

**PROGRAM NO : 1**

**AIM:** Define a class 'product' with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

**ALGORITHM:**

Step 1: Start.

Step 2: Define a class having name Product and members as pcode,pname and price.

Step 3: Declare three objects in the class and add the values of each data members into objects.

Step 4: Using if condition check which object has the lowest price and print it.

Step 5: Stop.

**PROGRAM CODE:**

product.java

```
public class product {

    int pcode , price;
    String pname;

    public static void main(String [] args)
    { //objects are created from product class
        product obj1 = new product();
        product obj2 = new product();
        product obj3 = new product();
        //values are given to the objects
        obj1.pcode = 1;
        obj1.pname = "HP";
        obj1.price = 20000;

        obj2.pcode = 2;
        obj2.pname = "Lenovo";
        obj2.price = 10000;

        obj3.pcode = 3;
        obj3.pname = "Asus";
        obj3.price = 50000;

        System.out.println("Details of product which has lowest price
: ");
    }
}
```

```
        if(obj1.price < obj2.price && obj1.price < obj3.price)
        {
            System.out.println("Product Code : "+ obj1.pcode+
"\n" + "Name : "+ obj1.pname+ "\n" + "Price: " + obj1.price );
        }
        else if (obj2.price < obj1.price && obj2.price < obj3.price)
        {
            System.out.println("Product Code "+ obj2.pcode+
"\n" + "Name : "+ obj2.pname+ "\n" + "Price: " + obj2.price );
        }
        else
        {
            System.out.println("Product Code : "+ obj3.pcode+
"\n" + "Name "+ obj3.pname+ "\n" + "Price : " + obj3.price );
        }
    }
}
```

**OUTPUT:**

```
Details of product which has lowest price :
Product Code 2
Name : Lenovo
Price: 10000
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main> □
```

## **PROGRAM NO : 2**

**AIM:** Read 2 matrices from the console and perform matrix addition.

### **ALGORITHM :**

Step 1: Start.

Step 2: Define a class having name AddMatrix.

Step 3: Read row number(m),column number (n) and initialize the double dimensional arrays mat1[],mat2[],res[] with same row number ,column number.

Step 4: Store the first matrix elements into the two-dimensional array matrix mat1[][] using two for loops. i indicates row number, j indicates column index. Similarly second matrix elements in to mat2[][].

Step 5: Add the two matrices using for loop.

for i=0 to i<m

for j=0 to j<n

mat1[i][j] + mat2[i][j] and store it in to the matrix res[i][j] .

Step 6: Print sum of matrices res[i][j].

Stop 7: Stop.

### **PROGRAM CODE:**

matrix.java	<pre> import java.util.Scanner; public class matrix {     public static void main(String args[])     {         int row, col,i,j;         Scanner in = new Scanner(System.in);         System.out.println("Enter the number of rows");         row = in.nextInt();         System.out.println("Enter the number columns");         col = in.nextInt();         int a[][] = new int[row][col];         int b[][] = new int[row][col];         int tot[][] = new int[row][col];         System.out.println("Enter the elements of matrix 1");         for ( i= 0 ; i &lt; row ; i++ )         {             for ( j= 0 ; j &lt; col ;j++ )             a[i][j] = in.nextInt(); </pre>
-------------	--

```

        System.out.println();
    }
    System.out.println("Enter the elements of matrix 2");
    for ( i= 0 ; i < row ; i++ )
    {
        for ( j= 0 ; j < col ;j++ )
        b[i][j] = in.nextInt();
        System.out.println();
    }
    for ( i= 0 ; i < row ; i++ )
    for ( j= 0 ; j < col ;j++ )
    tot[i][j] = a[i][j] + b[i][j] ;
    System.out.println("Sum of matrices:-");
    for ( i= 0 ; i < row ; i++ )
    {
        for ( j= 0 ; j < col ;j++ )
        System.out.print(tot[i][j]+"\\t");
        System.out.println();
    }
}

```

**OUTPUT:**

```

Enter the number of rows
2
Enter the number columns
2
Enter the elements of matrix 1
2
1
3
4

Enter the elements of matrix 2
6
5
4
8

Sum of matrices:-
8      6
7      12
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main>

```

**PROGRAM NO : 3**

**AIM:** Add complex numbers.

**ALGORITHM:**

Step 1: Start.

Step 2: Define a class having name ComplexNumber and data members are real and imaginary number.

Step 3: Define a function ComplexNumber and add values to variables.

Step 4 : Define a function ComplexNumber sum to add complex number using 3<sup>rd</sup> ComplexNumber object and return the value.

Step 5: Print the sum value.

Step 6: Stop.

**PROGRAM CODE:**

complex.java	<pre>import java.util.*;  public class complex {  public static void main(String args[])  {      int c1,i1,c2,i2,c3,i3;      Scanner s=new Scanner(System.in);      System.out.println("Enter the real part of first complex number");      c1=s.nextInt();      System.out.println("Enter the imaginary part of first complex number");      i1=s.nextInt();      System.out.println("Enter the real part of second complex number");      c2=s.nextInt();</pre>
--------------	---

```
System.out.println("Enter the imaginary part of second complex number");

i2=s.nextInt();

System.out.println("The complex numbers are \n");

System.out.println("First complex number =" +c1+" "+"i1+"i");

System.out.println("second complex number =" +c2+" "+"i2+"i");

c3=c1+c2;

i3=i1+i2;

System.out.println("Sum of the complex numbers are =" +c3+" "+"i3+"i");

}

}
```

**OUTPUT:**

```
Enter the real part of first complex number
4
Enter the imaginary part of first complex number
2
Enter the real part of second complex number
6
Enter the imaginary part of second complex number
3
The complex numbers are

First complex number =4+2i
second complex number =6+3i
Sum of the complex numbers are =10+5i
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main> █
```

**PROGRAM NO : 4**

**AIM:** Read a matrix from the console and check whether it is symmetric or not.

**ALGORITHM:**

Step 1: Start.

Step 2 : Read row number,column number and initialize the double dimensional array with same row number ,column number.

Step 3 : Store the first matrix elements into the two-dimensional array matrix using two for loops. i indicates row number, j indicates column index.

Step 4: Check whether the matrix is symmetric or not.

Step 5: Print the symmetric matrix or if not.

Step 6: Stop.

**PROGRAM CODE:**

symmetry.java	<pre> import java.util.*;  public class symmetry {      public static void main(String args[])      {          int row, col,i,j,val,flag = 1;          Scanner s = new Scanner(System.in);          System.out.println("Enter the number of rows");          row = s.nextInt();          System.out.println("Enter the number columns");          col = s.nextInt();          if(row != col)          { </pre>
---------------	--

```
        System.out.println("Order should be same");

    }

    else

    {

        int matrix[][] = new int[row][col];

        int transpos[][] = new int[col] [row];

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

        for ( i= 0 ; i < row ; i++ )

        {

            for ( j= 0 ; j < col ;j++ )

            {

                val =s.nextInt();

                matrix[i][j] = val;

                transpos[i][j]=val;

            }

            System.out.println();

        }

        System.out.println("Matrix is :-");

        for ( i= 0 ; i < row ; i++ )

        {

            for ( j= 0 ; j < col ;j++ )

                System.out.print(matrix[i][j]+"\\t");

            System.out.println();

        }

        System.out.println("Transpose of Given Matrix:-");
```



```
for ( i= 0 ; i < row ; i++ )  
  
    {  
  
        for ( j= 0 ; j < col ;j++ )  
  
            System.out.print(transpos[j][i]+"\\t");  
  
            System.out.println();  
  
        }  
  
        for (i = 0; i < row; i++)  
  
        {  
  
            for (j = 0; j < col; j++)  
  
            {  
  
                if (matrix[i][j] != transpos[j][i])  
  
                {  
  
                    flag = 0;  
  
                    break;  
  
                }  
  
            }  
  
        }  
  
        if( flag == 1)  
  
        {  
  
            System.out.println("\\n\\nThe Given Matrix is Symmetric");  
  
        }  
  
        else  
  
        {  
  
            System.out.println("\\n\\nThe Given Matrix Is Not  
Symmetric");
```

	}
	}
	}
	}

**OUTPUT:**

```
Enter the number of rows
2
Enter the number columns
2
Enter the elements of the matrix
4
2
3
1

Matrix is :-
4      2
3      1
Transpose of Given Matrix:-
4      3
2      1

The Given Matrix Is Not Symmetric
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main> █
```

**PROGRAM NO : 5****AIM:**

Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

**ALGORITHM :**

Step 1: Start.

Step 2: Define a class cpu with data member price and class processor.

Step 3: Class processor contain data members no\_cores,manufacturer and a nested class RAM.

Step 4: class RAM contain memory and manufacturer as data members.

Step 5: Create objects in corresponding classes and display it's details.

Step 6: Stop.

**PROGRAM CODE:**

CPU.java	<pre> import java.util.Scanner; import java.lang.String; public class CPU {     int price;     public class processor{         float no_core;         String manufact;          void proc_info(float n1,String m)         {             no_core=n1;             manufact=m;             System.out.println("PROCESSOR INFORMATION: ");             System.out.println("No. of core: "+no_core);             System.out.println("Manufacturer: "+manufact);         }     } }</pre>
----------	--

```

    }
    static class RAM{
        float mem;
        String manufacturer;

        void mem_info(float n2,String manu)
        {
            mem=n2;
            manufacturer=manu;
            System.out.println("MEMORY
INFORMATION: ");
            System.out.println("Memory size : "+mem+
"gb");
            System.out.println("Manufacturer:
"+manufacturer);
        }
    }

    public static void main(String[] args) {
        CPU obj=new CPU();
        CPU.processor obj1=obj.new processor();
        CPU.RAM obj2 = new CPU.RAM();

        System.out.println("Enter the price of CPU: ");
        Scanner sc = new Scanner(System.in);
        obj.price = sc.nextInt();

        System.out.println("Enter no of core :");
        Scanner sc1 = new Scanner(System.in);
        float n1= sc1.nextFloat();

        System.out.println("Enter the processor manufacturer: ");
        Scanner sc2 = new Scanner(System.in);
        String m=sc2.nextLine();

        System.out.println("Enter memory size in gb :");
        Scanner sc3 = new Scanner(System.in);
        float n2= sc3.nextFloat();

        System.out.println("Enter the memory manufacturer: ");
        Scanner sc4 = new Scanner(System.in);
        String manu=sc4.nextLine();

        sc.close();
        sc1.close();
        sc2.close();
        sc3.close();
        sc4.close();

        System.out.println("CPU Price: "+obj.price);

```

	<pre>obj1.proc_info(n1,m); obj2.mem_info(n2,manu);      }  }</pre>
--	--

**OUTPUT:**

```
Enter the price of CPU:
1500
Enter no of core :
2
Enter the processor manufacturer:
Intel
Enter memory size in gb :
4
Enter the memory manufacturer:
Malosh
CPU Price: 1500
PROCESSOR INFORMATION:
No. of core: 2.0
Manufacturer: Intel
MEMORY INFORMATION:
Memory size : 4.0gb
Manufacturer: Malosh
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main> █
```

**PROGRAM NO : 6****AIM:** Program to Sort strings.**ALGORITHM :**

Step 1: Start

Step 2: Select the first element of the list (i.e., Element at first position in the list).

