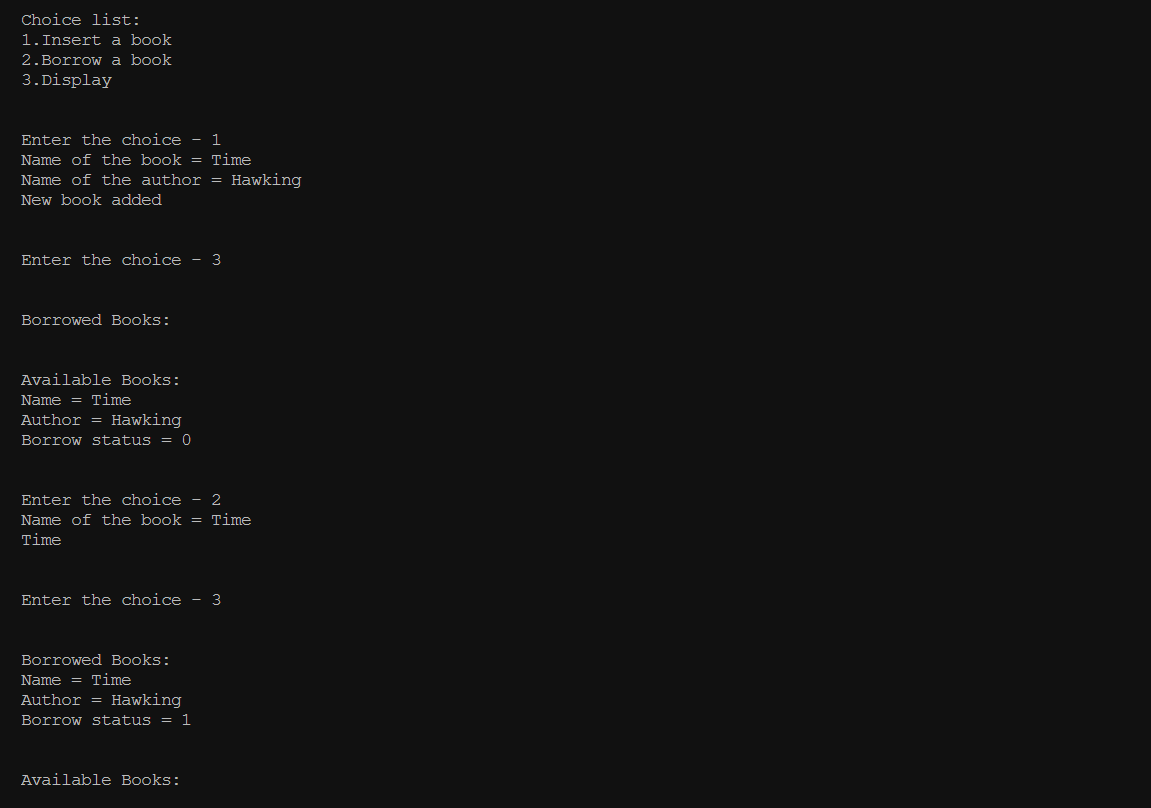
                  }

            }

      }

}

Output:



Result:

Thus, the java programs for the given problems is completed and the output is displayed and verified

|  |  |
| --- | --- |
| Exp.No:3 | Constructor, Method overloading, Array of objects |
| Date:7/9/2022 |

Aim:

To write java programs for the given problems using the given concepts.

Programs:

1.Parameterized Constructor

import java.util.\*;

class Circle

{

    float r;

    Circle(float r)

    {

        this.r=r;

        System.out.println("Perimeter of the circle  = "+2\*Math.PI\*r);

        System.out.println();

    }

}

class Rectangle

{

    float l,b;

    Rectangle(float l,float b)

    {

        this.l=l;

        this.b=b;

        System.out.println("Perimeter of the Rectangle  = "+2\*(l+b));

        System.out.println();

    }

}

class Square

{

    int a;

    Square(int a)

    {

        this.a=a;

        System.out.println("Perimeter of the Square  = "+4\*a);

        System.out.println();

    }

}

public class perimeter {

    public static void main(String[] args) {

        Scanner inp=new Scanner(System.in);

        //Circle

        System.out.print("Enter the radius of the circle = ");

        float r=inp.nextFloat();

        new Circle(r);

        //Rectangle

        System.out.print("Enter the length and breadth of the rectangle = ");

        float a=inp.nextFloat();

        float b=inp.nextFloat();

        new Rectangle(a,b);

        System.out.print("Enter the side of the square = ");

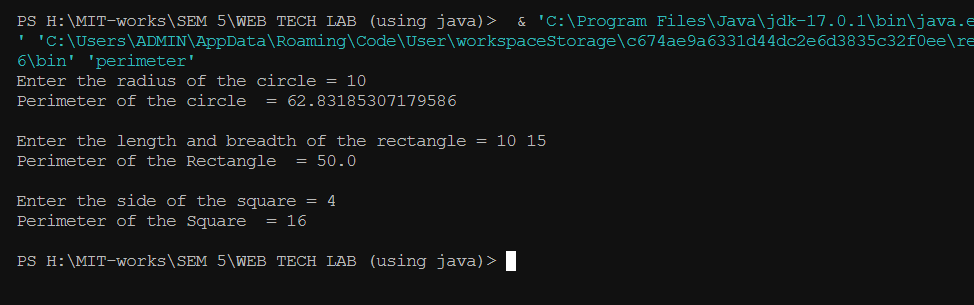
        int c=inp.nextInt();

        new Square(c);

    }

}

Output:



2.Method overloading:

import java.util.\*;

class Volume

{

    void vol(int a)

    {

        System.out.println("The volume of the Cube is = "+Math.pow(a, 3));

    }

    void vol(int r,int h)

    {

        System.out.println("The volume of the Cylinder is = "+(Math.PI\*Math.pow(r, 2)\*h));

    }

    void vol(float r)

    {

        System.out.println("The volume of the sphere is = "+(4/3)\*Math.PI\*Math.pow(r, 3));

    }

}

public class Volume\_cal {

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        int a,r,h;

        System.out.print("Enter the side of the cube = ");

        a=inp.nextInt();

        //cube

        new Volume().vol(a);

        System.out.print("Enter the radius and breadth of the Cylinder = ");

        r=inp.nextInt();

        h=inp.nextInt();

        //Cyliner

        new Volume().vol(r,h);

        System.out.print("Enter the side of the Sphere = ");

        float r2=inp.nextFloat();

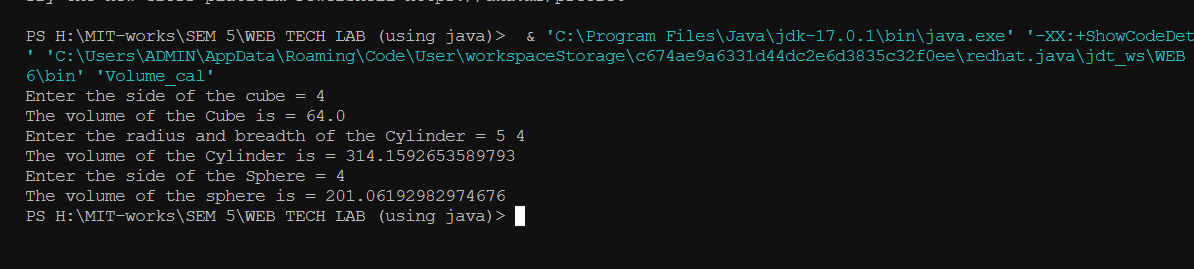
        //cube

        new Volume().vol(r2);

    }

}

Output:



3.Constructor Overloading:

import java.util.\*;

class Srt

{

    int[] arr=new int[5];

    Scanner inp=new Scanner(System.in);

    Srt()

    {

        for(int i=0;i<5;++i){

            System.out.print("Enter the value - ");

            this.arr[i]=inp.nextInt();

        }

    }

    Srt(int[] arr)

    {

        this.arr=arr;

    }

    void insertionsort()

    {

        for(int i=1;i<arr.length;++i)

        {

            int key=arr[i];

            int j=i-1;

            while(j>-1&&key<arr[j])

            {

                arr[j+1]=arr[j];

                j--;

            }

            arr[j+1]=key;

        }

        System.out.println(Arrays.toString(arr));

    }

}

public class Sorting {

    public static void main(String[] args)

    {

        new Srt().insertionsort();

        System.out.println("the above is by using Default Constructor = ");

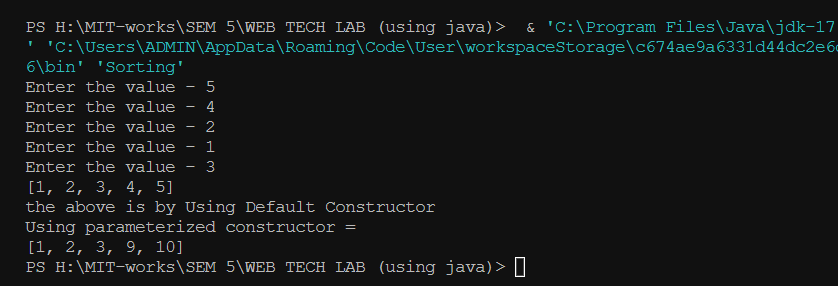
        System.out.println("Using parameterized constructor = ");

        new Srt(new int[]{10,9,1,2,3}).insertionsort();

    }

}

Output:



4.Money Box

import java.util.\*;

class AddAmount

{

    static float amount = 500;

    AddAmount()

    {

        System.out.println("No amount is deposited");

    }

    AddAmount(float x)

    {

        amount+=x;

        System.out.println("Amount is deposited");

    }

}

public class MoneyBx {

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        new AddAmount();

        System.out.print("Enter the amount = ");

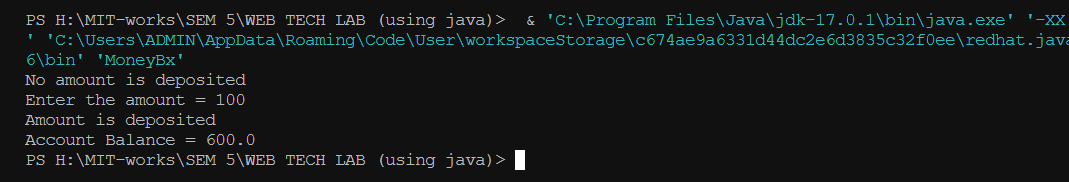
        new AddAmount(inp.nextFloat());

        System.out.println("Account Balance = "+AddAmount.amount);

    }

}

Output:



5. Different Sequence

class Print

{

    public void print(char x,int y)

    {

        System.out.println("Character = "+x);

        System.out.println("Integer = "+y);

    }

    public void print(int x,char y)

    {

        System.out.println("Integer = "+x);

        System.out.println("Characer = "+y);

    }

}

public class DiffSeq {

    public static void main(String[] args)

    {

        Print obj=new Print();

        obj.print('A',1);

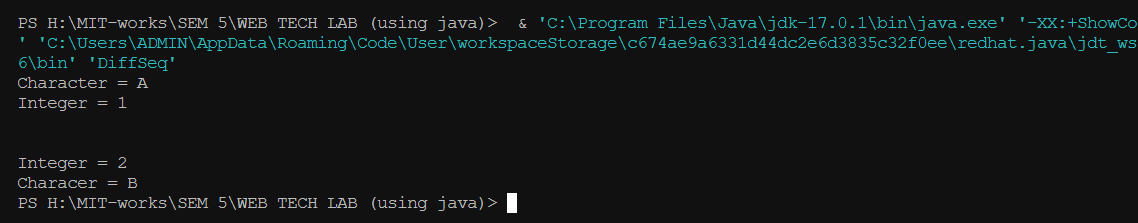
        System.out.println("\n");

        obj.print(2,'B');

    }

}

Output:



6. Array if objects:

import java.util.\*;

class Comp

{

    String name,date;

    float salary;

    Scanner inp=new Scanner(System.in);

    Comp()

    {

        System.out.print("Name = ");

        name=inp.nextLine();

        System.out.print("Salary = ");

        salary = inp.nextFloat();

        System.out.print("Date = ");

        inp.nextLine();

        date=inp.nextLine();

    }

    void display(int i)

    {

        System.out.println(i+".");

        System.out.println(" Name\t= "+name);

        System.out.println(" Salary  = "+salary);

        System.out.println(" Date of joining   = "+date);

        System.out.println();

    }

}

public class Company {

    public static void main(String[] args)

    {

        Comp obj[]=new Comp[10];

        for(int i=0;i<10;++i)

        {

            obj[i]=new Comp();

        }

        for(int i=0;i<10;++i)

        {

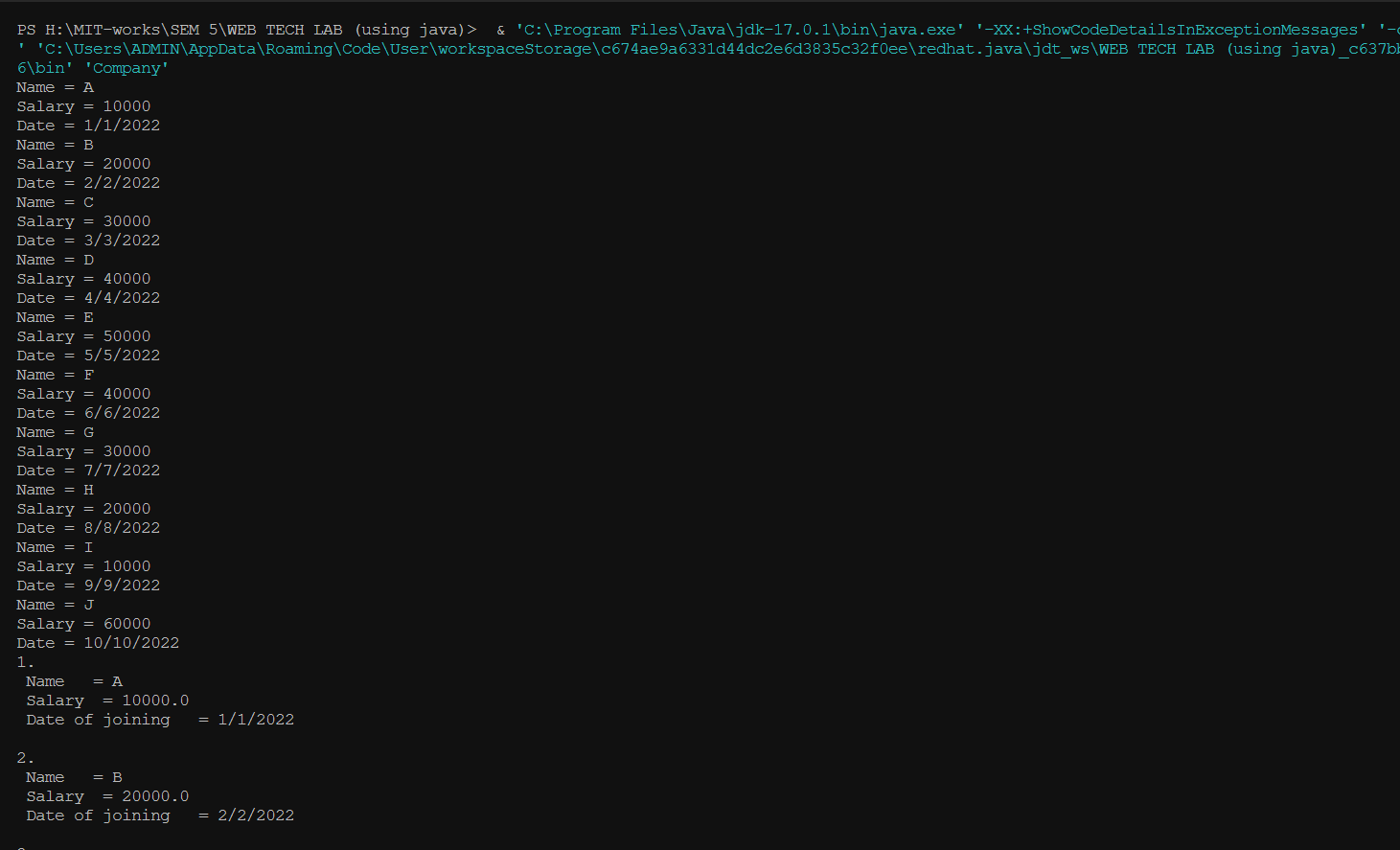
            obj[i].display(i+1);

        }

    }

}

Output:





Result:

Thus, the java programs for the given problems is completed and the output is displayed and verified.

|  |  |
| --- | --- |
| Exp.No:4 | **Static, This, Arrays, String Class** |
| Date:10/9/2022 |

Aim:

To write Java programs for the given problems with the given concepts.