Step 3: Compare the selected element with all the other elements in the list.

Step 4: In every comparison, if any element is found smaller than the selected element (for Ascending order), then both are swapped.

Step 5: Repeat the same procedure with element in the next position in the list till the entire list is sorted.

Step 6: Stop

**PROGRAM CODE:**

sortstring.java	<pre> import java.util.*; public class sortstring {     public static void main(String[] args) {         int count;         String str;         Scanner sc=new Scanner(System.in);         System.out.println("Enter number of strings: ");         count = sc.nextInt();          String str_arr[]=new String[count];         Scanner sc1=new Scanner(System.in);         System.out.println("Enter the strings: ");         for(int i=0;i&lt;count;i++)         {             str_arr[i]=sc1.nextLine();         }         sc.close();         sc1.close();          for(int i=0;i&lt;count;i++) </pre>
-----------------	---

```
{
    for(int j=i+1;j<count;j++)
    {
        if(str_arr[i].compareTo(str_arr[j])>0)
        {
            str=str_arr[i];
            str_arr[i]=str_arr[j];
            str_arr[j]=str;
        }
    }

    System.out.println("String after sorting: ");
    for(int i=0;i<count;i++)
    {
        System.out.println(str_arr[i]);
    }
}
```

**OUTPUT:**

```
Enter number of strings:
3
Enter the strings:
Ann
Abhishek
Alisha
String after sorting:
Abhishek
Alisha
Ann
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main> █
```

**PROGRAM NO : 7****AIM:**

Search an element in an array.

**ALGORITHM :**

Step 1: Start

Step 2: Check each element in the given stored array with the number provided by the user.

Step 3: If number is found, display the position of the array found, else display number not found.

Step : Stop

**PROGRAM CODE:**

Searching.java	<pre> import java.util.*; public class searching {     public static void main(String[] args) {         int n,key,flag=0;         Scanner s=new Scanner(System.in);         System.out.println("Enter the number of array elements: ");         n=s.nextInt();         int []a=new int[n];         Scanner r=new Scanner(System.in);         System.out.println("Enter the array elemnts: ");         for(int i=0;i&lt;n;i++)         {             a[i]=r.nextInt();         }          System.out.println("Enter the element to be searched: ");         Scanner t =new Scanner(System.in);         key=t.nextInt();          for(int i=0;i&lt;n;i++)         {             if(a[i]==key)             { </pre>
----------------	--



```
        System.out.println("Element "+key+" is  
found at "+ (i+1)+ " position");  
        flag=1;  
        break;  
    }  
}  
if(flag==0)  
{  
    System.out.println("Element"+key+"not found");  
}  
}  
}
```

**OUTPUT:**

```
Enter the number of array elements:  
3  
Enter the array elemnts:  
34  
23  
31  
Enter the element to be searched:  
23  
Element 23 is found at 2 position  
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main> █
```

**PROGRAM NO : 8****AIM:** Perform string manipulations**ALGORITHM :**

Step 1: Start

Step 2: Take the strings provided by the user and concatenate them.

Step 3: Display the combined string with lower case.

Step 3: Display the combined string with upper case.

Step 4: Display the combined string after replacing all the 's' &amp; 'S' characters with '\$' character.

Step 5: Stop

**PROGRAM CODE:**

string\_manipulation.java

```

import java.util.Scanner;
public class string_manipulation {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        String str1,str2;
        System.out.println("Enter string1 : ");
        str1=sc.nextLine();
        System.out.println("Enter string2 : ");
        str2=sc.nextLine();
        System.out.println("Length of string2:
"+str2.length());
        System.out.println("Character at position 3 of
string1 :"+str1.charAt(3));
        System.out.println("Concating string1 and
string2 : "+str1.concat(str2));
        System.out.println("Does string1 contains
'hai': "+str1.contentEquals("hai"));
        System.out.println("Unicode of first character
in string1: "+str1.codePointAt(0));
        System.out.println("Compare string1 and
string2: "+str2.compareTo(str1));
        System.out.println("String1 in lowercase:
"+str1.toLowerCase());
        System.out.println("String2 in upppercase:
"+str2.toUpperCase());

```

	<pre>        System.out.println("String2 after removing space: "+str2.trim());         System.out.println("String1 after replacing : "+str1.replace('i', 'p'));         System.out.println("Is string1 empty?: "+str1.isEmpty());      } }</pre>
--	--

**OUTPUT:**

```
Enter string1 :  
shad  
Enter string2 :  
ibrahim  
Length of string2: 7  
Character at position 3 of string1 :d  
Concating string1 and string2 : shadibrahim  
Does string1 contains 'hai': false  
Unicode of first character in string1: 115  
Compare string1 and string2: -10  
String1 in lowercase: shad  
String2 in uppercase: IBRAHIM  
String2 after removing space: ibrahim  
String1 after replacing : shad  
Is string1 empty?: false  
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main>
```

**PROGRAM NO : 9****AIM:**

Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

**ALGORITHM :**

Step 1: Start

Step 2: Search the 'eNo' attribute of the list of Employee Objects for the 'eNo' provided by the user.

Step 3: If user provided 'eNo' is found inside the Employee object list, display the details of the corresponding employee.

Step 4: Stop

**PROGRAM CODE:**

employeedetails.java	<pre> import java.util.Scanner; public class employeedetails {     int eno;     String ename;     double esalary;      void getinfo()     {         Scanner sc=new Scanner(System.in);         System.out.println("Enter employee number: ");         eno=sc.nextInt();         Scanner sc1=new Scanner(System.in);         System.out.println("Enter employee name: ");         ename=sc1.nextLine();         Scanner sc2=new Scanner(System.in);         System.out.println("Enter employee salary: ");         esalary=sc2.nextDouble();     }     void display()     { </pre>
----------------------	---

	<pre> System.out.println("Employee no: "+eno); System.out.println("Employee name: "+ename); System.out.println("Salary: "+esalary);  } public static void main(String args[]) {     int n;     Scanner sc3=new Scanner(System.in);     System.out.println("Enter the no of employees: ");     n=sc3.nextInt();     employeeedetails e[]=new employeeedetails[n];      for(int i=0;i&lt;n;i++)     {         e[i]=new employeeedetails();         e[i].getinfo();     }     System.out.println("The employee details are:");     for(int i=0;i&lt;n;i++)     {         e[i].display();     }      int no,flag=0;     Scanner sc4=new Scanner(System.in);     System.out.println("Enter employee no to display details: ");     no=sc4.nextInt();      for(int i=0;i&lt;n;i++)     {         if(no==e[i].eno)         {             e[i].display();             flag=1;             break;         }     }     if(flag==0)     {         System.out.println("No such employee");     } } </pre>
--	---

**OUTPUT:**

```
Enter the no of employees:
2
Enter employee number:
102
Enter employee name:
Deepu
Enter employee salary:
2450
Enter employee number:
105
Enter employee name:
Faheem
Enter employee salary:
3400
The employee details are:
Employee no: 102
Employee name: Deepu
Salary: 2450.0
Employee no: 105
Employee name: Faheem
Salary: 3400.0
Enter employee no to display details:
102
Employee no: 102
Employee name: Deepu
Salary: 2450.0
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main> █
```

**PROGRAM NO : 10****AIM:**

Program to find Area of different shapes using overloaded functions

**ALGORITHM :**

Step 1: Start

Step 2: Define the main class

Step 3: Define methods with the same methodname that performs the area operation for each shape

Step 4: Display the areas of each shapes

**PROGRAM CODE:**

areaoverloading.java	<pre> import java.util.*; public class areaoverloading {     double a;     int s;      double area()     {         Scanner sc1=new Scanner(System.in);         System.out.println("Enter value of s: ");         int s=sc1.nextInt();         a=s*s;         System.out.println("Area of square: "+a);         return a;     }     double area(int r)     {         a=3.14*r*r;         System.out.println("Area of circle: "+a);         return a;     }     double area(int b,int h)     {         a=0.5*b*h;         System.out.println("Area of triangle: "+a);         return a;     }     double area(int h1,int b1, int b2) </pre>
----------------------	---

```

        {
            a=h1*(b1+b2)/2;
            System.out.println("Area of trapezoid: "+a);
            return a;
        }
    public static void main(String[] args) {
        areaoverloading obj=new areaoverloading();
        Scanner sc=new Scanner(System.in);

        System.out.println("Enter value of r: ");
        int r=sc.nextInt();
        System.out.println("Enter value of b and h: ");
        int b=sc.nextInt();
        int h=sc.nextInt();
        System.out.println("Enter value of h1,b1 and b2: ");
        int h1=sc.nextInt();
        int b1=sc.nextInt();
        int b2=sc.nextInt();
        obj.area();
        obj.area(r);
        obj.area(b, h);
        obj.area(h1, b1, b2);
    }
}

```

**OUTPUT:**

```

Enter value of r:
2
Enter value of b and h:
4
5
Enter value of h1,b1 and b2:
1
2
3
Enter value of s:
6
Area of square: 36.0
Area of circle: 12.56
Area of triangle: 10.0
Area of trapezoid: 2.0
PS D:\20MCA234-SHADIBRAHIM-OOPS-LAB-main> 

```



**PROGRAM NO : 11****AIM:**

Program to create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class 'Teacher' that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

**ALGORITHM :**

Step 1: Start

Step 2: create class "employee" with the provided data members and define the constructors

Step 3: create another class "Teachers" that performs inheritance of employee class and define constructors for the same

Step 4: create an array of objects in the corresponding class

Step 5: Display the details for the number of teachers provided

**PROGRAM CODE:**

inheritance\_employee.java

```
import java.util.Scanner;
class Employee2
{
    int Empid;
    String Name;
    float Salary;
    String Address;

    Employee2()
    {

    }

    public Employee2(int id, String name, float sal, String
addr)
    {
        Empid = id;
        Name = name;
        Salary = sal;
        Address = addr;
    }
}
```

```

}
class Teacher extends Employee2
{
    String department;
    String Subjects;

    public Teacher(int id, String name, float sal, String addr,
String dept, String sub)
    {
        super(id, name, sal, addr);
        department = dept;
        Subjects = sub;

    }

    Teacher()
    {

    }

    public void display()
    {
        System.out.println("Employee ID - "+Empid);
        System.out.println("Employee Name - "+Name);
        System.out.println("Salary - "+Salary);
        System.out.println("Address - "+Address);
        System.out.println("Department - "+department);
        System.out.println("Subject - "+Subjects);
    }

}

public class inheritance_employee {
    public static void main(String[] args)
    {
        int n;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of Teachers to be
added:");
        n = sc.nextInt();
        Teacher obj[] = new Teacher[n];
        for (int i=0; i<n; i++)
        {
            obj[i] = new Teacher();
        }
        for(int i=0;i<n;i++)
        {
            System.out.println("\t*****");
            System.out.println("Enter Employee ID");
            obj[i].Empid = sc.nextInt();

```

```
        sc.nextLine();
        System.out.println("Enter Employee Name");
        obj[i].Name = sc.nextLine();
        System.out.println("Enter Employee Salary");
        obj[i].Salary = sc.nextFloat();
        sc.nextLine();
        System.out.println("Enter Employee Address");
        obj[i].Address = sc.nextLine();
        System.out.println("Enter Employee Department");
        obj[i].department = sc.nextLine();
        System.out.println("Enter Employee Subject");
        obj[i].Subjects = sc.nextLine();
    }
    System.out.println("\t*****");
    System.out.println("Employee Details:-");
    for(int i=0;i<n;i++)
        obj[i].display();
    }
}
```

**OUTPUT:**

```
Enter the number of Teachers to be added:  
2
```

```
*****
```

```
Enter Employee ID  
101  
Enter Employee Name  
Nithin  
Enter Employee Salary  
2990  
Enter Employee Address  
tvm  
Enter Employee Department  
mca  
Enter Employee Subject  
network
```

```
*****
```

```
Enter Employee ID  
104  
Enter Employee Name  
brahmman  
Enter Employee Salary  
37483  
Enter Employee Address  
kzkd  
Enter Employee Department  
mech  
Enter Employee Subject  
mechanics
```

```
*****
```

```
Employee Details:-  
Employee ID - 101  
Employee Name - Nithin  
Salary - 2990.0  
Address - tvn  
Department - mca  
Subject - network  
Employee ID - 104  
Employee Name - brahmman  
Salary - 37483.0  
Address - kzkd  
Department - mech  
Subject - mechanics  
PS C:\Users\hp> █
```

**PROGRAM NO : 12****AIM:**

Program to create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company\_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

**ALGORITHM :**

Step 1: Start

Step 2: create class "person" with the provided data members and define the constructors

Step 3: create another class "employee" that performs inheritance of person class and another class "teacher" that further inherits the properties of its former class

Step 4: create an array of objects in the corresponding class

Step 5: Display the details of the specific teacher

**PROGRAM CODE:**

inheritance_person.java	<pre>import java.util.Scanner;  class Person {     String Name;     String Gender;     String Address;     int Age;     Person()     {      }     Person(String name, String gender, String addr, int age)     {         Name = name;</pre>
-------------------------	---

```

        Gender = gender;
        Address = addr;
        Age = age;
    }
}

class Employee extends Person
{
    int Empid;
    String Company_name;
    String Qualification;
    float Salary;
    Employee()
    {

    }

    public Employee(String name, String gender, String addr, int
age)
    {
        super(name, gender, addr, age);
    }
    public Employee(int id,String name, String qual, float sal)
    {
        Empid = id;
        Company_name = name;
        Qualification = qual;
        Salary = sal;
    }

}

class Teacher extends Employee
{
    String Subject;
    String Department;
    String Teachersid;
    Teacher()
    {

    }

    Teacher(String sub, String dept, String id)
    {
        Subject = sub;
        Department = dept;
        Teachersid = id;
    }
    public void display()
    {
        System.out.println("Name:" + Name);
        System.out.println("Age:" + Age);
        System.out.println("Gender:" + Gender);
    }
}

```

```

        System.out.println("Address:" + Address);
        System.out.println("Emp id:" + Empid);
        System.out.println("Salary:" + Salary);
        System.out.println("Qualification:" + Qualification);
        System.out.println("Company Name:" +
Company_name);
        System.out.println("Teacher id:" + Teachersid);
        System.out.println("Subject:" + Subject);
        System.out.println("Department:" + Department);
        System.out.println("\n\n");
    }
}
public class inheritance_person {
    public static void main(String[] args)
    {
        int n;
        System.out.println("Enter the no. of Teachers:");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        Teacher obj[] = new Teacher[n];
        for(int i=0;i<n;i++)
            obj[i] = new Teacher();
        sc.nextLine();
        for(int i=0;i<n;i++)
        {
            System.out.println("\t*****");
            System.out.println("Enter the name:");
            obj[i].Name = sc.nextLine();

            System.out.println("Enter the Age:");
            obj[i].Age = sc.nextInt();
            sc.nextLine();
            System.out.println("Enter the Gender:");
            obj[i].Gender = sc.nextLine();
            System.out.println("Enter the Address:");
            obj[i].Address = sc.nextLine();
            System.out.println("Enter the Emp id:");
            obj[i].Empid = sc.nextInt();
            System.out.println("Enter the Salary:");
            obj[i].Salary = sc.nextFloat();
            sc.nextLine();
            System.out.println("Enter the Qualification:");
            obj[i].Qualification = sc.nextLine();
            System.out.println("Enter the Company Name:");
            obj[i].Company_name = sc.nextLine();
            System.out.println("Enter the Teacher id:");
            obj[i].Teachersid = sc.nextLine();
            System.out.println("Enter the Subject:");
            obj[i].Subject = sc.nextLine();
            System.out.println("Enter the Department:");

```

	<pre> obj[i].Department = sc.nextLine(); } System.out.println("\t*****"); System.out.println("Teachers Details:-\n"); for(int i=0;i&lt;n;i++)     obj[i].display(); } } </pre>
--	--

**OUTPUT:**

```

Enter the no. of Teachers:
1
    *****
Enter the name:
reeja
Enter the Age:
23
Enter the Gender:
female
Enter the Address:
tvm
Enter the Emp id:
102
Enter the Salary:
3400
Enter the Qualification:
mca
Enter the Company Name:
infosys
Enter the Teacher id:
10741
Enter the Subject:
oops
Enter the Department:
it
    *****
Teachers Details:-

Name:reeja
Age:23
Gender:female
Address:tvm
Emp id:102
Salary:3400.0
Qualification:mca
Company Name:infosys
Teacher id:10741
Subject:oops
Department:it

```



**PROGRAM NO : 13****AIM:**

Write a program that has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

**ALGORITHM :**

Step 1: Start

Step 2: create class “publisher” and initialize its data members

Step 3: create classes book, literature, fiction. Each class inherit from their subsequent previous class and have its own data members

Step 4: create an array of objects in the corresponding class

Step 5: Display the details of the books required

**PROGRAM CODE:**

Books.java	<pre> import java.util.*;  class publisher {     String pname;     Scanner a=new Scanner(System.in);     public publisher()     {          System.out.println("Enter publisher name:");         pname=a.next();     } } class book extends publisher {     String bookname;     String Author;     public book()     {          System.out.println("Enter book name:");         bookname=a.next();     } } </pre>
------------	---

```

        System.out.println("Enter Author name:");
        Author=a.next();

    }
}
class literature extends book
{
    public literature()
    {
        System.out.println("category:literature");

        System.out.println("_____")
    ;
    }
    void display()
    {
        System.out.println("publishername:"+pbname);
        System.out.println("book name:"+bookname);
        System.out.println("Author name:"+Author);
    }
}
class fiction extends book
{
    public fiction()
    {
        System.out.println("category:fiction");

        System.out.println("_____
_____");
        System.out.println("\n");
    }
    void display()
    {
        System.out.println("publishername:"+pbname);
        System.out.println("book name:"+bookname);
        System.out.println("Author name:"+Author);
    }
}
public class Books {
    public static void main(String[] args) {

        int i,nb;

        Scanner b=new Scanner(System.in);
        System.out.println("Enter the no of literature books
you need to store:");
        nb=b.nextInt();
        literature l[]=new literature[nb];
    }
}

```

	<pre> for( i=0;i&lt;nb;i++) {     l[i]=new literature();  } System.out.println("Enter the no of fictional books you need to store:"); int m; m=b.nextInt(); fiction f[]=new fiction[m]; for( i=0;i&lt;m;i++) {     f[i]=new fiction();  }  System.out.println(" Displaying literature books:\n"); for(i=0;i&lt;nb;i++) {     System.out.println("Displaying details of book no" +(i+1));     l[i].display(); } System.out.println(" Displaying fictional books:\n"); for(i=0;i&lt;m;i++) {     System.out.println("Displaying details of book no: \t" +(i+1));     f[i].display(); }  } </pre>
--	--

**OUTPUT:**

```
Enter the no of literature books you need to store:
```

```
2
```

```
Enter publisher name:
```

```
tolins
```

```
Enter book name:
```

```
Scarewitch
```

```
Enter Author name:
```

```
Albert
```

```
category:literature
```

```
-----  
Enter publisher name:
```

```
yenkhas
```

```
Enter book name:
```

```
illusion
```

```
Enter Author name:
```

```
henry
```

```
category:literature
```

```
-----  
Enter the no of fictional books you need to store:
```

```
1
```

```
Enter publisher name:
```

```
redshamps
```

```
Enter book name:
```

```
science of living
```

```
Enter Author name:
```

```
category:fiction
```

```
-----
```

```
Displaying literature books:
```

```
Displaying details of book no1
```

```
publishername:tolins
```

```
book name:Scarewitch
```

```
Author name:Albert
```

```
Displaying details of book no2
```

```
publishername:yenkhas
```

```
book name:illusion
```

**PROGRAM NO : 14****AIM:**

Program to Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

**ALGORITHM :**

Step 1: Start

Step 2: Create an interface results.