Programs:

1.Static members:

import java.util.\*;

class Ari

{

    static int a;

    static int b;

    Ari()

    {

        System.out.println("Addition of a & b = "+(a+b));

        System.out.println("Subtraction of a & b = "+(a-b));

        System.out.println("Multiplication of a & b = "+(a\*b));

        System.out.println("Division of a & b = "+((float)a/b));

    }

}

public class Arithmetic {

    /\*\*

     \* @param args the command line arguments

     \*/

    public static void main(String[] args) {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the numbers a & b = ");

        Ari.a=inp.nextInt();

        Ari.b=inp.nextInt();

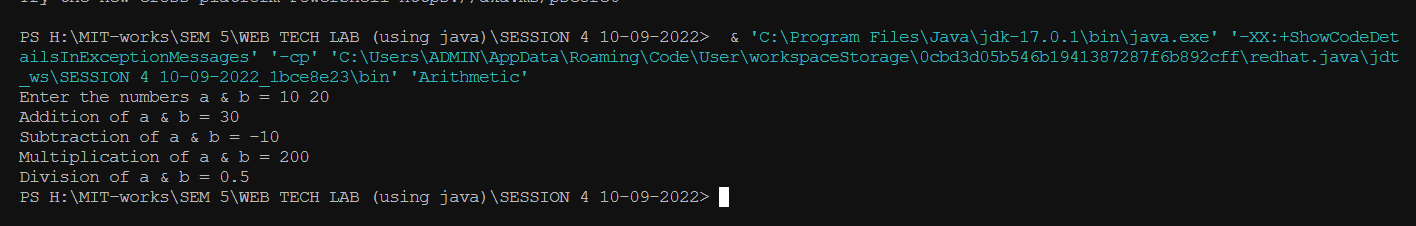
        //calling  the constructor

        new Ari();

    }

}

Output:



2.This keyword:

import java.util.\*;

class Bk

{

    long accno;

    static float balance=1000;

    String name;

    Scanner inp =new Scanner(System.in);

    Bk(String name,long accno)

    {

        this.name=name;

        this.accno=accno;

    }

    void credit()

    {

            System.out.print("Enter the amount to deposit = ");

            float x=inp.nextFloat();

            this.balance+=x;

            System.out.println("Amount Deposited!\nCurrent balance - "+this.balance);

    }

    void debit()

    {

            System.out.print("Enter the amount to withdraw = ");

            float x=inp.nextFloat();

            this.balance-=x;

            System.out.println("Amount Debited!\nCurrent balance - "+this.balance);

    }

    void display()

    {

        System.out.println("Name = "+name);

        System.out.println("Acc no = "+accno);

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

    }

}

public class Bank {

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        System.out.println("Enter the name = ");

        String nm=inp.nextLine();

        System.out.println("Enter the Acc no = ");

        long acc=inp.nextLong();

        Bk obj=new Bk(nm,acc);

        obj.display();

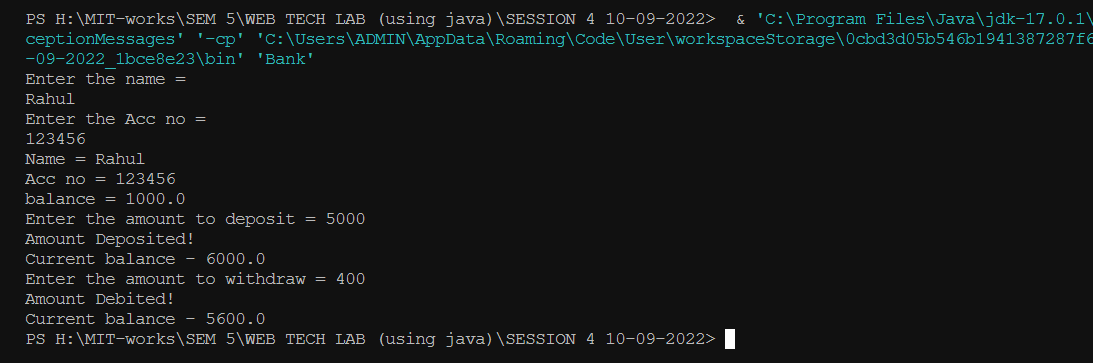
        obj.credit();

        obj.debit();

    }

}

Output:



3.Arrays:

import java.util.\*;

public class Arr\_manipulation {

    public static void main(String[] args){

        int Arr[]=new int[10];

        Scanner inp=new Scanner(System.in);

        for(int i=0;i<10;++i)

        {

            System.out.print("Mark = ");

            Arr[i]=inp.nextInt();

        }

        //methods

        Arrays.sort(Arr);

        System.out.println("Array = "+Arrays.toString(Arr));

        Arrays.fill(Arr,0,5,100);

        System.out.println("Array = "+Arrays.toString(Arr));

        System.out.print("Enter the number to search = ");

        Arrays.sort(Arr);

        System.out.println(Arrays.binarySearch(Arr,0,Arr.length, inp.nextInt()));

        System.out.println("Array = "+Arrays.toString(Arr));

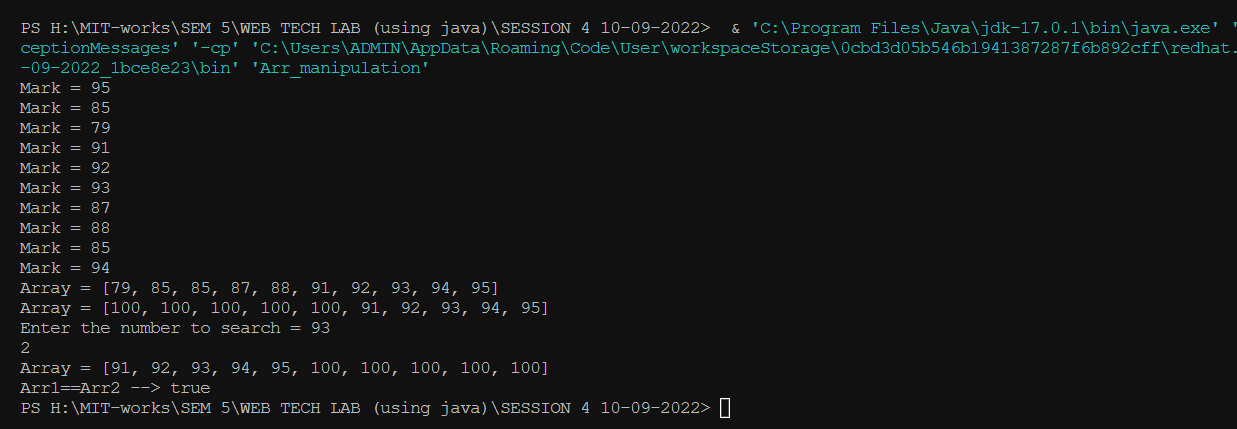
        int Arr2[]=Arr;

        System.out.println("Arr1==Arr2 --> "+Arrays.equals(Arr, Arr2));

    }

}

Output:



4.Complex Numbers:

import java.util.\*;

class Comp

{

    int real,imag;

    Comp(int real,int img)

    {

        this.real=real;

        this.imag=img;

    }

}

public class Complex {

    void add(Comp c1,Comp c2)

    {

        if(c1.imag+c2.imag <0) System.out.print((c1.real+c2.real)+""+(c1.imag+c2.imag)+"i");

        else System.out.print((c1.real+c2.real)+"+"+(c1.imag+c2.imag)+"i");

    }

    void sub(Comp c1,Comp c2)

    {

        if(c1.imag-c2.imag <0) System.out.println((c1.real-c2.real)+""+(c1.imag-c2.imag)+"i");

        else System.out.println((c1.real-c2.real)+"+"+(c1.imag-c2.imag)+"i");

    }

    void mult(Comp c1,Comp c2)

    {

        if((c2.real\*c1.imag+c2.imag\*c1.real)<0)

        {

System.out.println((c2.real\*c1.real + (-1\*(c1.imag\*c2.imag))) +""+ (c2.real\*c1.imag+c2.imag\*c1.real) +"i");

        }

        else

        {

            System.out.println((c2.real\*c1.real + (-1\*(c1.imag\*c2.imag))) + "+" + (c2.real\*c1.imag+c2.imag\*c1.real) + "i");

        }

    }

    public static void main(String[] args)

    {

        Comp c1=new Comp(1,-2);

        Comp c2=new Comp(1,2);

        Complex obj=new Complex();

        System.out.println("Addition = ");

        obj.add(c1,c2);

        System.out.println("Subtraction = ");

        obj.sub(c1,c2);

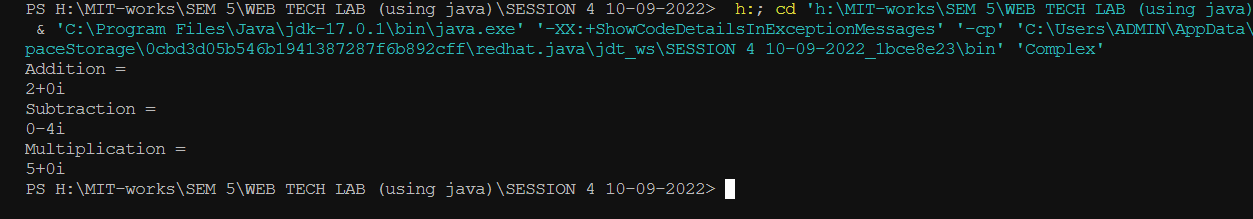
        System.out.println("Multiplication = ");

        obj.mult(c1,c2);

    }

}

Output:



5.Library:

import java.lang.reflect.Array;

import java.util.\*;

import javax.sound.sampled.SourceDataLine;

class Details {

      String name, author;

      static int borrow=0;

      Details(String name, String author) {

            this.name = name;

            this.author = author;

            //this.borrow = borrow;

      }

      void display() {

            System.out.println("Name = " + name);

            System.out.println("Author = " + author);

            //System.out.println("Borrow status = "+borrow);

      }

}

public class Library {

      static void update(ArrayList<Details> arr,String name)

      {

            int siz=arr.size();

            for(int i=0;i<siz;++i)

            {

                  if((arr.get(i).name).equals(name)){

                        if(arr.get(i).borrow==0) {arr.get(i).borrow=1;System.out.println("Book is borrowed\n");}

                        else System.out.println("Already Book is borrowed");

                        return;

                  }

                  else{

                        System.out.println("Book is not available");

                        return;

                  }

            }

      }

      static void List(ArrayList<Details> arr)

      {

            System.out.println("\n\nBorrowed Books:");

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

            {

                  if(arr.get(i).borrow==1) arr.get(i).display();

            }

            System.out.println("\n\nAvailable Books:");

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

                  if (arr.get(i).borrow == 0)

                        arr.get(i).display();

            }

      }

      static int search(ArrayList<Details> arr,String name)

      {

            int n=arr.size();

            for(int i=0;i<n;++i)

            {

                  if((arr.get(i).name).equals(name)) return 1;

            }

            return 0;

      }

      public static void main(String[] args)

      {

            ArrayList<Details> arr = new ArrayList<Details>();

            System.out.println("Choice list: ");

            System.out.println("1.Insert a book \n2.Borrow a book\n3.Display\n4.Search");

            Library obj=new Library();

            while(true)

            {

                  Scanner inp = new Scanner(System.in);

                  System.out.print("\n\nEnter the choice - ");

                  int ch=inp.nextInt();

                  if(ch==111) break;

                  String name,author;

                  switch(ch)

                  {

                        case 1:

                              System.out.print("Name of the book = ");

                              name=inp.next();

                              System.out.print("Name of the author = ");

                              author=inp.next();

                              Details det=new Details(name,author);

                              arr.add(det);

                              System.out.println("New book added");

                              break;

                        case 2:

                              System.out.print("Name of the book = ");

                              String Name=inp.next();

                              Library.update(arr, Name);

                              break;

                        case 3:

                              Library.List(arr);

                              break;

                        case 4:

                              System.out.print("Name of the book = ");

                              String Nam=inp.next();

                              if(Library.search(arr, Nam)==1) System.out.println("\nBook is present\n");

                              else System.out.println("\nNo book is present\n");

                              break;

                        default:

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

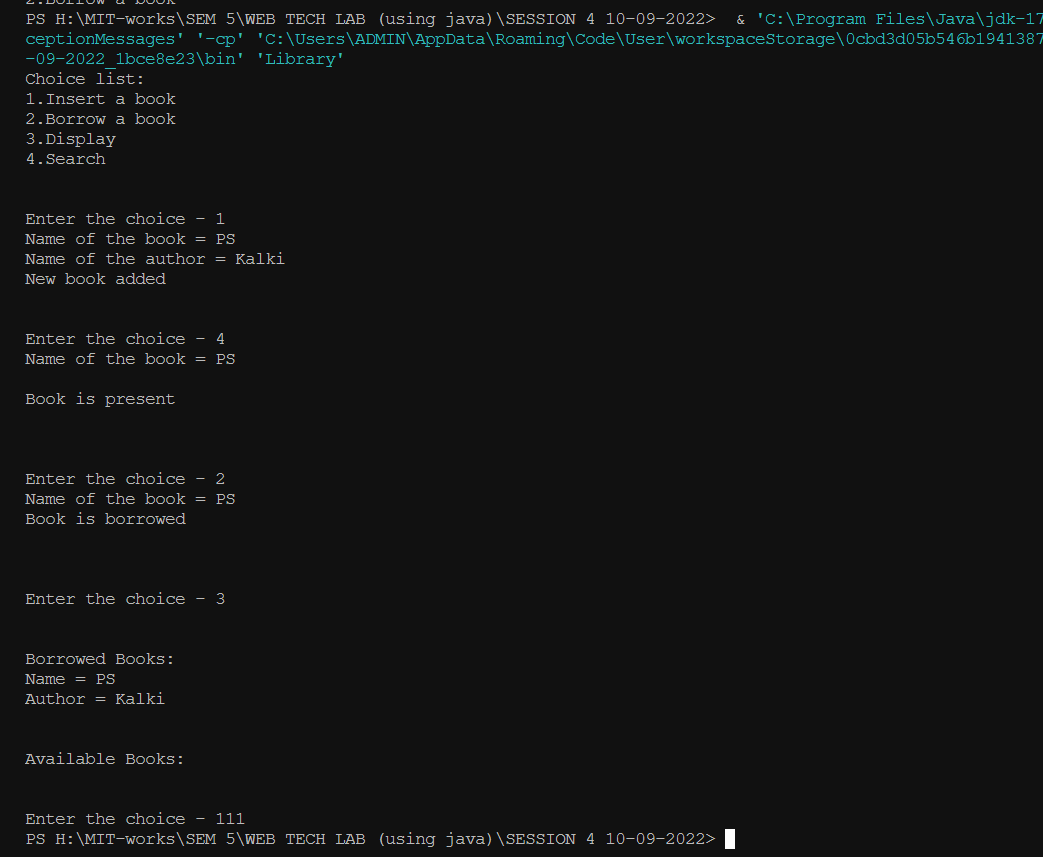
                  }

            }

      }

}

Output:



6.String Object:

import java.util.\*;

public class Str\_man {

    public static void main(String[] args)

    {

        String name = new String("Rahul Prasanth D");

        name=name.toLowerCase();

        System.out.println("Length of name  = "+name.length());

        System.out.println(name);

        int cnt=0;

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

        {

            if(name.charAt(i)==name.charAt(0)) cnt++;

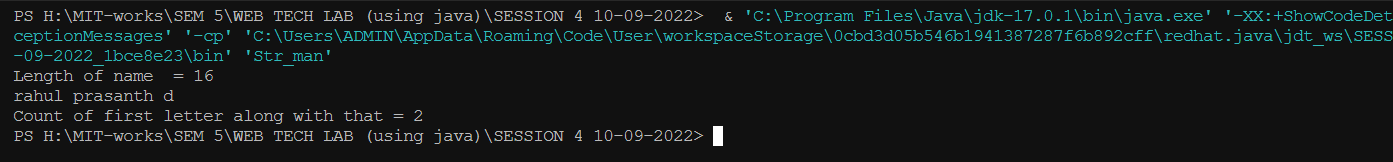
        }

        System.out.println("Count of first letter along with that = "+cnt);

    }

}

Output:



7. String Handling

import java.util.\*;

public class StringHandling {

    public static void main(String[] args)

    {

        StringBuilder str=new StringBuilder("Rahul ");

        System.out.println("Capacity = "+str.capacity());

        System.out.println("Lowercase = "+(str.toString()).toLowerCase());

        System.out.println("Upper case = "+(str.toString()).toUpperCase());

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the string to append = ");

        str.append(inp.nextLine());

        System.out.println("New String = "+str);

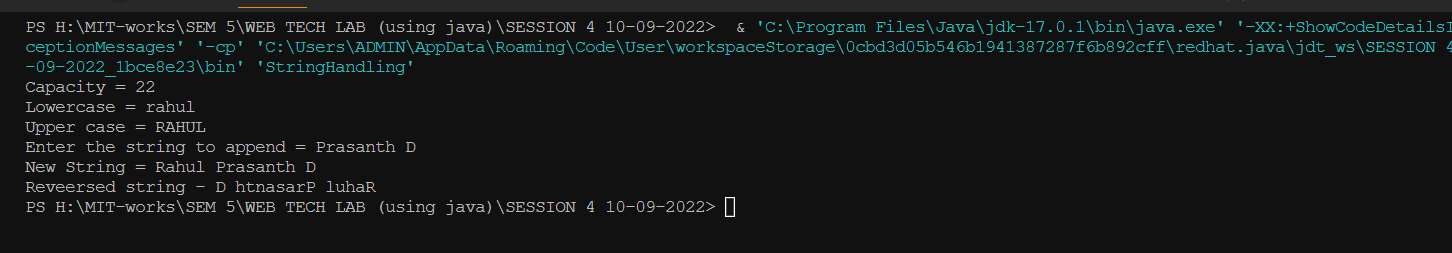
        StringBuilder x=str.reverse();

        System.out.println("Reveersed string - "+str);

    }

}

Output:



Result:

Thus, the Java programs for the given problems is completed and the output is displayed and verified.

|  |  |
| --- | --- |
| Exp.No:5 | **Inheritance, Super, Final, ArrayList** |
| Date:14/9/2022 |

Aim:

To write Java programs for the java problems using the given concepts.

Programs:

1. Access Specifiers:

import java.util.\*;

class Member

{

    public String name,address;

    protected int mobile,age,salary;

    Member()

    {

        Scanner inp =new Scanner(System.in);

        System.out.print("Name = ");

        this.name=inp.next();

        System.out.print("Age and mobile number = ");

        this.age=inp.nextInt();

        this.mobile=inp.nextInt();

        System.out.print("Salary = ");

        this.salary=inp.nextInt();

        System.out.print("Address = ");

        this.address=inp.next();

    }

    void display()

    {

        System.out.println("Name = "+name+"\nSalary = "+salary);

    }

}

class Employee extends Member

{

    String spec,dep;

    public Employee(String spec,String dep){

        this.spec=spec;

        this.dep=dep;

    }

    void Assign()

    {

        //Member obj=new Member();

        System.out.println("\n\nEmployee Details ----------> ");

        super.display();

        System.out.println("Department = "+dep);

        System.out.println("Specialization = "+spec);

    }

}

class Manager extends Member

{

    String spec,dep;

    public Manager(String spec,String dep){

        this.spec=spec;

        this.dep=dep;

    }

    void Assign()

    {

        //Member obj=new Member();

        System.out.println("\n\nManager Details ------------> ");

        super.display();

        System.out.println("Department = "+dep);

        System.out.println("Specialization = "+spec);

    }

}

public class Salary {

    public static void main(String[] args) {

        String spec,dep;

       Scanner inp=new Scanner(System.in);

        //Emp

        System.out.print("Enter the Employee Specialization and department = ");

        spec=inp.nextLine();

        dep=inp.nextLine();

        Employee emp=new Employee(spec,dep);

        emp.Assign();

         //manager

        System.out.print("Enter the Manager Specialization and department = ");

        spec=inp.next();

        dep=inp.next();

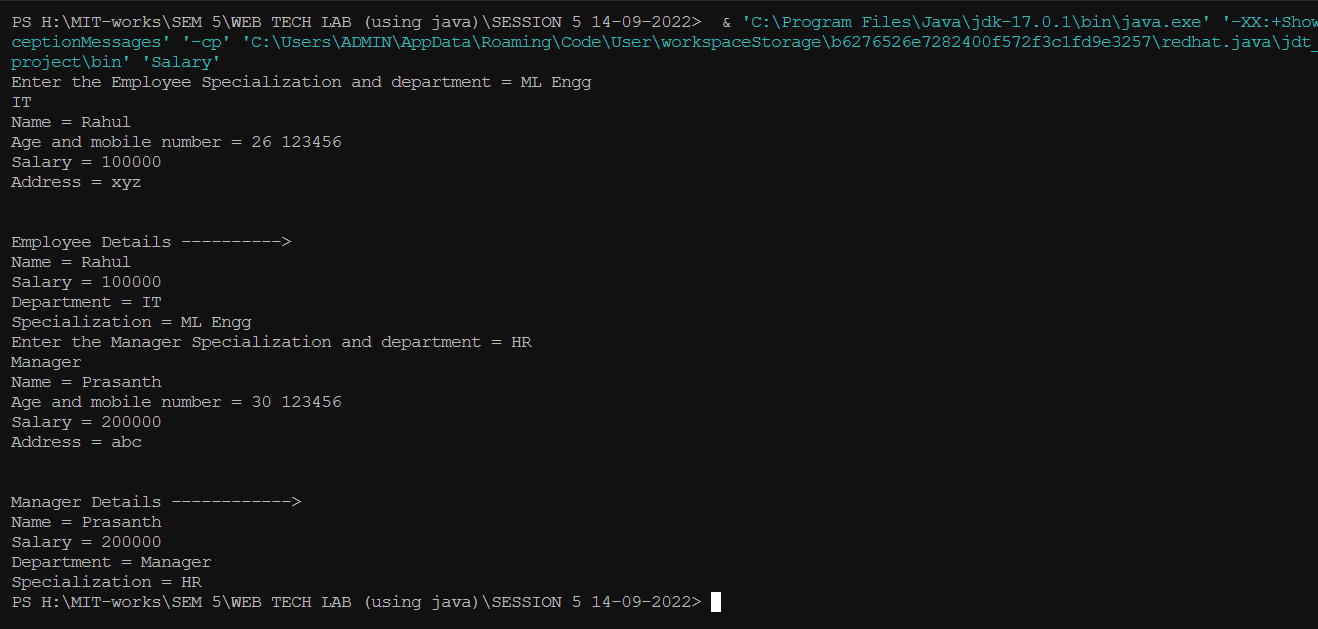
        Manager man=new Manager(spec,dep);

        man.Assign();

    }

}

Output:



1. Hierarchical Inheritance:

import java.util.\*;

class Country

{

    String country;

    void assign()

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the country name = ");

        this.country=inp.nextLine();

        System.out.println("Country  = "+country);

    }

}

class State extends Country

{

    String state;

    State()

    {

        super.assign();

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the State name = ");

        this.state=inp.nextLine();

    }

}

class City extends Country

{

    String city;

    City()

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the City name = ");

        this.city=inp.nextLine();

    }

}

class Village extends Country

{

    String vill;

    Village()

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the Village name = ");

        this.vill=inp.nextLine();

    }

}

public class Hierarchial {

    public static void main(String[] args)

    {

        System.out.println("State = "+new State().state);

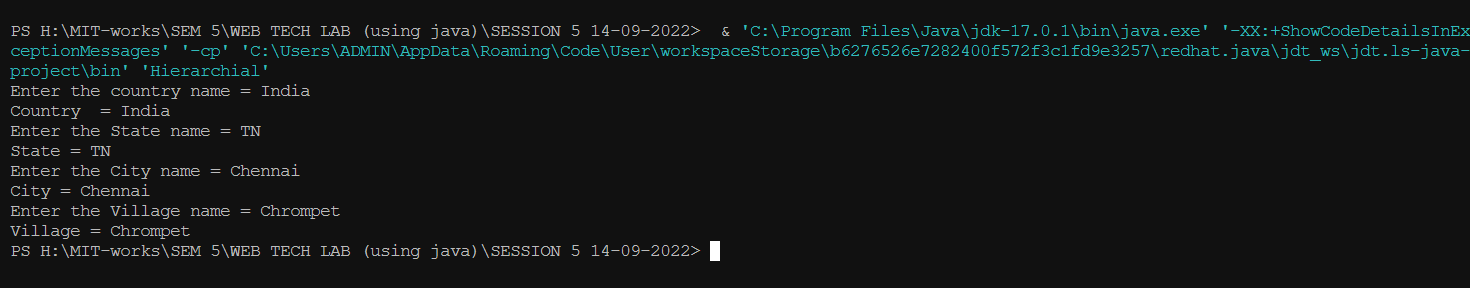
        System.out.println("City = "+new City().city);

        System.out.println("Village = "+new Village().vill);

    }

}

Output:



1. Vehicle:

import java.util.\*;

class Vehicle

{

    String type;

    int weight,power,wheels;

    void assignVehicle()

    {

        Scanner inp =new Scanner(System.in);

        System.out.print("Type of the vehicle = ");

        this.type=inp.nextLine();

        System.out.print("Weight of the vehicle(kgs) = ");

        this.weight=inp.nextInt();

        System.out.print("Power of the vehicle = ");

        this.power=inp.nextInt();

        System.out.print("No of wheels on the vehicle = ");

        this.wheels=inp.nextInt();

    }

    void display()

    {

        System.out.println("\n\nType of the vehicle = "+type);

        System.out.println("Weight of the vehicle(kgs) = "+weight);

        System.out.println("Power of the vehicle = "+power);

        System.out.println("No of wheels on the vehicle = "+wheels+"\n\n");

    }

}

class Twowheel extends Vehicle

{

    Twowheel()

    {

        System.out.println("\nTwo wheel\n");

        super.assignVehicle();

        //super.display();

    }

}

class Threewheel extends Vehicle

{

    Threewheel()

    {

        System.out.println("\nThree wheel\n");

        super.assignVehicle();

        //super.display();

    }

}

class Fourwheel extends Vehicle

{

    Fourwheel()

    {

        System.out.println("\nFour wheel\n");

        super.assignVehicle();

        //super.display();

    }

}

public class Vehicles {

    public static void main(String[] args)

    {

        Twowheel obj1 = new Twowheel();

        Threewheel obj2 = new Threewheel();

        Fourwheel obj3 = new Fourwheel();

        obj1.display();

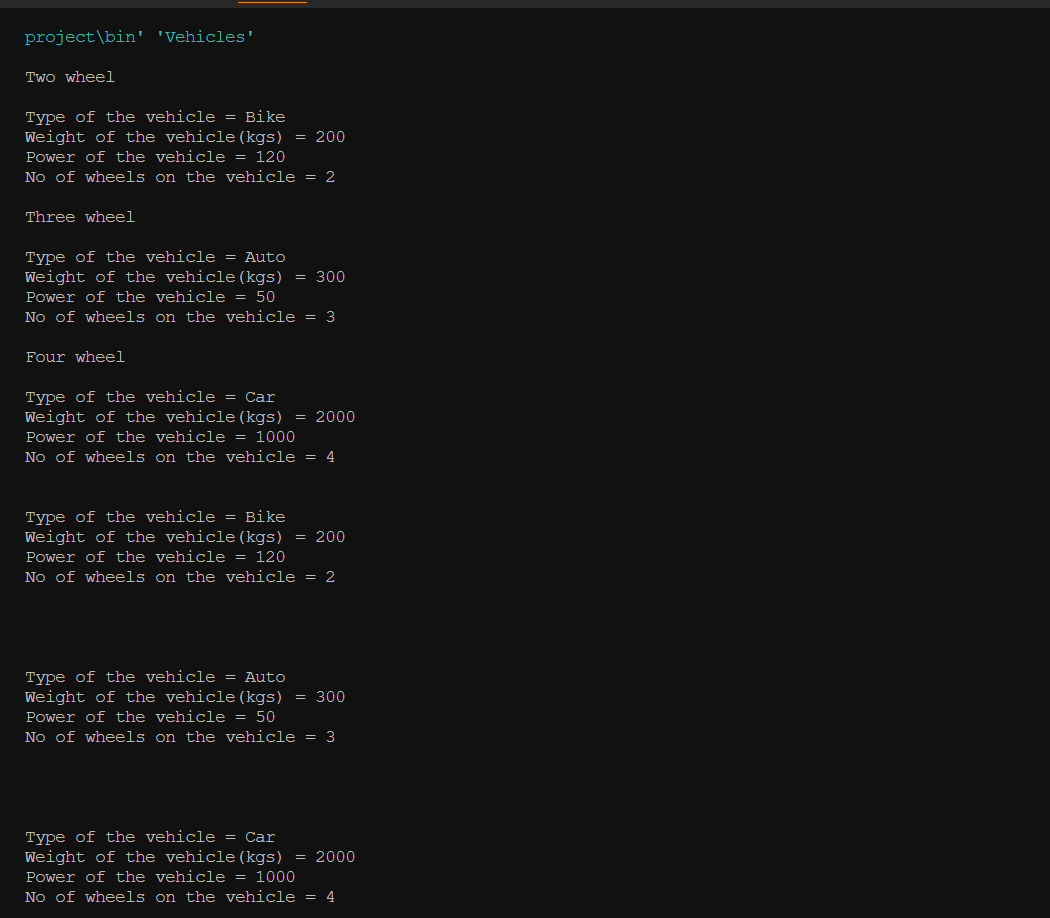
        obj2.display();

        obj3.display();