Step 3: Create classes Student and Sports that implements the interface results

Step 4: Display the academic and sports score of the student

**PROGRAM CODE:**

inheritance_student.java	<pre> import java.util.Scanner;  interface results {     void getdata();     int display(); }  class Student implements results {     int std_id,std_tmark;     String std_name;     @Override     public void getdata()     {         Scanner sc = new Scanner(System.in);         System.out.println("Enter the name of the student:");         std_name = sc.nextLine();         System.out.println("Enter the student id:");         std_id = sc.nextInt();         System.out.println("Enter total academic mark:");         std_tmark = sc.nextInt();     }     @Override     public int display() </pre>
--------------------------	---

```

    {
        System.out.println("\t-----Student details-----");
        System.out.println("Student name: " +std_name);
        System.out.println("Student id: " +std_id);
        System.out.println("Total mark:" +std_tmark);
        return std_tmark;
    }
}

class Sports implements results
{
    int tmarks;
    @Override
    public void getdata()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the marks obtained in
sports:");
        tmarks = sc.nextInt();
    }
    @Override
    public int display()
    {
        System.out.println("Marks obtained in Sports:"+
tmarks);
        return tmarks;
    }
}

public class inheritance_student {
    public static void main(String[] args)
    {
        int mark;
        Student ob = new Student();
        Sports obj = new Sports();
        ob.getdata();
        obj.getdata();
        mark = ob.display();
        mark = mark + obj.display();
        System.out.println("Marks(Academic+Sports)="+
mark);
    }
}

```

**OUTPUT:**

```
Enter the name of the student:
shad
Enter the student id:
2028
Enter total academic mark:
78
Enter the marks obtained in sports:
45
-----Student details-----
Student name: shad
Student id: 2028
Total mark:78
Marks obtained in Sports:45
Marks(Academic+Sports)=123
PS C:\Users\hp> █
```

**PROGRAM NO : 15****AIM:**

Program to create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

**ALGORITHM :**

Step 1: Start

Step 2: Create an interface Calculation that has the methods to take inputs and compute area and perimeter

Step 3: Create classes Circle and Rectangle that implements calculation

Step 4: Display the area and perimeter of circle or rectangle depending upon the choice the user selects.

**PROGRAM CODE:**

Areaperimeter.java	<pre> import java.util.Scanner; interface calculation{     void input();     void area();     void perimeter(); } class Circle implements calculation {     int r;     double pi=3.14,ar,pr;     @Override     public void input()     {         Scanner sc1=new Scanner(System.in);         System.out.println("Enter radius:");         r=sc1.nextInt();     }      @Override     public void area()     { </pre>
--------------------	---



```

        ar=pi*r*r;
        System.out.println("Area of the circle:"+ar);
    }

    @Override
    public void perimeter()
    {
        pr=2*pi*r;
        System.out.println("Perimeter of the circle:"+pr);
    }
}

class Rectangle extends Circle{
    int l,b;
    double ar,pr;
    public void input()
    {
        super.input();
        Scanner sc2=new Scanner(System.in);
        System.out.println("Enter length:");
        l=sc2.nextInt();
        System.out.println("Enter breadth:");
        b=sc2.nextInt();
    }
    public void area()
    {
        super.area();
        ar=l*b;
        System.out.println("Area of rectangle:"+ar);
    }
    public void perimeter()
    {
        super.perimeter();
        pr=(2*l)+(2*b);
        System.out.println("Perimeter of rectangle:"+pr);
    }
}

public class Areaperimeter {
    public static void main(String args[])
    {
        int choice;
        Rectangle obj=new Rectangle();
        while(true)
        {
            Scanner sc3=new Scanner(System.in);
            System.out.println("\n" + "1.Input the values"+"\\n"
+ "2.Find area" + "\\n" + "3.Find perimeter" + "\\n" +
"4.Exit");
            System.out.println("Enter the choice:");
            choice=sc3.nextInt();
            switch(choice)

```

```

        {
            case 1:
                obj.input();
                break;
            case 2:
                obj.area();
                break;
            case 3:
                obj.perimeter();
                break;
            case 4:
                return;
            default:
                System.out.println("Enter correct choice:");
        }
    }
}

```

**OUTPUT:**

```

1.Input the values
2.Find area
3.Find perimeter
4.Exit
Enter the choice:
1
Enter radius:
5
Enter length:
7
Enter breadth:
8

1.Input the values
2.Find area
3.Find perimeter
4.Exit
Enter the choice:
2
Area of the circle:78.5
Area of rectangle:56.0

1.Input the values
2.Find area
3.Find perimeter
4.Exit
Enter the choice:
3
Perimeter of the circle:31.400000000000002
Perimeter of rectangle:30.0

1.Input the values
2.Find area
3.Find perimeter
4.Exit
Enter the choice:
4
PS C:\Users\hp> 

```

**PROGRAM NO : 16****AIM:**

Prepare bill with the given format using calculate method from interface.

Order No.				
Date :				
Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100
Net. Amount				150

**ALGORITHM :**

Step 1: Start

Step 2: Create interface calc that performs the calculation operations.

Step 3: Create class bill that implements the interface calc

Step 4: Display the net amount by acquiring the data for the specific inputs.

**PROGRAM CODE:**

elec_bill.java	<pre>import java.util.Scanner;  interface calc {     void calculate(); }  class bill implements calc {     String date,name,p_id;</pre>
----------------	---

```

int quantity;
double unit_price,total,namount=0;
Scanner sc = new Scanner(System.in);
public void getdata()
{
    System.out.println("\nEnter product id:");
    p_id = sc.nextLine();
    System.out.println("Enter product name:");
    name = sc.nextLine();
    System.out.println("Enter the Quantity:");
    quantity = sc.nextInt();
    System.out.println("Enter the unit price:");
    unit_price = sc.nextDouble();
}

@Override
public void calculate()
{
    total = quantity * unit_price;
}
public void display()
{
    System.out.println(p_id+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit_price+"\t\t"+total);
}
}
public class elec_bill
{
    public static void main(String[] args)
    {
        int n,i;
        double namount=0,t;
        int ran;
        String date;
        t = Math.random() * 1000000;
        ran = (int) t;
        Scanner sc = new Scanner(System.in);
        System.out.println("Order no. #"+ran);
        System.out.println("Enter the date:");
        date = sc.nextLine();
        System.out.println("Enter how many products are there:");
        n = sc.nextInt();
        bill ob[] = new bill[n];
        for(i=0;i<n;i++)
            ob[i] = new bill();
        for(i=0;i<n;i++){
            ob[i].getdata();
            ob[i].calculate();
        }
    }
}

```

```

        System.out.println("Date:"+date);
        System.out.println("Product Id \tName\t Quantity\t unit
price\t Total ");
        System.out.println("-----");
        -----");
        for(i=0;i<n;i++){
            ob[i].display();
            namount += ob[i].total;
        }
        System.out.println("-----");
        -----");
        System.out.println("\t\t\tNet.Amount\t"+ namount);

    }
}

```

**OUTPUT:**

```

Order no. #154861
Enter the date:
23-12-2021
Enter how many products are there:
2

Enter product id:
42362
Enter product name:
led bulb
Enter the Quantity:
45
Enter the unit price:
560

Enter product id:
42363
Enter product name:
cfl tube
Enter the Quantity:
10
Enter the unit price:
78
Date:23-12-2021
Product Id      Name      Quantity      unit price      Total
-----
42362          led bulb          45          560.0      25200.0
42363          cfl tube          10          78.0       780.0
-----
                        Net.Amount      25980.0

PS C:\Users\hp>

```

**PROGRAM NO : 17****AIM:**

Program to Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

**ALGORITHM :**

Step 1:Start

Step 2: To create a package named graphics, create a folder of the same name in the directory.

Step 3: Inside the graphics folder, create modules for finding the areas of rectangle, circle, triangle and square.