    }

}

Output:



4.Train:

import java.util.\*;

class Train\_journey

{

    static String start,terminating;

    int journey\_time;

    void Set\_start(String start){ this.start=start;}

    void Get\_start(String termination){this.terminating=termination;}

    void Get\_journey\_time()

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the total journey time = ");

        this.journey\_time=inp.nextInt();

    }

}

class Freight extends Train\_journey

{

    final int capacity=200;int carriages;

    boolean hazard;

    Scanner inp=new Scanner(System.in);

    void getParent()

    {

        String start,end;

        System.out.print("Enter the starting point - ");

        start=inp.nextLine();

        System.out.print("Enter the ending point - ");

        end=inp.nextLine();

        super.Set\_start(start);

        super.Get\_start(end);

        super.Get\_journey\_time();

    }

    void set\_hazard()

    {

        System.out.print("Is it hazardous? true/false= ");

        this.hazard=inp.nextBoolean();

        if(hazard==true) System.out.println("Hazardous materials are not allowed\n");

    }

    void get\_capacity()

    {

        System.out.print("capacity = "+capacity+" tonnes");

    }

}

class Passenger extends Train\_journey

{

    int first\_class,carriages;

    boolean catering;

    Scanner inp=new Scanner(System.in);

    void get\_first\_class()

    {

        System.out.print("Enter the first class number = ");

        this.first\_class=inp.nextInt();

    }

    void addCarraiges(int x)

    {

        if(carriages<x) System.out.println("You can't carry more than the capacity\n");

        this.carriages=x;

    }

    void In\_service()

    {

        System.out.print("Whether the catering services are available true/false");

        this.catering=inp.nextBoolean();

    }

}

public class Train {

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        Freight frt=new Freight();

        Passenger psr=new Passenger();

        frt.getParent();

        System.out.println("\nFREIGHT DETIALS->\n");

        frt.set\_hazard();

        frt.get\_capacity();

        System.out.println("\n\nPASSENGER DETIALS->\n");

        psr.get\_first\_class();

        System.out.print("Enter the total number of carriages = ");

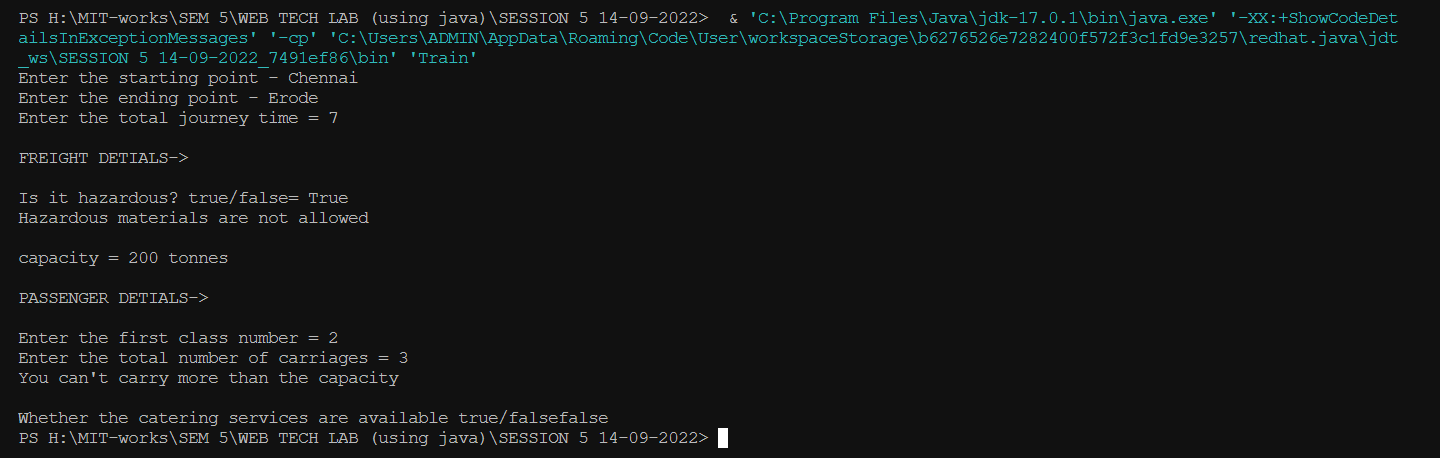
        psr.addCarraiges(inp.nextInt());

        psr.In\_service();

    }

}

Output:



5.ArrayList with Student details:

import java.util.\*;

class Details

{

    String name,dep;

    int gpa;

    void getDetails()

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the name = ");

        this.name=inp.next();

        System.out.print("Enter the department = ");

        this.dep=inp.next();

        System.out.print("Enter the GPA= ");

        this.gpa=inp.nextInt();

    }

}

public class Student {

    void delete(ArrayList<Details> arr,String name)

    {

        for(int i=0;i<arr.size();++i)

        {

            if((arr.get(i).name).equals(name))

            {

                arr.remove(i);

                System.out.println("\nStudent successfully removed||\n");

            }

        }

    }

    void display(ArrayList<Details> arr)

    {

        for(int i=0;i<arr.size();++i)

        {

            System.out.println("Name = "+arr.get(i).name);

            System.out.println("Department = "+arr.get(i).dep);

            System.out.println("gpa = "+arr.get(i).gpa);

            System.out.println("\n");

        }

    }

    ArrayList<Details> groupadd()

    {

        Scanner inp=new Scanner(System.in);

        ArrayList<Details> arr2=new ArrayList<Details>();

        System.out.println("Enter the number of students= ");

        int n=inp.nextInt();

        for(int i=0;i<n;++i)

        {

            Details obj=new Details();

            obj.getDetails();

            arr2.add(obj);

        }

        return arr2;

    }

    void find(ArrayList<Details> arr,String name)

    {

        int flag=0;

        for(int i=0;i<arr.size();++i)

        {

            if((arr.get(i).name).equals(name))

            {

                flag=1;

                System.out.println("\nStudent is present\n");

                return;

            }

        }

        if(flag==0) System.out.println("Student is not in the list\n");

    }

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        ArrayList<Details> arr=new ArrayList<Details>();

        System.out.println("Choice List");

        System.out.println("1.Insert a student\n2.Insert a group of students\n3.remove a student\n4.find \n5.Display\n111 to exit");

        while(true)

        {

            System.out.print("Enter the choice = ");

            int ch=inp.nextInt();

            if(ch==111) break;

            Student own=new Student();

            switch(ch)

            {

                case 1:

                    Details obj=new Details();

                    obj.getDetails();

                    arr.add(obj);

                    System.out.println("Student Added succesfully");

                    break;

                case 2:

                    arr.addAll(own.groupadd());

                    System.out.println("All are added Succssfuly");

                    break;

                case 3:

                      System.out.print("Enter the name = ");

                     own.delete(arr, inp.next());

                     break;

                case 4:

                    System.out.print("Enter the name to search = ");

                     own.find(arr, inp.next());

                     break;

                case 5:

                        own.display(arr);break;

                default:

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

                        break;

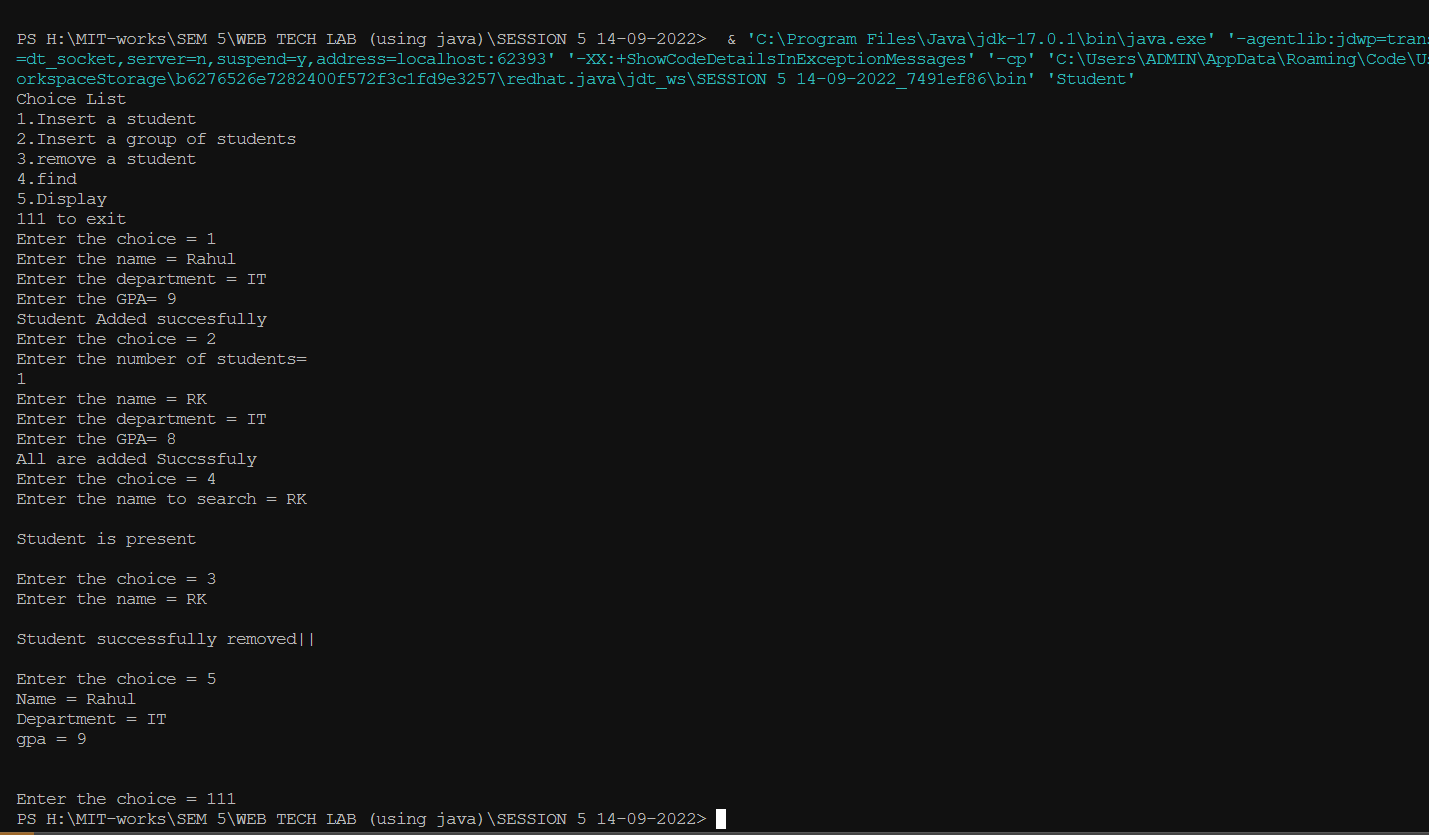
            }

        }

    }

}

Output:



6. String Operations to remove the string ends with ‘s’:

import java.lang.reflect.Array;

import java.util.\*;

import javax.sound.sampled.SourceDataLine;

class ArrayList\_string {

      void rev(List<String> arr)

      {

            Collections.reverse(arr);

      }

      void capitalize(List<String> arr)

      {

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

            {

                  String str=arr.get(i);

                  if(str.charAt(str.length()-1)=='S'|| str.charAt(str.length() - 1) == 's')

                  {

                        arr.remove(i);

                  }

            }

      }

      public static void main(String[] args)

      {

            Scanner inp=new Scanner(System.in);

            System.out.print("Enter the number of strings = ");

            int n=inp.nextInt();

            List<String> arr=new ArrayList<String>();

            for(int i=0;i<n;++i)

            {

                  System.out.print("Enter the string = ");

                  arr.add(inp.next());

            }

            System.out.println(arr);

            ArrayList\_string obj=new ArrayList\_string();

            obj.rev(arr);

            System.out.println(arr);

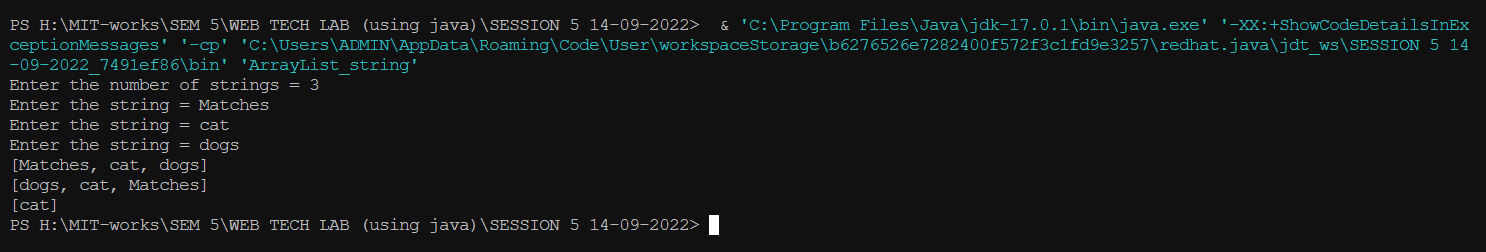
            obj.capitalize(arr);

            System.out.println(arr);

      }

}

Output:



7. ArrayList Operations

import java.util.\*;

public class Arraylist {

    public static void main(String[] args)

    {

        ArrayList<Integer> arr=new ArrayList<>();

        Scanner inp = new Scanner(System.in);

        System.out.print("Enter the number of elements = ");

        int n=inp.nextInt();

        for(int  i=0;i<n;++i)

        {

            System.out.print("Enter the element - ");

            arr.add(inp.nextInt());

        }

        System.out.println(Arrays.asList(arr));

        int sum=0;

        for(int i=0;i<n;++i)

        {

            sum+=arr.get(i);

        }

        System.out.println("Average of the array = "+sum/arr.size());

        System.out.println("Maximum - "+Collections.max(arr));

        System.out.println("Minimum - "+Collections.min(arr));

        for(int i=0;i<arr.size();++i)

        {

            if(arr.get(i)%2!=0)

            {

                arr.remove(Integer.valueOf(arr.get(i)));

            }

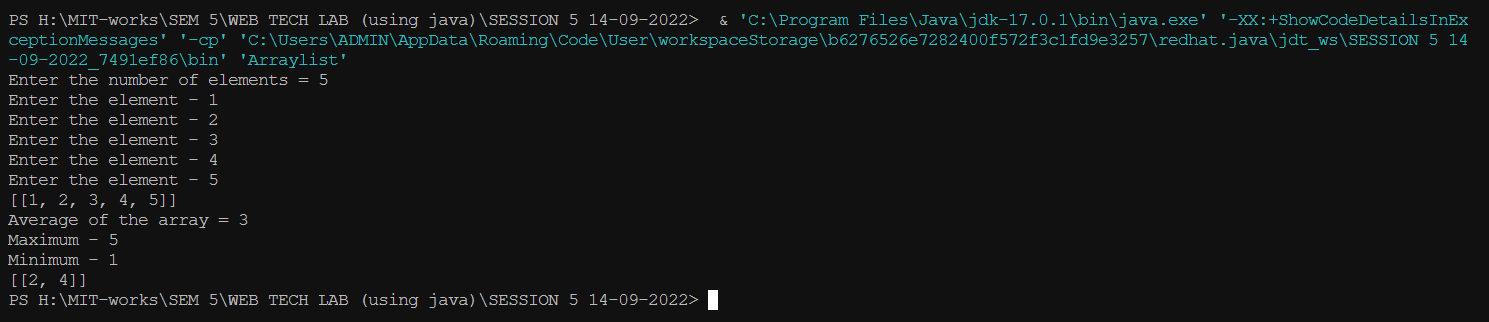
        }

        System.out.println(Arrays.asList(arr));

    }

}

Output:



Result:

Thus, the Java programs for the given problems is completed and the output is displayed and verified.

|  |  |
| --- | --- |
| Exp.No:6 | **Interface, Abstract, Polymorphism, Regular Expression** |
| Date:21/9/2022 |

Aim:

To write Java programs for the given problems using the given concepts.

Programs:

1. Payment

import java.util.\*;

interface Payable

{

    public double getPaymentAmount(double wage,double hours,double weeklysal,double grosssales,double comm);

}

class Employee implements Payable

{

    String f\_name,l\_name,socialSecuritynumber;

    public double getPaymentAmount(double wage,double hours,double weeklysal,double grosssales,double comm)

    {

        double t1=wage\*hours;

        double t2= grosssales\*comm;

        return t1+t2+weeklysal;

    }

    void disp()

    {

        System.out.println("Name = "+f\_name+" "+l\_name);

        System.out.println("Social Security number = "+socialSecuritynumber);

    }

}

class Invoice implements Payable

{

    String PartNo,partDescription;

    int qty;

    double pricePeritem;

    public double getPaymentAmount(double wage,double hours,double weeklysal,double grosssales,double comm)

    {

        double t1=wage\*hours;

        double t2= grosssales\*comm;

        return t1+t2+weeklysal;

    }

    void disp()

    {

        System.out.println("Quantity = "+qty);

        System.out.println("Price per item - "+pricePeritem);

    }

}

class CommisionEmployee extends Employee

{

    double grossSales,commisionRATE;

}

class HourlyEmployee extends Employee

{

    double wage,hours;

}

class BaseplusCommision extends CommisionEmployee

{

    double baseSalary;

}

public class Salary {

    /\*\*

     \* @param args the command line arguments

     \*/

    public static void main(String[] args) {

        double wage, hours,weeklysal,grosssales,comm,basepay;

        Scanner inp=new Scanner(System.in);

        Employee obj1=new Employee();

        Invoice obj2=new Invoice();

        System.out.print("Enter the first name and lastname = ");

        obj1.f\_name=inp.nextLine();

        obj1.l\_name=inp.nextLine();

        System.out.print("Enter the social security number = ");

        obj1.socialSecuritynumber=inp.nextLine();

        System.out.print("Quantity and price per item = ");

        obj2.qty=inp.nextInt();

        obj2.pricePeritem=inp.nextDouble();

        System.out.print("Enter the wage = ");

        wage=inp.nextDouble();

        System.out.print("Enter the hours = ");

        hours=inp.nextDouble();

        System.out.print("Enter the weeklysal = ");

        weeklysal=inp.nextDouble();

        System.out.print("Enter the grosssales = ");

        grosssales=inp.nextDouble();

        System.out.print("Enter the commision rate = ");

        comm=inp.nextDouble();

        System.out.print("Enter the basepay = ");

        basepay=inp.nextDouble();

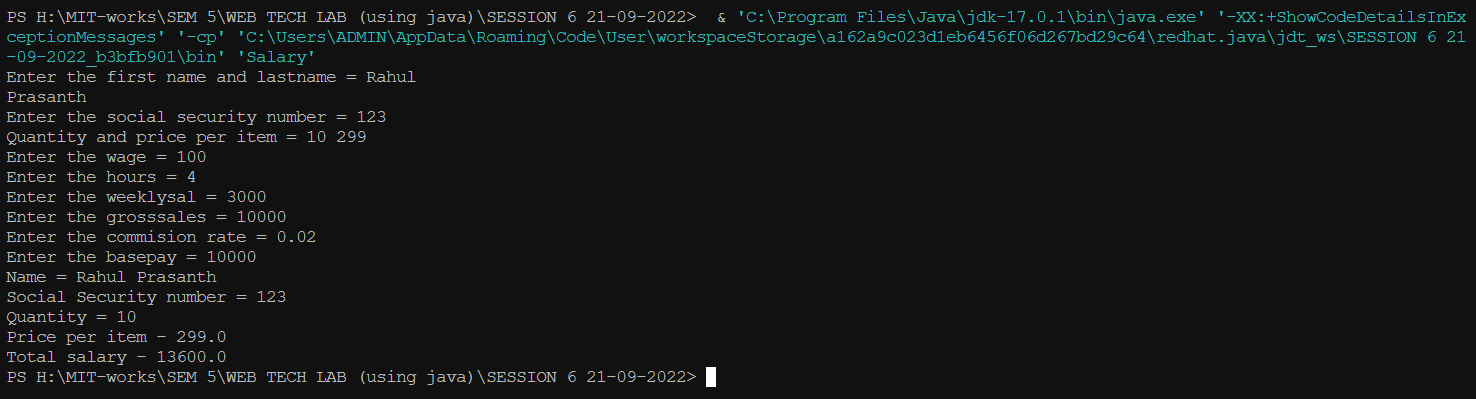
        obj1.disp();obj2.disp();

        System.out.println("Total salary - "+(obj1.getPaymentAmount(wage, hours, weeklysal, grosssales, comm)+basepay));

    }

}

Output:



2.Mark Percentage:

import java.util.\*;

abstract class Mks

{

    abstract void getPercentage();

}

class A extends Mks

{

    double s1,s2,s3;

    A(double a,double b,double c)

    {

        this.s1=a; this.s2=b; this.s3=c;

    }

    void getPercentage()

    {

        System.out.println("Total marks of class A = "+((s1+s2+s3)/300)\*100);

    }

}

class B extends Mks

{

    double s1,s2,s3,s4;

    B(double a,double b,double c,double d)

    {

        this.s1=a; this.s2=b; this.s3=c; this.s4=d;

    }

    void getPercentage()

    {

        System.out.println("Total marks of class B = "+((s1+s2+s3+s4)/400)\*100);

    }

}

public class Marks {

    public static void main(String[] args)

    {

        double s1,s2,s3,s4;

        System.out.println("Enter the 3 subject marks = ");

        Scanner inp=new Scanner(System.in);

        s1=inp.nextDouble();

        s2=inp.nextDouble();

        s3=inp.nextDouble();

        A obj1=new A(s1,s2,s3);

        obj1.getPercentage();

        System.out.println("Enter the 4 subject marks = ");

        s1=inp.nextDouble();

        s2=inp.nextDouble();

        s3=inp.nextDouble();

        s4=inp.nextDouble();

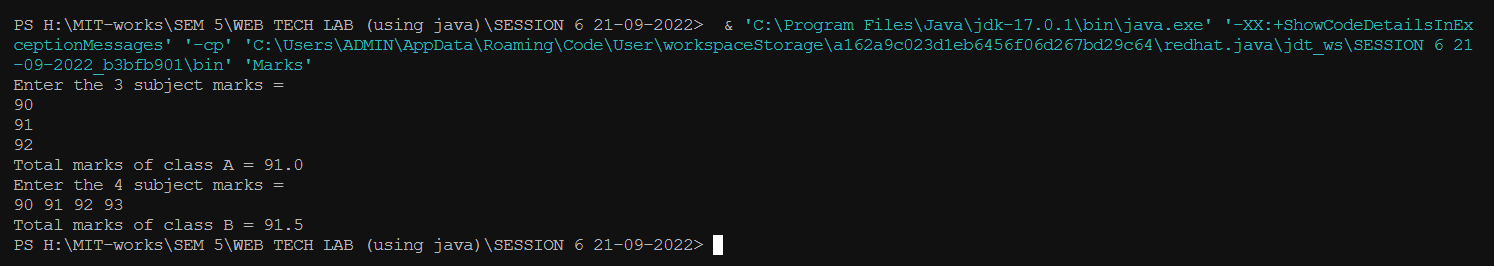
        B obj2=new B(s1,s2,s3,s4);

        obj2.getPercentage();

    }

}

Output:



3.Online Book shop:

import java.util.\*;

class Book

{

    String name,publisher;

    int year,edition;

    double price;

    void SetDetails()

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the book name = ");

        name=inp.next();

        System.out.print("Enter the Publisher name = ");

        publisher=inp.next();

        System.out.print("Enter the published year and edition = ");

        year=inp.nextInt();

        edition=inp.nextInt();

        System.out.print("Price = ");

        price=inp.nextDouble();

    }

    void Display()

    {

        System.out.println("DETAILS ------>");

    }

}

class Tech extends Book

{

        void Display()

        {

            System.out.println("\n\n TECH BOOK : ");

            SetDetails();

            System.out.println("Name = "+name);

            System.out.println("Publisher name = "+publisher);

            System.out.println("Published year = "+year);

            System.out.println("Edition = "+edition);

            System.out.println("Price = "+price);

        }

}

class NonTech extends Book

{

        void Display()

        {

            System.out.println("\n\nNON TECH BOOK : ");

            SetDetails();

            System.out.println("Name = "+name);

            System.out.println("Publisher name = "+publisher);

            System.out.println("Published year = "+year);

            System.out.println("Edition = "+edition);

            System.out.println("Price = "+price);

        }

}

public class Polymorphism {

    public static void main(String[] args)

    {

        //upcasting

        Book obj1=new Tech();

        //downcasting

        Book obj2=new NonTech();

        NonTech dwn=(NonTech)obj2;

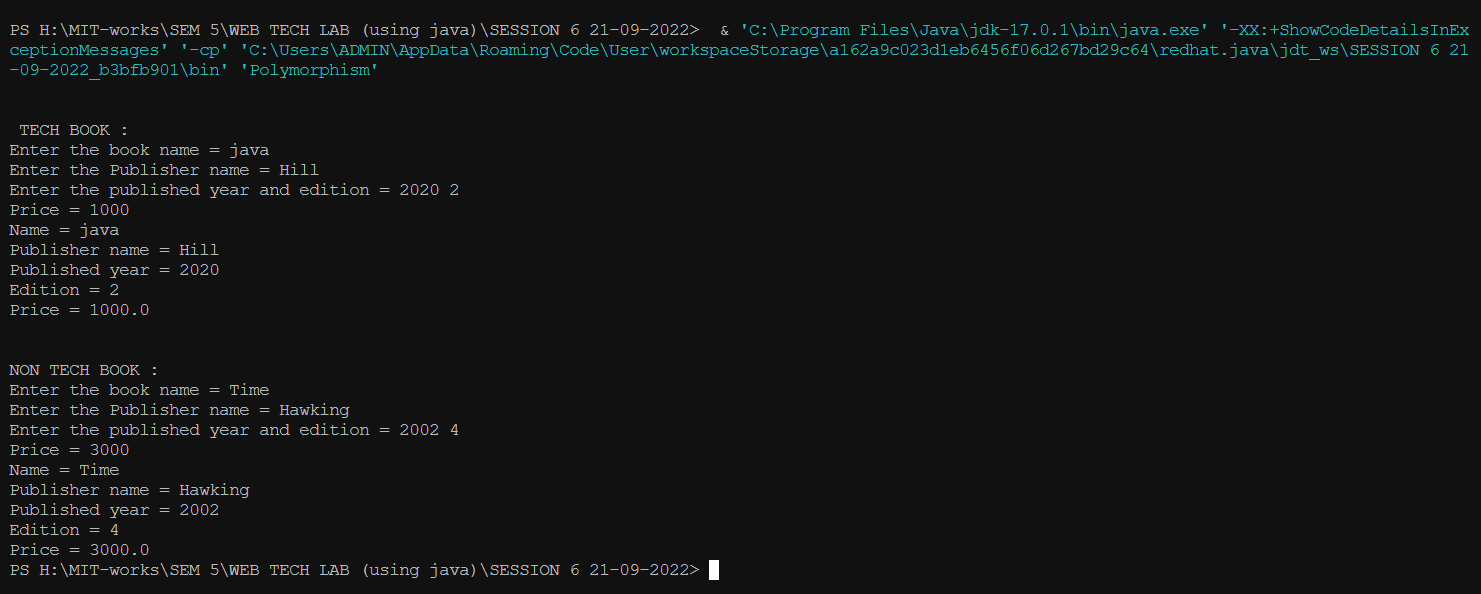
        obj1.Display();

        dwn.Display();

    }

}

Output:



4.Job Seeker:

import java.util.regex.\*;

import java.util.Scanner;

/\*\*

 \*

 \* @author student

 \*/

public class RegEx {

    void Job()

    {

        Scanner inp = new Scanner(System.in);

        System.out.print("Input - ");

        String role=inp.nextLine();

        String reg=".\*job$";

        if(Pattern.matches(reg, role))

        {

            System.out.println("String matches to job");

        }

        else

        {

            System.out.println("String mismatch at end");

        }

    }

    void email()

    {

        Scanner inp = new Scanner(System.in);

        System.out.print("Input - ");

        String role=inp.nextLine();

        String reg="(.\*)@(.\*).com$";

        if(Pattern.matches(reg, role))

        {

            System.out.println("String matches to job");

        }

        else

        {

            System.out.println("String mismatch at end");

        }

    }

    void password()

    {

        Scanner inp = new Scanner(System.in);

        System.out.print("Input - ");

        String role=inp.nextLine();

        String reg="^(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=\\S+$).{8,20}$";

        if(Pattern.matches(reg, role))

        {

            System.out.println("String matches to job");

        }

        else

        {

            System.out.println("String mismatch at end");

        }

    }

    public static void main(String[] args)

    {

        RegEx obj=new RegEx();

        obj.Job();

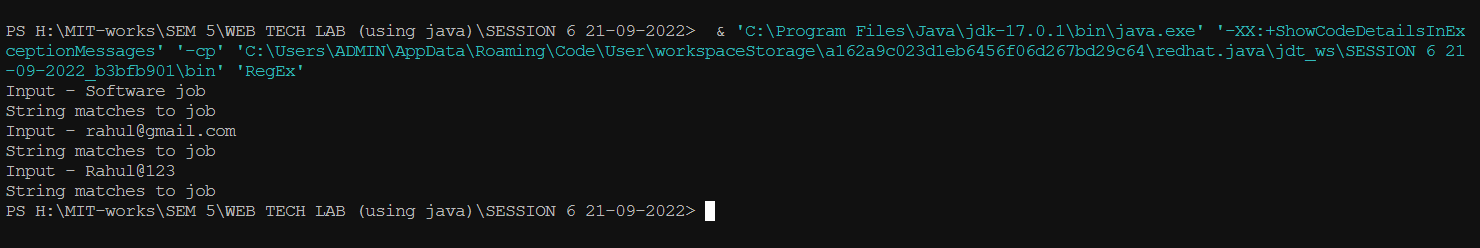
        obj.email();

        obj.password();

    }

}

Output:



5.Secret code:

import java.util.regex.Pattern;

import java.util.regex.Matcher;

import java.util.\*;

public class Validation {

    public static void main(String[] args)

    {

        Scanner inp=new Scanner(System.in);

        System.out.print("Enter the string  = ");

        String str=inp.nextLine();

        String regex="^(?=.\*Cod)(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=\\S+$).{6}$";

        if(!Character.isUpperCase(str.charAt(0))) {System.out.println("Not validated because It is not Capitalized");}

        if(Pattern.matches(regex, str))

        {

            System.out.println("Validated");

        }

        else

        {

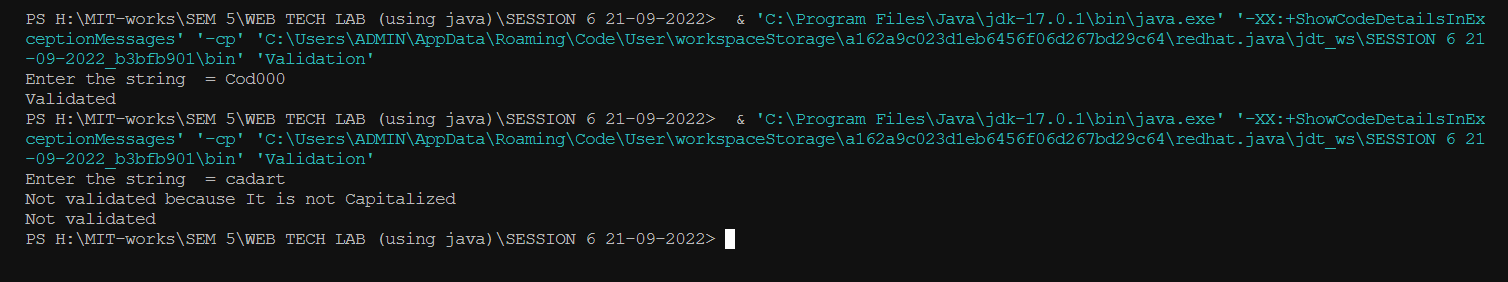
            System.out.println("Not validated");

        }

    }

}

Output:



Result:

Thus, the Java programs for the given problems are completed and the output is displayed and verified.

|  |  |
| --- | --- |
| Exp.No:7 | **Exception Handling** |
| Date:28/9/2022 |

Aim:

To write Java programs for the given problems using the Exception Handling.

Programs:

1. Vending Machine

import java.io.\*;

import java.util.\*;

interface Reseting

{

      public void reset();

}

abstract class Vend implements Reseting

{

      public int choco\_cnt=50,snk\_cnt=10,nut\_cnt=50,juice\_cnt=10;

      public static int tot\_amount=0;

      public int given\_amt=0;

      final int chocolate=10,snack=25,nuts=50,juice=20;

      public void reset()

      {

            this.tot\_amount=0;

            this.given\_amt=0;

            choco\_cnt=50;snk\_cnt=10;nut\_cnt=50;juice\_cnt=10;

            System.out.println("\nResetted Sucessfully\nThank you for visiting!!!");

      }

}

class Access extends Vend{}

class NotPaidFullAmoutException extends Exception

{

      public NotPaidFullAmoutException(String str)

      {

            super(str);

      }

}

class NoSufficientChangeException extends Exception

{

      public NoSufficientChangeException(String str)

      {

            super(str);

      }

}

class SoldOutException extends Exception {

      public SoldOutException(String str) {

            super(str);

      }

}

class VendingMachineJava

{

      void display()

      {

            System.out.println("-----------VENDING MACHINE------------");

            System.out.println("Coins Accestable are 1,5,10,25,50");

            System.out.println("Price list: \n1.Chocolate = (10), 2.Snack = (25), 3.Nuts = (50), 4.Juice = (20) 111.Exit");

      }

      public static void main(String[] args)

      {

            Access obj=new Access();

            Scanner inp=new Scanner(System.in);

            new VendingMachineJava().display();

            while(true)

            {

                  try{

                        int n;

                        System.out.println("Enter the items number = ");

                        n = inp.nextInt();

                        if (n == 111)

                              break;

                        switch (n) {

                              case 1:

                                    obj.choco\_cnt -= 1;

                                    Vend.tot\_amount += obj.chocolate;

                                    break;

                              case 2:

                                    obj.snk\_cnt -= 1;

                                    Vend.tot\_amount += obj.snk\_cnt;

                                    break;

                              case 3:

                                    obj.nut\_cnt -= 1;

                                    Vend.tot\_amount += obj.nuts;

                                    break;

                              case 4:

                                    obj.juice\_cnt -= 1;

                                    Vend.tot\_amount += obj.juice;

                                    break;

                        }

                        if(obj.choco\_cnt<=0||obj.juice\_cnt<=0||obj.nut\_cnt<=0||obj.snk\_cnt<=0)

                        {

                              throw new SoldOutException("Out of STOCK\n");

                        }

                  }

                  catch(SoldOutException ex)

                  {

                        System.out.println("Exception = "+ex.getMessage());

                  }

            }

            System.out.println("\nAmount Need to pay = "+Vend.tot\_amount);

            while (true) {

                  try{

                        System.out.println("Enter the amount = ");

                        int x = inp.nextInt();

                        if (x == 111)

                              break;

                        if(x==1||x==5||x==10||x==25||x==50)

                        {

                              obj.given\_amt += x;

                        }

                        else{

                              throw new NoSufficientChangeException(

                                          "\nPlease enter the correct change!!!\n" + obj.given\_amt);

                        }

                  }

                  catch (NoSufficientChangeException ex) {

                        System.out.println("Exception = " + ex.getMessage());

                  }

            }

            try

            {

                  if(Vend.tot\_amount-obj.given\_amt>0)

                  {

                        throw new NotPaidFullAmoutException("You have not paid the full amount");

                  }

                  else if(Vend.tot\_amount - obj.given\_amt ==0)

                  {

                        System.out.println("Thank you|||");

                        return;

                  }

                  else

                  {

                        System.out.println("Remaining amount to collect = "+Math.abs( Vend.tot\_amount - obj.given\_amt));

                  }

            }

            catch(NotPaidFullAmoutException ex)

            {

                  System.out.println("Exception = " + ex.getMessage());

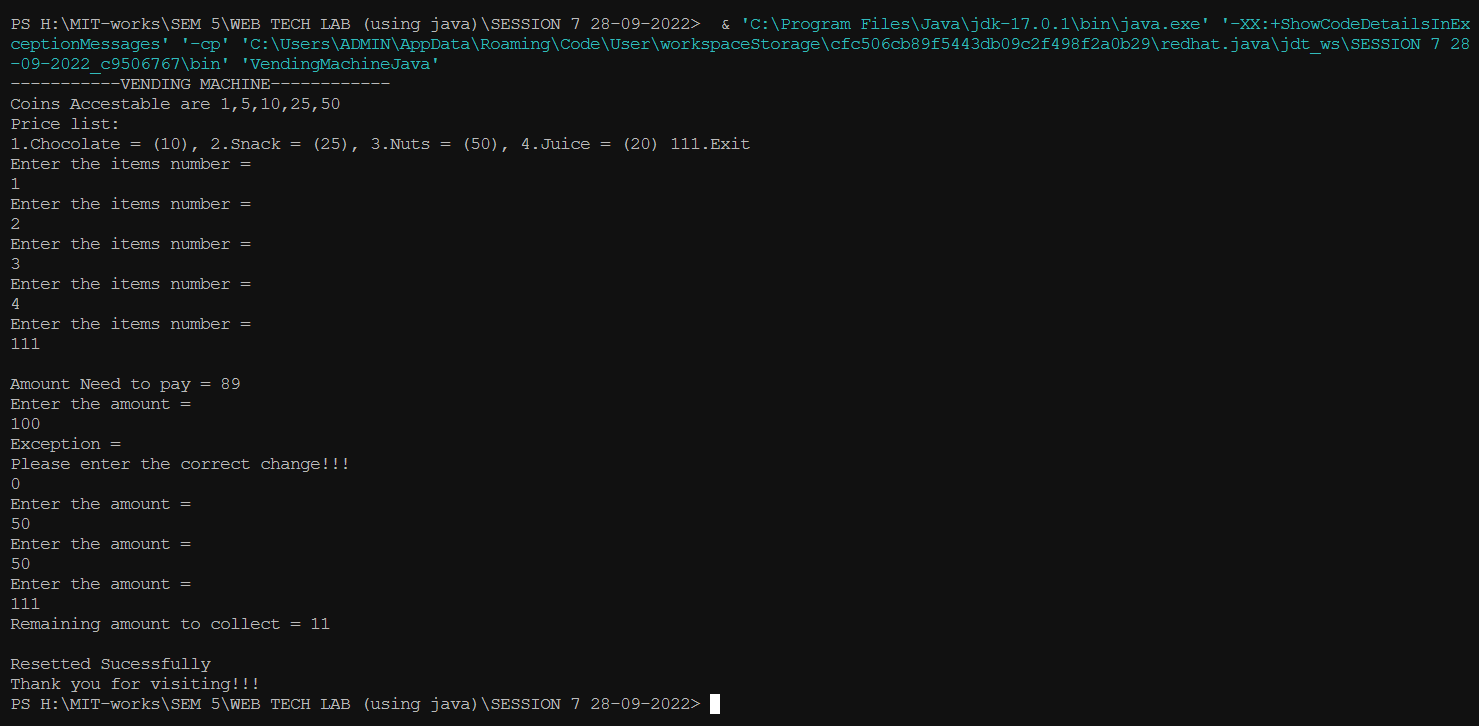
            }

            obj.reset();

      }

}

Output:



2.EMI:

  import java.io.\*;

  import java.util.\*;

import javax.naming.InitialContext;

  class InsufficientBalanceException extends Exception

  {

      public InsufficientBalanceException(String str)

      {

            super(str);

      }

  }

  class Loan

  {

      double EMI(int p,double r,int n)

      {

            return p\*r\*((Math.pow(1+r, n))/((Math.pow(1+r,n))-1));

      }

      public static void main(String[] args)

      {

            double balance=50000;

            int p=100000;

            double r=0.13;

            int n=3;

            try{

                  double ans=new Loan().EMI(p,r,n);

                  if(ans>balance) throw new InsufficientBalanceException("Insufficient Balance");

                  else{

                        balance-=ans;

                  }

                  System.out.printf("Current Balance = %.2f" ,balance);

            }

            catch(InsufficientBalanceException ae)

            {

                  System.out.println("Exception : "+ae.getMessage());

            }

            finally

            {

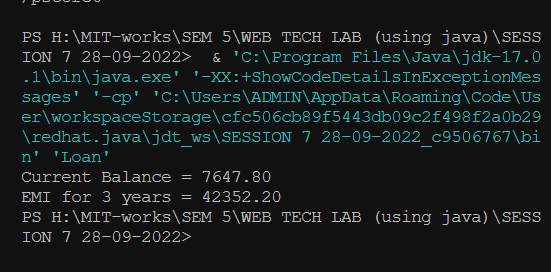
                  System.out.printf("EMI for 3 years = %.2f",new Loan().EMI(p,r,n));

            }

      }

  }

Output:



Result:

Thus, the Java Programs for the given problems are completed and the output is displayed and verified.

|  |  |
| --- | --- |
| Exp.No:8 | **Input and Output Manipulation on files, Serialization** |
| Date:12/10/2022 |

Aim:

To write Java programs for the given problems using the given concepts.

Programs:

1. Word List

import java.util.\*;

import java.io.\*;

import java.util.Map.Entry;

class Alphabetical\_list

{

      void createfile()

      {

            try {

                  File fl = new File("text1.txt");

                  if (fl.createNewFile()) {

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

                  } else {

                        System.out.println("Already exists\n");

                  }

            } catch (IOException e) {

                  System.out.println(e);

                  e.printStackTrace();

            }

      }

      ArrayList<String> readfile()

      {

            ArrayList<String> arr=new ArrayList<>();

            try{

                  File f1=new File("text1.txt");

                  Scanner rdr=new Scanner(f1);

                  while(rdr.hasNext())

                  {

                        String data=rdr.next();

                        arr.add(data);

                        writefile(data);

                        System.out.println(data);

                  }

                  rdr.close();

            }

            catch(FileNotFoundException e)

            {

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

            }

            return arr;

      }

      void writefile(String s)

      {

            try{

                  File fl=new File("newFile1.txt");

                  fl.createNewFile();

                  FileWriter Writer=new FileWriter("newFile1.txt",true);

                  Writer.write(s);

                  Writer.write("\n");

                  Writer.close();

            }

            catch(IOException e)

            {

                  System.out.println(e);

            }

      }

      public static void main(String[] args)

      {

            LinkedHashMap<String, Integer> sortedMap = new LinkedHashMap<>();

            Alphabetical\_list obj=new Alphabetical\_list();

            obj.createfile();

            ArrayList<String> arr=obj.readfile();

            Map<String,Integer> mp=new HashMap<>();

            for(String i:arr)

            {

                  if(mp.containsKey(i))

                  {

                        mp.put(i,mp.get(i)+1);

                  }

                  else{

                        mp.put(i,1);

                  }

            }

            ArrayList<String> kys = new ArrayList<>();

            ArrayList<Integer> values = new ArrayList<>();

            for (Map.Entry entry : mp.entrySet()) {

                  kys.add((String)entry.getKey());

                  values.add((Integer)entry.getValue());

            }

            Collections.sort(kys);

            Collections.sort(values,Collections.reverseOrder());

            try{

                  FileWriter writer=new FileWriter("newFile1.txt",true);

                  writer.write("\n\n");

                  for(String i:kys)

                  {     System.out.println(i + " " + mp.get(i));

                        writer.write(i+" "+mp.get(i)+"\n");

                  }

                  writer.write("\n\n");

                  writer.close();

                  //list 2

                  FileWriter writer2 = new FileWriter("newFile1.txt", true);

                  writer2.write("\n\n");

                  for (Integer i : values) {

                        for (Entry<String, Integer> entry : mp.entrySet()) {

                              if (entry.getValue().equals(i)) {

                                    sortedMap.put(entry.getKey(), i);

                              }

                        }

                  }

                  writer2.close();

                  System.out.println(sortedMap);

                  FileWriter wr=new FileWriter("newFile1.txt",true);

                  wr.write("\n\n");

                  for(Entry<String,Integer> entry : sortedMap.entrySet())

                  {

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

                        wr.write(entry.getKey()+" "+entry.getValue()+"\n");

                  }

                  wr.close();