Step 4: Outside the graphics folder, write a program to access the modules mention above and print the output

Step 5: Stop

**PROGRAM CODE:**

Shapes.java	<pre> package Graphics; import java.util.*; interface calculation {     void carea();     void rarea();     void tarea();     void sarea(); }  public class Shapes implements calculation {     Scanner s=new Scanner(System.in);     public void carea()     {         System.out.println("Enter the radius");     } } </pre>
-------------	--

	<pre> int r=s.nextInt(); System.out.println("Area of Circle =" +3.14*r*r); } public void rarea() {     System.out.println("Enter length:");     int l=s.nextInt();     System.out.println("Enter breadth:");     int b=s.nextInt();     System.out.println("Area of Rectangle:"+(l*b)); } public void tarea() {     System.out.println("Enter the base:");     int b = s.nextInt();      System.out.println("Enter the height:");     int h = s.nextInt();      System.out.println("Area of the triangle     =" +(0.5*b*h)); } public void sarea() {     System.out.println("Enter the side:");     int side = s.nextInt();      System.out.println("Area of the square     =" +(side*side)); } } </pre>
area.java	<pre> import Graphics.Shapes; public class area {     public static void main(String args[])     {         Shapes obj=new Shapes();         System.out.println("Area of different Shapes"+"\n"+"----         -----");         System.out.println("Circle"+"\n"+"-----");         obj.carea();         System.out.println("Rectangle"+"\n"+"-----");         obj.rarea();         System.out.println("Triangle"+"\n"+"-----");         obj.tarea();         System.out.println("Square"+"\n"+"-----");         obj.sarea();     } } </pre>

## OUTPUT:

```
PROBLEMS 45 OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\hp\Desktop\JAVA> & 'c:\Users\hp\.vscode\extensions\vscjava.vscode-java-debug-0.34.0\
' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-Dfile.encoding=UTF-8' '-cp' 'C:\
4c0d21fc24caefe8\redhat.java\jdt_ws\JAVA_64e4e41d\bin' 'area'
Area of different Shapes
-----
Circle
-----
Enter the radius
4
Area of Circle =50.24
Rectangle
-----
Enter length:
5
Enter breadth:
4
Area of Rectangle:20
Triangle
-----
Enter the base:
2
Enter the height:
3
Area of the triangle =3.0
Square
-----
Enter the side:
4
Area of the square =16
PS C:\Users\hp\Desktop\JAVA> █
```



**PROGRAM NO : 18****AIM:**

Program to create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers

**ALGORITHM :**

Step 1: Start

Step 2: To create a package named arithmetic, create a folder of the same name in the directory. Here inside that we have another module named operation

Step 3: Inside arithmetic package, create modules to perform addition, subtraction, multiplication and division of 2 numbers.

Step 4: Outside the folder, write another program that access the above module and print the output.

Step 5: Stop

**PROGRAM CODE:**

arithmeticoperations.java	<pre> import java.util.Scanner; interface cal {     void add();     void sub();     void mul();     void div(); }  public class arithmeticoperations implements cal {     Scanner sc=new Scanner(System.in);      public void add()     {         System.out.println("Enter first number:");         int a=sc.nextInt();         System.out.println("Enter second number:"); </pre>
---------------------------	---

	<pre> int b=sc.nextInt(); System.out.println("Sum="+(a+b));  } public void sub() {     System.out.println("Enter first number:");     int a=sc.nextInt();     System.out.println("Enter second number:");     int b=sc.nextInt();     System.out.println("Difference="+(a-b));  } public void mul() {     System.out.println("Enter first number:");     int a=sc.nextInt();     System.out.println("Enter second number:");     int b=sc.nextInt();     System.out.println("Product="+(a*b));  } public void div() {     System.out.println("Enter first number:");     int a=sc.nextInt();     System.out.println("Enter second number:");     int b=sc.nextInt();     System.out.println("Quotient="+(a/b));  } } </pre>
Operation.java	<pre> import Arithmetic.arithmeticoperations; public class Operation {     public static void main(String args[])     {         arithmeticoperations obj=new arithmeticoperations();         System.out.println("ARITHMETIC OPERATIONS"+"\\n"+"-----");         System.out.println("\\nAddition"+"\\n"+"-----");         obj.add();         System.out.println("\\nSubtraction"+"\\n"+"----- ");         obj.sub();         System.out.println("\\nMultiplication"+"\\n"+"----- --");         obj.mul();         System.out.println("\\nDivision"+"\\n"+"-----");         obj.div();     } } </pre>

	}
--	---

**OUTPUT:**

```
ARITHMETIC OPERATIONS
```

```
-----
```

```
Addition
```

```
-----
```

```
Enter first number:
```

```
5
```

```
Enter second number:
```

```
4
```

```
Sum=9
```

```
Subtraction
```

```
-----
```

```
Enter first number:
```

```
8
```

```
Enter second number:
```

```
7
```

```
Difference=1
```

```
Multiplication
```

```
-----
```

```
Enter first number:
```

```
4
```

```
Enter second number:
```

```
2
```

```
Product=8
```

```
Division
```

```
-----
```

```
Enter first number:
```

```
6
```

```
Enter second number:
```

```
2
```

```
Quotient=3
```

```
PS C:\Users\hp\Desktop\JAVA> █
```

**PROGRAM NO : 19****AIM:**

Program to create a user defined exception class to authenticate the user name and password.

**ALGORITHM :**

Step 1: Start

Step 2: Create a class named usernameexception that inherits Exception class with a constructor that calls Exception class constructor and pass error message.

Step 3: Create a class named passwordexception that inherits Exception class with a constructor that calls Exception class constructor and pass error message.

Step 4: Inside the main(), Read the username and password.

Step 5: Inside the try block, we throw usernameexception and passwordexception with appropriate message if any of the condition is true:

- If username is empty
- If password is empty
- If password doesn't contain special characters
- If username length is less than 6
- If password is not strong enough

Step 6: Inside the catch block with parameter usernameexception's object, print "USERNAME EXCEPTION OCCURED"

Step 7: Inside the catch block with parameter passwordexception's object, print "PASSWORD EXCEPTION OCCURED"

Step 8: Stop

**PROGRAM CODE:**

userauthentication.java	<pre>import java.util.Scanner; class passwordException extends Exception {      private static final long serialVersionUID = 1L;      passwordException(String s){         super(s);     } }</pre>
-------------------------	--

```

    }

}

class UsernameException extends Exception {

    private static final long serialVersionUID =
1L;

    public UsernameException(String msg) {
        super(msg);
    }

}

public class Userauthentication
{
    public static void main(String[] args) {
        String username, password;

        Scanner sc = new Scanner(System.in);
        System.out.print("Please enter the Username and
Password"+"\\n"+"-----");
        System.out.print("\\nENTER USERNAME:");
        username = sc.nextLine();
        System.out.print("ENTER PASSWORD:");
        password = sc.nextLine();

        sc.close();
        try
        {

            if(username==""){
                throw new UsernameException("Fields
cannot be empty!!!");
            }
            if(password==""){
                throw new passwordException("Fields
cannot be empty!!!");
            }
            else if (username.length()<6) {
                throw new UsernameException("Username
must be atleast 5 characters!");
            }
            else if (password.length()<8) {
                throw new passwordException("Please
enter a strong password");
            }
            if (!(password.contains("@") ||
password.contains("#")
|| password.contains("!") || password.contains("~")

```

```

        || password.contains("$") || password.contains("%")
        || password.contains("^") || password.contains("&")
        || password.contains("*") || password.contains("(")
        || password.contains(")") || password.contains("-")
        || password.contains("+") || password.contains("/")
        || password.contains(":") || password.contains(".")
        || password.contains(",") || password.contains("<")
        || password.contains(">") || password.contains("?")
        || password.contains("|")) {
            throw new passwordException("Password should
            contain at least one special character");
        }

        else {
            System.out.println("Login
            Successfully!!!");
        }
    }
    catch (UsernameException e) {
        System.out.println("Exception Occurred. .
        "+e);
    }
    catch (passwordException e) {
        System.out.println("Exception Occurred. .
        "+e);
    }
}
}

```

## OUTPUT:

```

PROBLEMS 45 OUTPUT TERMINAL DEBUG CONSOLE

Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\hp\Desktop\JAVA> & 'c:\Users\hp\.vscode\
' '--enable-preview' '-XX:+ShowCodeDetailsInException
4c0d21fc24caefe8\redhat.java\jdt_ws\JAVA_64e4e41d\bin
Please enter the Username and Password
-----
ENTER USERNAME: shad08
ENTER PASSWORD: sahd@123#
Login Successfully!!!
PS C:\Users\hp\Desktop\JAVA>

```

PROBLEMS 45 OUTPUT TERMINAL DEBUG CONSOLE

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\hp\Desktop\JAVA> & 'c:\Users\hp\.vscode\extensions\vscjava.vscode-java-debug-0.34.0\
' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-Dfile.encoding=UTF-8' '-cp' 'C:\
4c0d21fc24caefe8\redhat.java\jdt_ws\JAVA_64e4e41d\bin' 'Userauthentication'
Please enter the Username and Password
```

-----  
ENTER USERNAME:shad08

ENTER PASSWORD:shaddddd

Exception Occurred. . passwordException: Password should contain at least one special character

PS C:\Users\hp\Desktop\JAVA> █

PROBLEMS 45 OUTPUT TERMINAL DEBUG CONSOLE

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\hp\Desktop\JAVA> & 'c:\Users\hp\.vscode\extensions\vscjava.vscode-java
' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-Dfile.encoding=UTF
4c0d21fc24caefe8\redhat.java\jdt_ws\JAVA_64e4e41d\bin' 'Userauthentication'
Please enter the Username and Password
```

-----  
ENTER USERNAME:shad

ENTER PASSWORD:shad123

Exception Occurred. . UsernameException: Username must be atmost 5 characters!

PS C:\Users\hp\Desktop\JAVA> █

**PROGRAM NO : 20****AIM:**

Program to Find the average of N positive integers, raising a user defined exception for each negative input

**ALGORITHM :**

Step 1: Start

Step 2: Create a class named NegException that inherits Exception class with a constructor inside which we call the Exception class constructor and pass error message.

Step 3: Inside the main(), Read the limit of array

Step 4: Inside the try block, read the array and check if any element is less than 0

Step 5: If true, throw NegException with appropriate message.