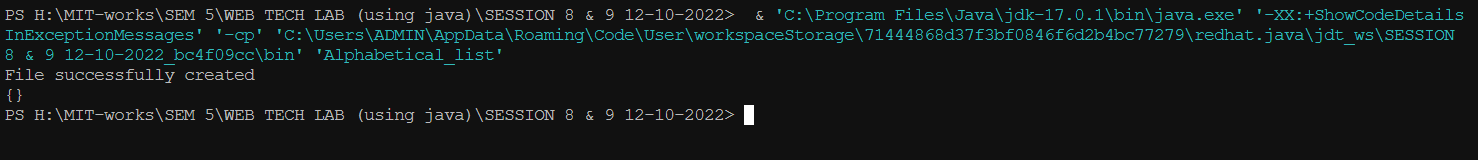
            }

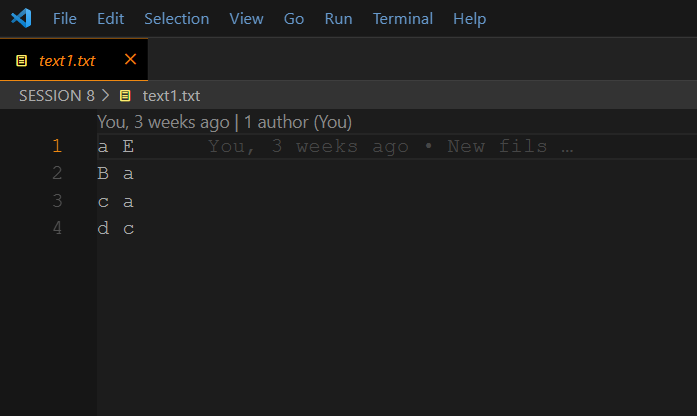
            catch(IOException a){System.out.println(a.getMessage());}

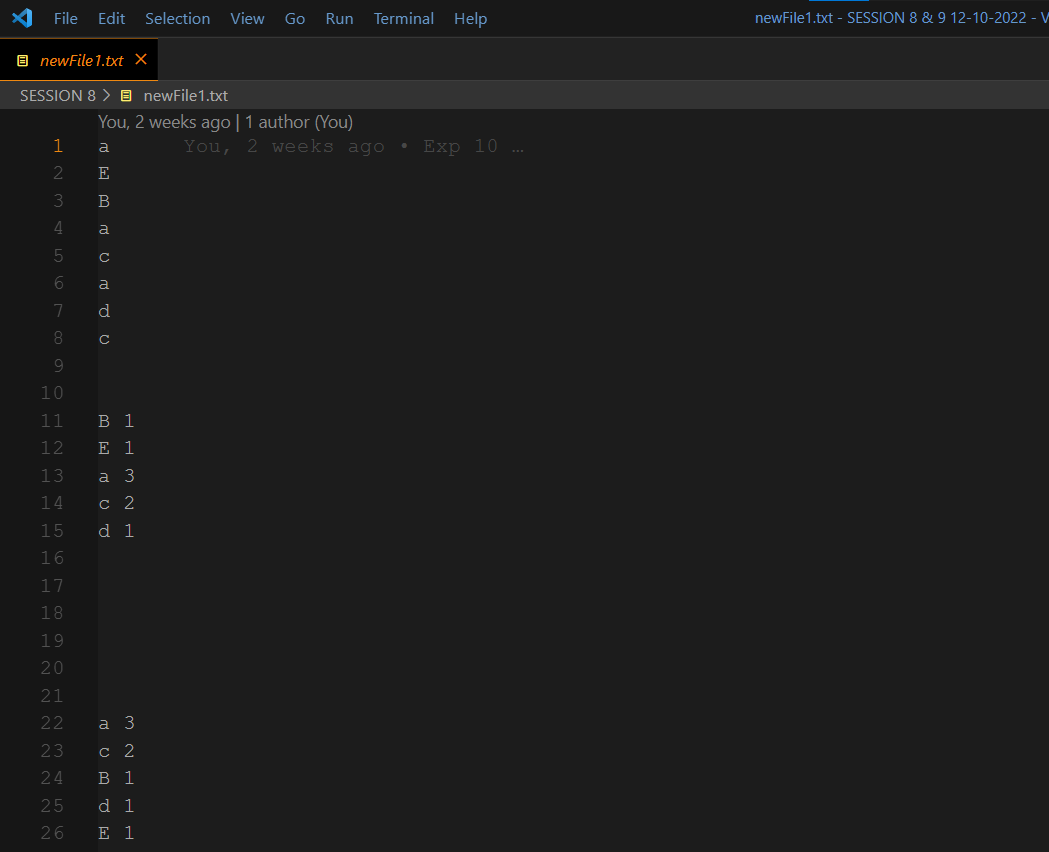
      }

}

Output:







2.Serialization and Deserialization:

import java.io.\*;

import java.util.\*;

class Content implements Serializable

{

      public String Country\_name, continent;

      public transient long population;

      Content()

      {

            System.out.println("Enter Country name, continent, population = ");

            Scanner obj=new Scanner(System.in);

            this.Country\_name=obj.next();

            this.continent = obj.next();

            this.population = obj.nextInt();

      }

}

class Serialize\_Deserailize

{

      void display(Content obj)

      {

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

            System.out.println("Country = "+obj.Country\_name);

            System.out.println("Continent = "+obj.continent);

            System.out.println("Population = "+obj.population);

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

      }

      public static void main(String[] args)

      {

            Content obj=new Content();

            //SERIALIZABLE

            //file creation

            try{

                  File fl=new File("Textfile2.txt");

                  fl.createNewFile();

            }catch(IOException e)

            {

                  System.out.println(e);

            }

            try{

                  FileOutputStream out = new FileOutputStream("Textfile2.txt");

                  ObjectOutputStream objout=new ObjectOutputStream(out);

                  objout.writeObject(obj);

                  out.close();

                  objout.close();

                  System.out.println("Successfully Serialized");

            }

            catch(IOException e)

            {

                  System.out.println(e);

            }

            //DESRIALIZE

            try{

                  Content oriobj=null;

                  FileInputStream fin=new FileInputStream("Textfile2.txt");

                  ObjectInputStream objinp=new ObjectInputStream(fin);

                  oriobj=(Content)objinp.readObject();

                  new Serialize\_Deserailize().display(oriobj);

                  fin.close();

                  objinp.close();

                  System.out.println("Successfully deserialized");

            }

            catch(IOException a)

            {

                  System.out.println(a);

            }

            catch (ClassNotFoundException e) {

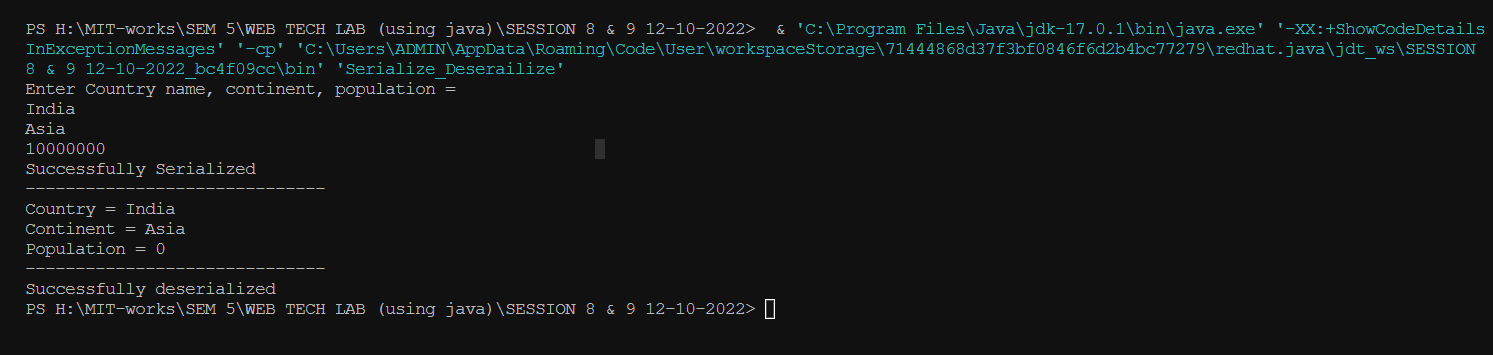
                  System.out.println(e);

            }

      }

}

Output:



3.Copying file contents:

import java.io.\*;

import java.util.\*;

public class Filecopy {

      public static void main(String[] args)

      {

             try{

                  File fl=new File("File3.txt");

                  fl.createNewFile();

             }

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

             try{

                  FileWriter writer=new FileWriter("File3.txt");

                  File fle=new File("text1.txt");

                  Scanner scn=new Scanner(fle);

                  while(scn.hasNext())

                  {

                        String x=(scn.next());

                        writer.write(x+"--->"+x.toUpperCase()+"\n");

                  }

                  writer.close();

             }

             catch(IOException e)

             {

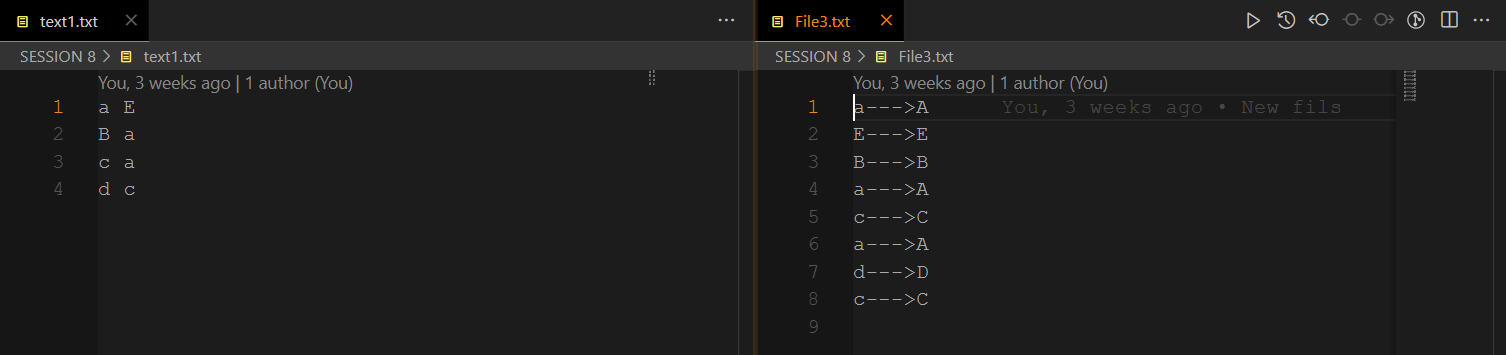
                  System.out.println(e);

             }

      }

}

Output:



Result:

Thus, the Java programs for the given problems is completed and the output is displayed and verified.

|  |  |
| --- | --- |
| Exp.No:9 | **Client-Server network application using Java Sockets** |
| Date:12/10/2022 |

Aim:

To write java programs for the given problems using the Java Sockets.

Programs:

1. TCP chat application

Server:

import java.io.\*;

import java.net.\*;

public class Chat\_Server {

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

        ServerSocket ss=new ServerSocket(1010);

        Socket s=ss.accept();

        System.out.println("Client Cnnected!!");

        InputStreamReader isr=new InputStreamReader(s.getInputStream());

        BufferedReader br=new BufferedReader(isr);

        InputStreamReader is=new InputStreamReader(System.in);

        BufferedReader br2=new BufferedReader(is);

        PrintWriter pr=new PrintWriter(s.getOutputStream(),true);

        String x,y;

        while(true)

        {

            x=br.readLine();

            System.out.println("Client = "+x);

            if(x.equals("Client down")) break;

            y=br2.readLine();

            if(y.equals("q")){pr.println("Server down");break;}

            else pr.println(y);

        }

        ss.close();

    }

}

Client:

import java.io.\*;

import java.net.\*;

public class Chat\_client {

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

        Socket s=new Socket("localhost",1010);

        System.out.println("Server Cnnected!!");

        PrintWriter pr=new PrintWriter(s.getOutputStream(),true);

        InputStreamReader isr=new InputStreamReader(System.in);

        BufferedReader br=new BufferedReader(isr);

        InputStreamReader is=new InputStreamReader(s.getInputStream());

        BufferedReader br2=new BufferedReader(is);

        String x,y;

        while(true)

        {

            x=br.readLine();

            if(x.equals("q")){ pr.println("Client down");break;}

            else pr.println(x);

            y=br2.readLine();

            System.out.println("Server = "+y);

            if(y.equals("Server down")) break;

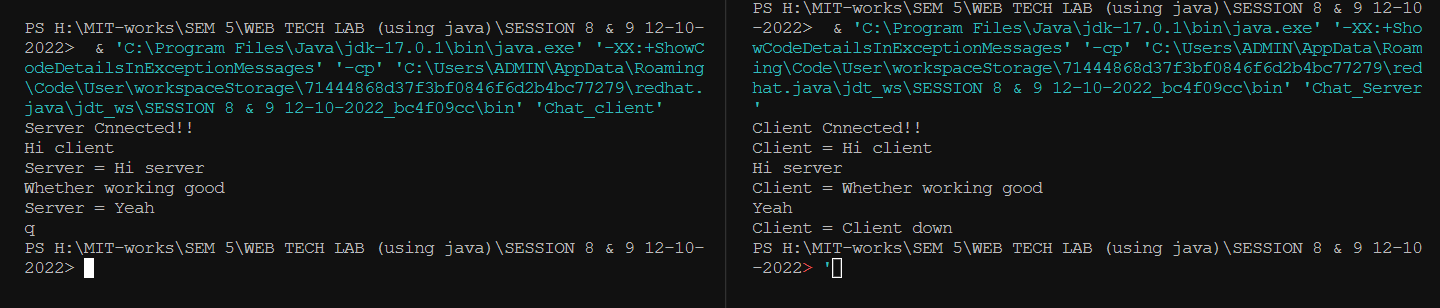
        }

        s.close();

    }

}

Output:



1. UPD chat application

Server:

import java.io.\*;

import java.net.\*;

public class UDPchatserver {

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

        int c\_port = 4150, s\_port = 4160;

        DatagramSocket server = new DatagramSocket(s\_port);

        byte[] buf = new byte[1024];

        DatagramPacket packet = new DatagramPacket(buf, buf.length);

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        InetAddress ia = InetAddress.getLocalHost();

        System.out.println("Server is Running...");

        while (true) {

            server.receive(packet);

            String str = new String(packet.getData(), 0, packet.getLength());

            if (str.equals("STOP")) {

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

                break;

            }

            System.out.println("Client: " + str.toUpperCase());

            String s = new String(br.readLine());

            buf = s.getBytes();

            server.send(new DatagramPacket(buf, s.length(), ia, c\_port));

        }

        server.close();

    }

}

Client:

import java.io.\*;

import java.net.\*;

public class UDPchatclient {

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

        int c\_port = 4150, s\_port = 4160;

        DatagramSocket client = new DatagramSocket(c\_port);

        InetAddress add = InetAddress.getByName("localhost");

        byte buf[] = new byte[1024];

        DatagramPacket dp = new DatagramPacket(buf, buf.length);

        BufferedReader dis = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Client is Running... Type 'STOP' to Quit");

        while (true) {

            String str = new String(dis.readLine());

            buf = str.getBytes();

            if (str.equals("STOP")) {

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

                client.send(new DatagramPacket(buf, str.length(), add, s\_port));

                break;

            }

            client.send(new DatagramPacket(buf, str.length(), add, s\_port));

            client.receive(dp);

            String str2 = new String(dp.getData(), 0, dp.getLength());

            System.out.println("Server: " + str2.toLowerCase());

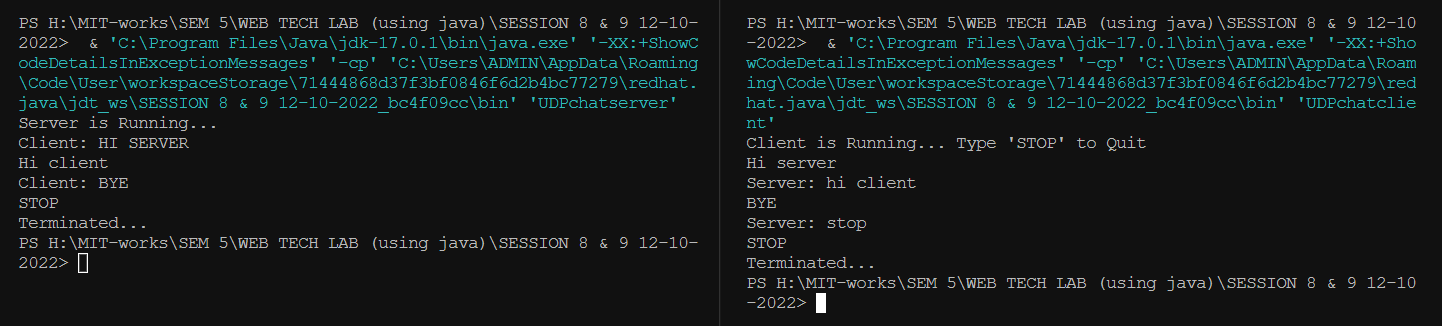
        }

        client.close();

    }

}

Output:



Result:

Thus, the Java programs for the given problems is completed and the output is displayed and verified

|  |  |
| --- | --- |
| Exp.No:10 | **Multithreading** |
| Date:26/10/2022 |

Aim:

To write java program for the given problems using multithreading concept.