Step 6: Calculate the average of the array and print it

Step 7: Inside the catch exception, Print "NEGATIVE EXCEPTION OCCURED"

Step 8: Stop

**PROGRAM CODE:**

average.java	<pre> import java.util.Scanner; class NegException extends Exception {     public NegException(String s)     {         super(s);     } } public class average {     public static void main(String[] args)     {         int i;         double sum=0,avg=0;         Scanner sc=new Scanner(System.in);         System.out.println("Enter the total count of numbers:");         int n=sc.nextInt();         for(i=1;i&lt;=n;i++)         {             try </pre>
--------------	---



```

        {
            System.out.println("Enter number"+i);
            int a=sc.nextInt();
            if(a<0)                //initate exception
            {
                i--;
                throw new NegException("Negative numbers
not allowed, Try again");
            }
            else
            {
                sum=sum+a;
            }
        }
        catch(NegException e)
        //action to be done is specified when a try block gets
        executed
        {
            System.out.println("NEGATIVE EXCEPTION
OCCURED:"+e);
        }
    }
    avg=sum/n;
    System.out.println("Average is "+avg);
    sc.close();
}
}

```

**OUTPUT:**

```

PROBLEMS 46 OUTPUT TERMINAL DEBUG CONSOLE

Enter number1
2
Enter number2
3
Enter number3
-3
NEGATIVE EXCEPTION OCCURED:NegException: Negative numbers not allowed, Try again
Enter number3
1
Enter number4
3
Average is 2.25
PS C:\Users\hp\Desktop\JAVA>

```

**PROGRAM NO : 21****AIM:**

Program to define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

**ALGORITHM :**

Step 1: Start

Step 2: Create a class named ThreadA that inherits Thread class with member function run()

Step 3: Inside run(), Print the multiplication table for 5

Step 4: Create a class named ThreadB that inherits Thread class with memebr function run()

Step 5: Inside run(),Print the prime numbers up to the limit of user's choice

Step 6: Inside the main(), create an object for the classes and call start() using each object

Step 7: Stop

**PROGRAM CODE:**

thread.java	<pre> import java.util.*; public class thread {     public static void main(String args[])throws     InterruptedException     {         ThreadA a=new ThreadA();         ThreadB b=new ThreadB();         a.start();         a.sleep(200);         b.start();         b.sleep(200);      }  } class ThreadA extends Thread {     public void run()     {         int n=5; </pre>
-------------	--

```

        System.out.println("Multiplication table Of
5***\n"+"n"+"n"+"-----");
        for(int i=1;i<=10;i++)
        {

            System.out.println("\t"+n+"X"+i+"="+n*i);
        }
        System.out.println("\n-----\n");
    }
}
class ThreadB extends Thread
{

    public void run()
    {
        int i,count,j;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the limit:");
        int l = s.nextInt();
        System.out.println("Prime numbers between 1 and " + l
+ " are:");
        for(i=1;i<=l;i++)
        {
            count=0;
            for(j=1;j<=i;j++)
            {
                if(i%j==0)
                {
                    count++;

                }
            }
            if(count==2)
            {
                System.out.println(i);
            }
        }

    }
}

```

**OUTPUT:**

```
\Users\hp\AppData\Roaming\Code\User\workspaceStorage\614d12f2db1e
Multiplication table of 5***

-----
      5X1=5
      5X2=10
      5X3=15
      5X4=20
      5X5=25
      5X6=30
      5X7=35
      5X8=40
      5X9=45
      5X10=50
-----

Enter the limit:
5
Prime numbers between 1 and 5 are:
2
3
5
PS C:\Users\hp\Desktop\JAVA> |
```

**PROGRAM NO : 22****AIM:**

Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)

**ALGORITHM :**

Step 1: Start

Step 2: Create another class Fibbonoci that implements Runnable interface with function run().

Step 3: Inside run(), Initialise n1 as 0,n2 as 1 and n3 as 0.

Step 4: Check if n<0,if true, print “Enter a positive number”, else go to step 7

Step 5: Repeat step 8 to 11 until n3>n

Step 6: Print n1

Step 7: n3=n1+n2

Step 8: n1=n2

Step 9: n2=n3

Step 10: Create a class named Even that implements Runnable interface with function run()

Step 11: Inside run(), we read the limit for printing even numbers and print it using for loop.

Step 12: Create object e of even and create an object t1 of Thread with its parameterized constructor passing e as parameter

Step 13: Call start() using t1

Step 14: Do the same for class odd with Thread object t2 and call start() using t2

Step 15: Stop

**PROGRAM CODE:**

Fibbonoci_Even.java	<pre>import java.util.Scanner; class Fibonacci implements Runnable {     int n,first,second,t;</pre>
---------------------	--

```

String str;
public Fibonacci(int num)
{
    n = num;
    first = 0;
    second = 1;
}
@Override
public void run()
{
    str = first+" "+second;
    for(int i=0;i<=n-3;i++)
    {
        t = first + second;
        first = second;
        second = t;
        str += " "+t;
    }
    System.out.println(str);
}
}
class Even implements Runnable
{
    int n;
    String str;
    public Even(int n)
    {
        this.n = n;
        str = "";
    }
    @Override
    public void run()
    {
        for(int i=0;i<n;i=i+2)
            if(i%2==0)
            {
                str+=i+" ";
            }
        System.out.println(str);
    }
}

public class Fibbonoci_Even {
    public static void main(String[] args) throws
    InterruptedException {
        int n1,n2;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the range to see the
        Fibanocci Series: ");
        n1 = sc.nextInt();

```

	<pre>Fibonacci fib = new Fibonacci(n1); Thread th = new Thread(fib); th.start(); Thread.sleep(400); System.out.println("Enter the range of even numbers: "); n2 = sc.nextInt(); Even e = new Even(n2); Thread th2 = new Thread(e); th2.start(); sc.close(); } }</pre>
--	---

**OUTPUT:**

```
Enter the range to see the Fibanocci Series:
5
0 1 1 2 3
Enter the range of even numbers:
20
0 2 4 6 8 10 12 14 16 18
PS C:\Users\hp> █
```

**PROGRAM NO : 23****AIM:**

Program to implement Producer/Consumer using ITC

**ALGORITHM :**

Step 1: Start

Step 2: In PC class (A class that has both produce and consume methods), a linked list of jobs and a capacity of the list is added to check that producer does not produce if the list is full.

Step 3: In Producer class, the value is initialized as 0.

Step 4: We have an infinite outer loop to insert values in the list. Inside this loop, we have a synchronized block so that only a producer or a consumer thread runs at a time.

An inner loop is there before adding the jobs to list that checks if the job list is full, the producer thread gives up the intrinsic lock on PC and goes on the waiting state.

Step 5: If the list is empty, the control passes to below the loop and it adds a value in the list.

Step 6: In the Consumer class, we again have an infinite loop to extract a value from the list. Inside, we also have an inner loop which checks if the list is empty.

Step 7: If it is empty then we make the consumer thread give up the lock on PC and passes the control to producer thread for producing more jobs.

Step 8: If the list is not empty, we go round the loop and removes an item from the list.

Step 9: In both the methods, we use notify at the end of all statements. The reason is simple, once you have something in list, you can have the consumer thread consume it, or if you have consumed something, you can have the producer produce something.

Step 10: sleep() at the end of both methods just make the output of program run in step wise manner and not display everything all at once so that you can see what actually is happening in the program.

Step 11: Stop

**PROGRAM CODE:**

ProducerConsumer.java	<pre>import java.util.ArrayList; import java.util.List; class Producer extends Thread {     List&lt;Integer&gt; list;     public Producer(List&lt;Integer&gt; list)</pre>
-----------------------	---



```

{
    this.list=list;
}
@Override
public void run(){
    try{
        synchronized(list){
            while(true){
                if(list.size()>0)
                    list.wait();
                else
                    produceItem();
            }
        }
    }catch(InterruptedException e){
        e.printStackTrace();
    }
}
private void produceItem() throws
InterruptedException{
    for(int i=1;i<=5;i++){
        Thread.sleep(1000);
        list.add(i);
        System.out.println("Added element in list by
producer=" +i);
    }
    list.notifyAll();
}
}
class Consumer extends Thread{
    List<Integer> list;
    public Consumer(List<Integer> list)
    {
        this.list=list;
    }
    public void run(){
        try{
            while(true){
                synchronized(list){
                    if(list.size()==0)
                        list.wait();
                    else
                        consume();
                }
            }
        }catch(InterruptedException e){
            e.printStackTrace();
        }
    }
}

```

```

private void consume() throws InterruptedException{
    while(!list.isEmpty()){
        Thread.sleep(1000);
        System.out.println("Removed element from list by
consumer="+list.remove(0));
    }
    list.notifyAll();

}
}
public class ProducerConsumer {
    public static void main(String[] args) {
        List<Integer> list=new ArrayList<Integer>();
        Producer p=new Producer(list);
        Consumer c=new Consumer(list);
        p.start();
        c.start();
    }
}

```

**OUTPUT:**

```

PROBLEMS 56 OUTPUT TERMINAL DEBUG CONSOLE
Removed element from list by consumer=2
Removed element from list by consumer=3
Removed element from list by consumer=4
Removed element from list by consumer=5
Added element in list by producer=1
Added element in list by producer=2
Added element in list by producer=3
Added element in list by producer=4
Added element in list by producer=5
Removed element from list by consumer=1
Removed element from list by consumer=2
Removed element from list by consumer=3
Removed element from list by consumer=4
Removed element from list by consumer=5
Added element in list by producer=1
Added element in list by producer=2
Added element in list by producer=3
Added element in list by producer=4
Added element in list by producer=5
Removed element from list by consumer=1
Removed element from list by consumer=2

```

**PROGRAM NO : 24****AIM:**

Program to create a generic stack and do the Push and Pop operations

**ALGORITHM :**

Step 1: Start

Step 2: Create a class named stack with data members as a(an array),top(set as -1),ch,item,i;  
a function named menu()

Step 3: Inside menu(), give choices to push,pop and display the stack

Step 4: If the choice is 1, then check whether the stack is full, else add an element into the stack.

Step 5: If the choice is 2, then check whether the stack is empty, else delete an element into the stack.

Step 6: If the choice is 3, then check whether the stack is empty, else print all the elements in the stack.

Step 7: If the choice is greater than 4, then print "Invalid option".

Step 8: Inside the main(), create an object of type stack and call the menu() function.