Programs:

1. Timed Messages

import java.util.\*;

import java.lang.\*;

class Th1 extends Thread

{

      public void run()

      {

            for (int i = 0; i < 5; ++i) {

            try{

                  Thread.sleep(1000);

            }

            catch(InterruptedException e)

            {

                  System.out.println("Interrupt occurs in thread 1");

            }

            System.out.println("HELLO");}

      }

}

class Th2 extends Thread {

      public void run() {

            for (int i = 0; i < 5; ++i) {

            try {

                  Thread.sleep(2000);

            } catch (InterruptedException e) {

                  System.out.println("Interrupt occurs in thread 1");

            }

            System.out.println("WEAR MASK ! ");}

      }

}

class Th3 extends Thread {

      public void run() {

            for (int i = 0; i < 5; ++i) {

            try {

                  Thread.sleep(5000);

            } catch (InterruptedException e) {

                  System.out.println("Interrupt occurs in thread 1");

            }

            System.out.println("USE SANITIZER !");}

      }

}

public class Q1

{

      public static void main(String[] args)

      {

                  Th1 thread1 = new Th1();

                  Th2 thread2 = new Th2();

                  Th3 thread3 = new Th3();

                  thread1.start();

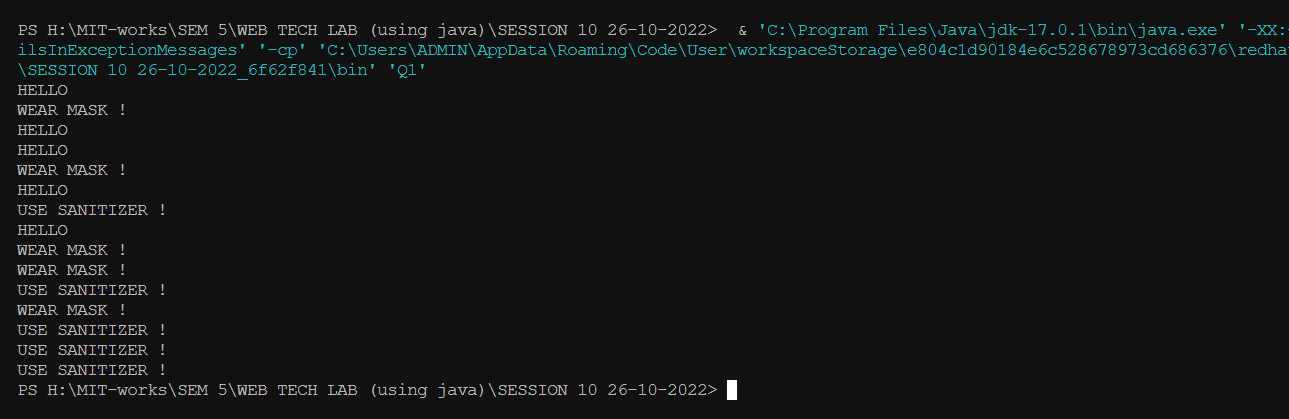
                  thread2.start();

                  thread3.start();

      }

}

Output:



2. Threads with priorities

import java.util.\*;

import java.lang.\*;

class Thr extends Thread{

      public void run(){

            try{

            Thread.sleep(10000);

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

            System.out.println("Content 1");

      }

}

class Thr2 extends Thread {

      public void run() {

            System.out.println("Content 2");

      }

}

public class Q2 {

      public static void main(String[] args)

      {

            Thr2 t1 = new Thr2();

            Thr2 t2 = new Thr2();

            Thr t3 = new Thr();

            Thr t4 = new Thr();

            Thr t5 = new Thr();

            t1.setPriority(10);

            t2.setPriority(10);

            System.out.println("Priorities:");

            System.out.println("Thread 1 = " + t1.getPriority());

            System.out.println("Thread 2 = " + t2.getPriority());

            System.out.println("Thread 3 = " + t3.getPriority());

            System.out.println("Thread 4 = " + t4.getPriority());

            System.out.println("Thread 5 = " + t5.getPriority());

            t1.start();

            t2.start();

            t3.start();

            t4.start();

            t5.start();

            try{

                  t1.sleep(1000);

                  t2.sleep(1000);

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

            if (t1.isAlive())

                  System.out.println("Thread 1 is alive - " + t1.currentThread().getId());

            if (t2.isAlive())

                  System.out.println("Thread 2 is alive - " + t2.currentThread().getId());

            if (t3.isAlive())

                  System.out.println("Thread 3 is alive - " + t3.currentThread().getId());

            if (t4.isAlive())

                  System.out.println("Thread 4 is alive - " + t4.currentThread().getId());

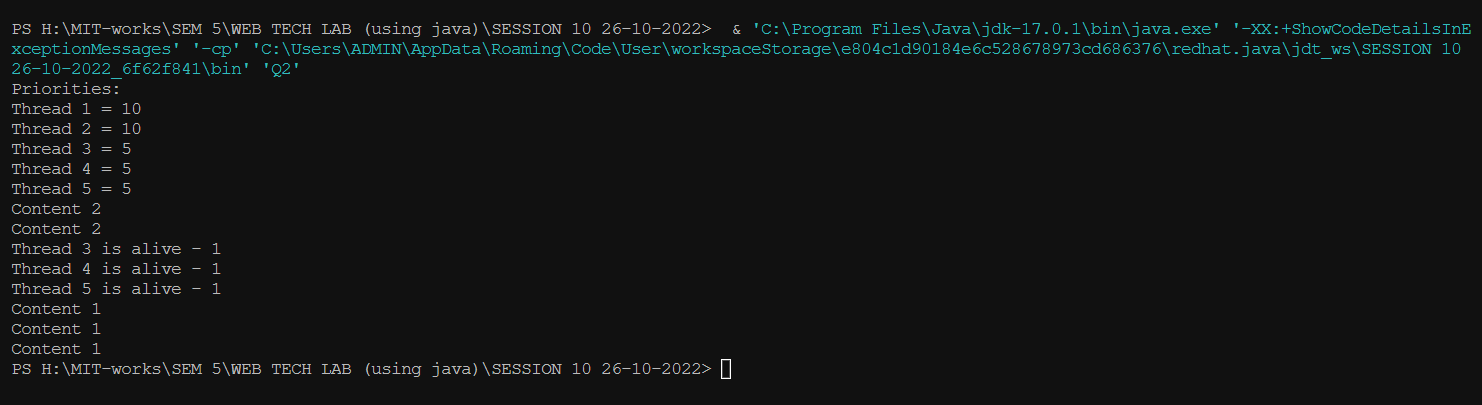
            if (t5.isAlive())

                  System.out.println("Thread 5 is alive - " + t5.currentThread().getId());

      }

}

Output:



3. Multi-thread application

import java.util.\*;

import java.lang.\*;

class thread1 extends Thread

{

      int val;

      public void run()

      {

            Random rand=new Random();

            val=rand.nextInt(100);

      }

      public int get()

      {

            return val;

      }

}

class thread2 extends Thread

{

      int val;

      public void set(int x){val=x;}

      public void run()

      {

      }

      public void compute()

      {

            if(val%2==0)

            {

                  System.out.println(val+" -even- "+Math.pow(val,2));

            }

            else{

                  System.out.println(val + " -odd- " + Math.pow(val, 1));

            }

      }

}

public class Q3

{

      public static void main(String[] args)

      {

            for(int i=0;i<10;++i){

                  thread1 t1=new thread1();

                  Thread thr1 = new Thread(t1);

                  thr1.start();

                  try{thr1.join();}

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

                  int z=t1.get();

                  thread2 t2 = new thread2();

                  Thread thr2 = new Thread(t2);

                  thr2.start();

                  try {

                        thr2.join();

                  } catch (Exception e) {

                        System.out.println("Exception occured");

                  }

                  t2.set(z);

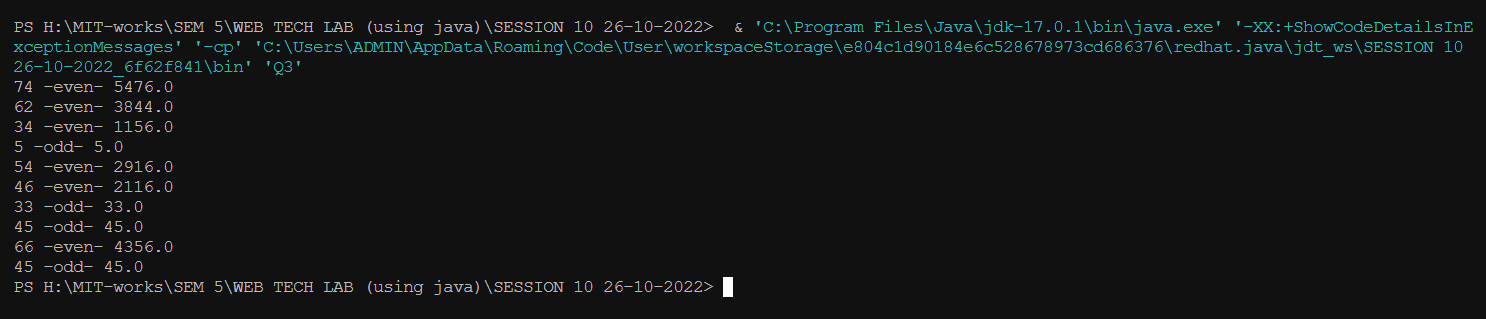
                  t2.compute();

            }

      }

}

Output:



4. Threads with synchronization:

import java.util.\*;

import java.lang.\*;

class Print extends Thread {

      void AZprint() {

            try {

                  for (int i = 0; i < 26; i++) {

                        Thread.sleep(1000);

                        System.out.print((char) (65 + i));

                  }

                  System.out.println();

            } catch (Exception e) {

                  System.out.println(e);

            }

      }

      void ZAprint() {

            try {

                  for (int i = 0; i < 26; i++) {

                        Thread.sleep(2000);

                        System.out.print((char) (90 - i));

                  }

                  System.out.println();

            } catch (Exception e) {

                  System.out.println(e);

            }

      }

}

class ThreadDemo extends Thread {

      String name;

      Print p;

      ThreadDemo(String name, Print p) {

            this.name = name;

            this.p = p;

      }

      public void run() {

            try {

                  synchronized (p) {

                        if (name.equals("thread 1"))

                              p.AZprint();

                        else if (name.equals("thread 2"))

                              p.ZAprint();

                  }

            } catch (Exception e) {

                  System.out.println(e);

            }

      }

}

public class synchonizeThreads {

      public static void main(String[] args) {

            Print p = new Print();

            ThreadDemo t1 = new ThreadDemo("thread 1", p);

            ThreadDemo t2 = new ThreadDemo("thread 2", p);

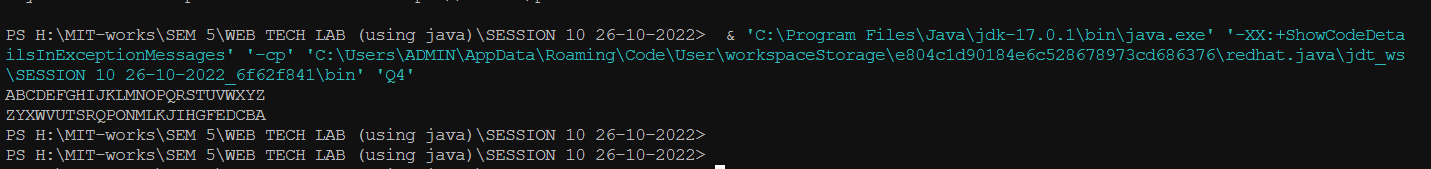
            t1.start();

            t2.start();

      }

}

Output:



Result:

Thus, the Java Programs for the given problems is completed and the output is displayed and verified.

|  |  |
| --- | --- |
| Exp.No:11 | **Interthread communication** |
| Date:29/10/2022 |

Aim:

To write java program for the given problems using interthread communication.

Programs:

1. Producer Consumer Problem

import java.util.\*;

public class WT11 {

    int Stock=5;

    synchronized void consume(int val)

    {

        if(this.Stock<val)

        {

            System.out.println("Stock is insufficient");

            try{wait();this.Stock-=val;}

            catch(Exception e){}

            //now it will call the next thread

        }

        else

        {

            this.Stock-=val;

        }

    }

    synchronized void addStock(int val)

    {

        this.Stock+=val;

        System.out.println("Stock is added");

        notify();

        //now it will notify the previous thread to execute

    }

    public static void main(String[] args) {

        WT11 obj=new WT11();

        Thread thr=new Thread(

                new Runnable()

                {

                    public void run(){obj.consume(94);}

                }

        );

        Thread thr2=new Thread(

                new Runnable()

                {

                    public void run(){obj.addStock(100);}

                }

        );

        System.out.println("Current stock value - "+obj.Stock);

        thr.start();

        thr2.start();

        try{

            thr.join();

            thr2.join();

        }

        catch(Exception e){}

        //if we didn't use join the below statement will execute before the thread execution

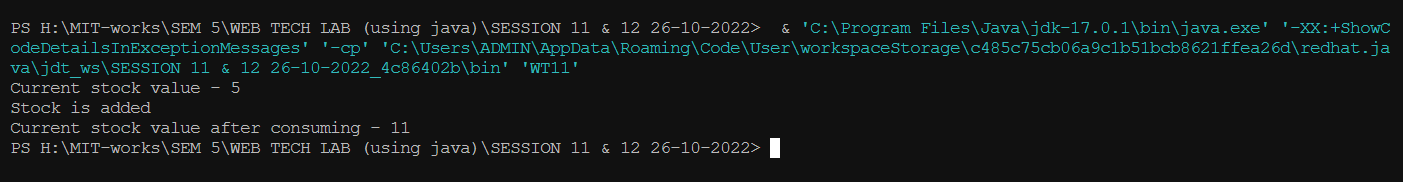
        //beacause of join the statement is printed after the execution of the threads

        System.out.println("Current stock value after consuming - "+obj.Stock);

    }

}

Output:



Result:

Thus, the Java programs for the given problems is completed and the output is displayed and verified.