Step 9: Stop

**PROGRAM CODE:**

GenericStack.java	<pre> import java.util.Scanner;  class stack{     int a[]=new int[20];     int choice,top=-1,item;      public void stack_operation()     {         System.out.println("enter the size of the stack");          Scanner s=new Scanner(System.in);         int n=s.nextInt();         do { </pre>
-------------------	--

	<pre> System.out.println("1.push\n2.pop\n3.display\n4.ex it"); choice:"); choice=s.nextInt(); switch(choice) { case 1 :if(top&gt;=n-1) { System.out.println("Stack Overflow"); } else { System.out.println("\nEnter the item to be pushed : "); item=s.nextInt(); top=top+1; a[top]=item; break; }  case 2:if(top== -1) { System.out.println("Stack Underflow");} else{ item = a[top]; System.out.println("Poped item is:" + item); top = top-1; break; } case 3: if( top &lt; 0) { System.out.println("\nStack is empty\n"); }  else { System.out.println("\nStack elements :\n\n"); for(int i=top;i&gt;=0;i--) { System.out.println(a[i] ); System.out.println("\n"); } break; } </pre>
--	---

	<pre>                                 case 4:                                 break;                                  default: System.out.println("Invalid Choice");                                 }                                  }while(choice!=4);                                 }                                 }                                 public class Genericstack {                                 public static void main(String args[])                                 {                                 stack ob=new stack();                                 ob.stack_operation();                                 }                                 } </pre>
--	--

**OUTPUT:**

```

enter the size of the stack
3
1.push
2.pop
3.display
4.exit
enter the choice:
1

Enter the item to be pushed :
42
1.push
2.pop
3.display
4.exit
enter the choice:
1

Enter the item to be pushed :
20
1.push
2.pop
3.display
4.exit
enter the choice:
2
Poped item is:20
1.push
2.pop
3.display
4.exit
enter the choice:
3

Stack elements :

42

```

**PROGRAM NO : 25****AIM:**

Program by using generic method perform Bubble sort.

**ALGORITHM :**

Step 1: Start

Step 2: Read number of numbers(N) to sort.

Step 3: Read the numbers

Step 4: Repeat step 5 for i=0 to N-1

Step 5: Repeat for j=i+1 to N

Step 6: Check if array[i] > array[j],

Step 7: if Step 6 true, swap them. End of inner loop. End of outer loop.

Step 8: Print the sorted array

**PROGRAM CODE:**

GenericStack.java	<pre> import java.util.Scanner; public class bubblesrt {     public static void main(String[] args) {         int n, i, j, temp;         Scanner sc = new Scanner(System.in);          System.out.println("Enter the number of integers to sort:");         n = sc.nextInt();          int array[] = new int[n];          System.out.println("Enter " + n + " numbers: ");          for (i = 0; i &lt; n; i++)             array[i] = sc.nextInt();          for (i = 0; i &lt; ( n - 1 ); i++) {             for (j = 0; j &lt; n - i - 1; j++) {                  if (array[j] &gt; array[j+1])                 {                     temp = array[j];                     array[j] = array[j+1]; </pre>
-------------------	--

	<pre>        array[j+1] = temp;     } } } System.out.println("Sorted list of numbers:");  for (i = 0; i &lt; n; i++)     System.out.println(array[i]); } }</pre>
--	--

**OUTPUT:**

```
Enter the number of integers to sort:
4
Enter 4 numbers:
21
6
15
7
Sorted list of numbers:
6
7
15
21
PS C:\Users\hp> █
```

**PROGRAM NO : 26****AIM:**

Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

**ALGORITHM :**

Step 1:Start

Step 2: Create an object of the class ArrayList.

Step 3: Adding elements to the object of ArrayList using method add() and display.

Step 4: Remove elements of object of ArrayList using method remove() and display .

Step 5: Sort elements of object of ArrayList using method sort() and display.

Step 6: Getting object of list which is present at the specified index using method get() and display.

Step 7: Checking whether an element is present in list using method contains() and display True or False.

Step 8: Display the size of list using the method size().

Step 9: Clear List using the method clear().

**PROGRAM CODE:**

ArrayList.java	<pre> import java.util.ArrayList; import java.util.Collections; public class Arraylistprog {     public static void main(String[] args) {          // Creating ArrayList of type "String" which         means we can only add "String" elements          ArrayList&lt;String&gt; obj = new         ArrayList&lt;String&gt;();          //adding elements to an ArrayList          obj.add("red"); </pre>
----------------	--



	<pre> obj.add("black"); obj.add("green"); obj.add("yellow");  // Displaying elements  System.out.println("\n Original ArrayList:"); for(String str:obj)     System.out.println(str);  //Add element at the given index or replace  obj.add(1, "majanda");  // Displaying elements  System.out.println("\n ArrayList after add operation:"); for(String str:obj)     System.out.println(str);  //Remove elements from ArrayList  obj.remove("green");  // Displaying elements  System.out.println("\n ArrayList after remove operation:"); for(String str:obj)     System.out.println(str);  //Remove element from the specified index  obj.remove(3);  // Displaying elements  System.out.println("\n Final ArrayList:"); for(String str:obj)     System.out.println(str);  //Sorting the ArrayList  Collections.sort(obj);  System.out.println("\n ArrayList after sorting:"); for (String str : obj)     System.out.println(str); </pre>
--	--

```
//returns the object of list which is present at the
specified index

// obj.get(2);

System.out.println("\n Object at index
2:"+obj.get(2));

// Checks whether the object is in the ArrayList

// obj.contains();

System.out.println("\n violet is in the ArrayList
:"+obj.contains("violet"));
System.out.println("\n red is in the ArrayList
:"+obj.contains("red"));

//Size of the ArrayList

//obj.size();

System.out.println("\n Size of the
ArrayList:"+obj.size());

// remove ArrayList

obj.clear();

System.out.println("\n** ArrayList Removed
**");

}

}
```

**OUTPUT:**

```
Original ArrayList:
red
black
green
yellow

ArrayList after add operation:
red
majanda
black
green
yellow

ArrayList after remove operation:
red
majanda
black
yellow

Final ArrayList:
red
majanda
black

ArrayList after sorting:
black
majanda
red

Object at index 2:red
violet is in the ArrayList :false
red is in the ArrayList :true

Size of the ArrayList:3

** ArrayList Removed **
PS C:\Users\hp\Desktop\JAVA> █
```

**PROGRAM NO : 27****AIM:**

Program to remove all the elements from a linked list

**ALGORITHM :**

Step 1:Start

Step 2: Create an object of the class linkedlist.

Step 3: Adding elements to the linked list using method add().

Step 4: Remove all the elements of linkedlist using method clear().

Step 5:Display linkedlist

**PROGRAM CODE:**

Linkedlist_elements _removal.java	<pre> import java.util.*; public class Linkedlist_elemnts_removal {     public static void main(String args[]) {         LinkedList&lt;String&gt; list=new LinkedList&lt;String&gt;();         Scanner sc =new Scanner(System.in);         System.out.println("enter the number of items to be added");          int num=sc.nextInt();         System.out.print("Add items ");         for(int i=0;i&lt;num;i++)         {              String s=sc.next();             list.add(s);         }         System.out.print(" ");         System.out.print(" ");         System.out.println("liked list after adding");          Iterator&lt;String&gt; itr=list.iterator();         while(itr.hasNext()){             System.out.println(itr.next());         }         list.clear();         System.out.println("Linked list After removing: " + list);       </pre>
--------------------------------------	--

	}
	}

**OUTPUT:**

```
enter the number of items to be added
4
Add items 52
21
36
41
    liked list after adding
52
21
36
41
Linked list After removing: []
PS C:\Users\hp> █
```

**PROGRAM NO : 28****AIM:**

Program to remove an object from the Stack when the position is passed as parameter

**ALGORITHM :**

Step 1:Start

Step 2: Create an object of the class Stack.

Step 3: Adding elements to the stack using method add().

Step 4: Remove the element of stack at position 'pos' using method remove(pos).

Step 5:Display removed element and Stack.

**PROGRAM CODE:**

Stack_obj_remove .java	<pre> import java.util.*; public class stack_obj_remove {     public static void main(String[] args)     {         Stack&lt;Integer&gt; st=new Stack&lt;Integer&gt;();         Scanner sc=new Scanner(System.in);         System.out.print("how many Elements do you want to add: ");         int num=sc.nextInt();         System.out.print("Add element: ");         for(int i=0;i&lt;num;i++)         {             int s=sc.nextInt();             st.add(s);         }         System.out.println("Original Stack:"+st);         System.out.println("Enter the index to be removed:");         int index = sc.nextInt();          int rm = st.remove(index);          System.out.println("Removed Element is:"+rm);          System.out.println("\nStack after remove:\n"+st);     } </pre>
---------------------------	--

	}
--	---

**OUTPUT:**

```
how many Elements do you want to add: 4
Add element: 21
41
26
53
Original Stack:[21, 41, 26, 53]
Enter the index to be removed:
3
Removed Element is:53

Stack after remove:
[21, 41, 26]
PS C:\Users\hp> █
```

**PROGRAM NO : 29****AIM:**

Program to demonstrate the creation of queue object using the PriorityQueue class

**ALGORITHM :**

Step 1:Start

Step 2: Create an object of the class PriorityQueue.

Step 3: Adding elements to the PriorityQueue using method add().

Step 5:Display PriorityQueue

**PROGRAM CODE:**

Priority_Queue.java	<pre> import java.util.*; public class Priority_Queue {     public static void main(String[] args) {         PriorityQueue&lt;String&gt; queue=new PriorityQueue&lt;String&gt;();         Scanner sc=new Scanner(System.in);         System.out.println("Enter Number Of elements ");         int n=sc.nextInt();         System.out.println("Enter the elements ");         for(int i =0;i&lt;n;i++)         {             String st=sc.next();             queue.add(st);          }         System.out.println("head:"+queue.element());         System.out.println("head:"+queue.peek());         System.out.println("Iterating the queue elements\n ");          Iterator itr=queue.iterator();         while(itr.hasNext()){             System.out.println(itr.next());         }         queue.remove();         queue.poll();         System.out.println("After removing two elements \n");          Iterator&lt;String&gt; itr2=queue.iterator();         while(itr2.hasNext()){             System.out.println(itr2.next()); </pre>
---------------------	--



	}
	}

**OUTPUT:**

```
Enter Number Of elements
4
Enter the elements
21
42
63
10
head:10
head:10
Iterating the queue elements

10
21
63
42
After removing two elements

42
63
PS C:\Users\hp> █
```

**PROGRAM NO : 30**

**AIM:** Program to demonstrate the addition and deletion of elements in deque.

**ALGORITHM :**

Step 1: Start

Step 2: Create an object of the class Deque.

Step 3: Adding elements to the queue using method add().

Step 4: Removing elements of queue using method pop().

Step 5: Display Queue

**PROGRAM CODE:**

Deque .java	<pre> import java.util.*; public class deque {     public static void main(String[] args) {         int choice;         String element;         Deque&lt;String&gt; deque= new LinkedList&lt;String&gt;();         Scanner sc = new Scanner(System.in);         do         {             System.out.println("1.INSERT THE ELEMENT AT FIRST");             System.out.println("2.INSERT THE ELEMENT AT LAST");             System.out.println("3.REMOVE THE ELEMENT AT FIRST");             System.out.println("4.REMOVE THE ELEMENT AT LAST");             System.out.println("5.DISPLAY");             System.out.println("6.EXIT");             System.out.println("\nENTER THE CHOICE:");             choice= sc.nextInt();             sc.nextLine();             switch(choice)             {                 case 1: System.out.println("ENTER THE ELEMENT TO BE INSERTED AT FIRST:");                     element = sc.next();                     deque.addFirst(element);                     break;                 case 2: System.out.println("ENTER THE ELEMENT TO BE INSERTED AT LAST:");                     element = sc.next();                     deque.addLast(element);                     break;                 case 3: System.out.println("ELEMENT DELETED FROM THE FIRST POSITION"); </pre>
-------------	--

```

        deque.removeFirst();
        break;
    case 4: System.out.println("ELEMENT DELETED FROM THE
LAST POSITION");
        deque.removeLast();
        break;
    case 5: System.out.println("PRINT ELEMENTS:");
        System.out.println(deque+"\n");
        break;
    case 6: System.exit(0);
        break;
    default: System.out.println("INVALID CHOICE");
    }
}while(true);
}
}

```

**OUTPUT:**

```

1.INSERT THE ELEMENT AT FIRST
2.INSERT THE ELEMENT AT LAST
3.REMOVE THE ELEMENT AT FIRST
4.REMOVE THE ELEMENT AT LAST
5.DISPLAY
6.EXIT

ENTER THE CHOICE:
1
ENTER THE ELEMENT TO BE INSERTED AT FIRST:
54
1.INSERT THE ELEMENT AT FIRST
2.INSERT THE ELEMENT AT LAST
3.REMOVE THE ELEMENT AT FIRST
4.REMOVE THE ELEMENT AT LAST
5.DISPLAY
6.EXIT

ENTER THE CHOICE:
2
ENTER THE ELEMENT TO BE INSERTED AT LAST:
24
1.INSERT THE ELEMENT AT FIRST
2.INSERT THE ELEMENT AT LAST
3.REMOVE THE ELEMENT AT FIRST
4.REMOVE THE ELEMENT AT LAST
5.DISPLAY
6.EXIT

ENTER THE CHOICE:
2
ENTER THE ELEMENT TO BE INSERTED AT LAST:
87
1.INSERT THE ELEMENT AT FIRST
2.INSERT THE ELEMENT AT LAST
3.REMOVE THE ELEMENT AT FIRST
4.REMOVE THE ELEMENT AT LAST
5.DISPLAY
6.EXIT

ENTER THE CHOICE:
5
PRINT ELEMENTS:
[54, 24, 87]

```

ENTER THE CHOICE:

5

PRINT ELEMENTS:

[54, 24, 87]

- 1.INSERT THE ELEMENT AT FIRST
- 2.INSERT THE ELEMENT AT LAST
- 3.REMOVE THE ELEMENT AT FIRST
- 4.REMOVE THE ELEMENT AT LAST
- 5.DISPLAY
- 6.EXIT

ENTER THE CHOICE:

3

ELEMENT DELETED FROM THE FIRST POSITION

- 1.INSERT THE ELEMENT AT FIRST
- 2.INSERT THE ELEMENT AT LAST
- 3.REMOVE THE ELEMENT AT FIRST
- 4.REMOVE THE ELEMENT AT LAST
- 5.DISPLAY
- 6.EXIT

ENTER THE CHOICE:

5

PRINT ELEMENTS:

[24, 87]

- 1.INSERT THE ELEMENT AT FIRST
- 2.INSERT THE ELEMENT AT LAST
- 3.REMOVE THE ELEMENT AT FIRST
- 4.REMOVE THE ELEMENT AT LAST
- 5.DISPLAY
- 6.EXIT

ENTER THE CHOICE:

**PROGRAM NO : 31****AIM:** Program to demonstrate the creation of Set object using the LinkedHashSet class**ALGORITHM :**

Step 1:Start

Step 2: Create an object of the class LinkedHashSet.

Step 3: Adding elements to the HashSet using method add().

Step 4:Display LinkedHashSet.

**PROGRAM CODE:**

Linked_hash_set.java	<pre> import java.util.LinkedHashSet; public class Linked_hash_set {     public static void main(String[] args) {         LinkedHashSet&lt;String&gt; LHS = new LinkedHashSet&lt;String&gt;();         LHS.add("Shad");         LHS.add("Vasim");         LHS.add("Fahad");         LHS.add("Adharsh");         LHS.add("Rashi");         LHS.add("Asshu");         LHS.add("Shameem");         LHS.add("Hudaif");         System.out.println("Size of LinkedHashSet = " +LHS.size());         System.out.println("The LinkedHashSet is"+LHS);     } } </pre>
----------------------	---

**OUTPUT:**

```

Size of LinkedHashSet = 8
The LinkedHashSet is[Shad, Vasim, Fahad, Adharsh, Rashi, Asshu, Shameem, Hudaif]
PS C:\Users\hp>

```

**PROGRAM NO : 32****AIM:** Write a Java program to compare two hash set**ALGORITHM :**

Step 1:Start

Step 2: Create two objects of the class LinkedHashSet.

Step 3: Adding elements to the two objects of LinkedHashSet using method add().

Step 4:Checking weather elements of first Hashset is present in second Hashset.

Step 5:If Step 4 is true print Yes,else print No.

**PROGRAM CODE:**

hashset_16.java	<pre> import java.util.HashSet;  public class hashset_16 {     public static void main(String[] args)     {         HashSet&lt;String&gt; hash_set = new HashSet&lt;String&gt;();         hash_set.add("Shad");         hash_set.add("Robin");         hash_set.add("Nihal");         hash_set.add("Nithin");          HashSet&lt;String&gt;hash_set2 = new HashSet&lt;String&gt;();         hash_set2.add("Shad");         hash_set2.add("Amjiyad");         hash_set2.add("Nihal");         hash_set2.add("Rasik");          System.out.println("Comparing the two hash sets");          HashSet&lt;String&gt;result_set = new HashSet&lt;String&gt;();         for (String element : hash_set)         {             System.out.println(hash_set2.contains(element) ? "Sets are same:Yes" : "Sets are same:No");         }     } } </pre>
-----------------	--

**OUTPUT:**

```
Comparing the two hash sets  
Sets are same:No  
Sets are same:Yes  
Sets are same:No  
Sets are same:Yes  
PS C:\Users\hp>
```

**PROGRAM NO : 33**

**AIM:** Program to demonstrate the working of Map interface by adding, changing and removing elements.

**ALGORITHM :**

Step 1:Start

Step 2:Initialization of a Map using Generics.

Step 3:Adding values into map using method put() and display.

Step 4:Updating values using method put() by mentioning index of value and display.

Step 5:Removing values from map using method remove() and display.

**PROGRAM CODE:**

map_17.java	<pre> import java.util.*; public class map_17 {     public static void main(String args[])     {          Map&lt;Integer, String&gt; mp = new HashMap&lt;&gt;();         //Inserting         System.out.println("Enter the limit:");         Scanner sc = new Scanner(System.in);         int n= sc.nextInt();         System.out.println("Enter the rollno and name:");         while(n!=0) {              int x= sc.nextInt();             String s= sc.next();             mp.put(x, s);             n--;         }          System.out.println("Initial Map:"+mp);         mp.put( 34, "Shad");         //Updating         System.out.println("Updated Map:"+mp);         //Removing         System.out.println("Enter the Roll number to be removed:");         int r=sc.nextInt();         mp.remove(r);         // Final Map         System.out.println("Final Map:"+mp); </pre>
-------------	---



	}
	}

**OUTPUT:**

```
Enter the limit:
2
Enter the rollno and name:
101
shad
102
amji
Initial Map:{101=shad, 102=amji}
Updated Map:{34=Shad, 101=shad, 102=amji}
Enter the Roll number to be removed:
█
```

**PROGRAM NO : 34****AIM:** Program to Convert HashMap to TreeMap**ALGORITHM :**

Step 1: Get the HashMap to be converted.

Step 2: Create a new TreeMap

Step 3: Pass the hashMap to putAll() method of treeMap

Step 4: Return the formed TreeMap

**PROGRAM CODE:**

convert_map_18.java	<pre> import java.util.*; public class convert_map_18 {     public static void main(String args[]) {          Map&lt;String, String&gt; map = new HashMap&lt;&gt;();         System.out.println("Enter the limit:");         Scanner inp = new Scanner(System.in);         int n= inp.nextInt();         System.out.println("Enter the rollno and name");         while(n!=0) {              String e= inp.next();             String s= inp.next();             map.put(e, s);             n--;         }          System.out.println("HashMap:"+map);         Map&lt;String, String&gt; treeMap = new TreeMap&lt;&gt;();         treeMap.putAll(map);         System.out.println("TreeMap:"+treeMap);     } } </pre>
---------------------	--

**OUTPUT:**

```
Enter the limit:
4
Enter the rollno and name
101
shad
102
rijash
103
tanish
104
emin
HashMap:{101=shad, 102=rijash, 103=tanish, 104=emin}
TreeMap:{101=shad, 102=rijash, 103=tanish, 104=emin}
PS C:\Users\hp> █
```