

OBJECT ORIENTED PROGRAMMING LAB

Submitted by,

RASIKA V V

TKM20MCA2029

COURSE OUTCOME 1

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

Step1: Define a class 'product'

Step2: Initialise product code,name,price

Step3: Create three objects in the class and add the values of each data members into Objects.

Step4:Using if condition check which object has the lowest price and print it.

PROGRAM CODE

product.java	<pre>public class product { int pcode , price; String pname; public static void main(String [] args) { product obj1 = new product(); product obj2 = new product(); product obj3 = new product(); obj1.pcode = 101; obj1.pname = "HP"; obj1.price = 50000; obj2.pcode = 102; obj2.pname = "MI"; obj2.price = 35000; obj3.pcode = 103; obj3.pname = "Asus"; obj3.price = 20000; } }</pre>
--------------	--

	<pre>System.out.println("Print details of product which has lowest price : "); if(obj1.price < obj2.price && obj1.price < obj3.price) { System.out.println("Product Code : "+ obj1.pcode+ "\n" + "Product Name : "+ obj1.pname+ "\n" + "Product Price: " + obj1.price); } else if (obj2.price < obj1.price && obj2.price < obj3.price) { System.out.println("Product Code "+ obj2.pcode+ "\n" + "Product Name : "+ obj2.pname+ "\n" + "Product Price: " + obj2.price); } else { System.out.println("Product Code : "+ obj3.pcode+ "\n" + "Product Name "+ obj3.pname+ "\n" + "Product Price : " + obj3.price); } }</pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> javac product.java
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> java product
Print details of product which has lowest price :
Product Code : 103
Product Name Asus
Product Price : 20000
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> █
```

PROGRAM NO: 2

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

ALGORITHM

Step1: Define a class matrix

Step2: Enter the number of rows and columns.

Step3: Enter the elements of first matrix store them into the two-dimensional array x[][].

i indicates row number, j indicates column number.

Step4: Similarly second matrix elements into y[][].

Step5: Add the two matrices using for loop.

for i=0 to i<row

For j=0 to j<column

x[i][j]+y[i][j] and store it into the matrix total[i][j]

Step6: Print total[i][j]

PROGRAM CODE

matrix.java	<pre>import java.util.Scanner; public class matrix { public static void main(String args[]) { int row, column,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:"); column = in.nextInt(); int x[][] = new int[row][column]; int y[][] = new int[row][column]; int total[][] = new int[row][column]; System.out.println("Enter the elements of first matrix:"); for (i= 0 ; i < row ; i++) { for (j= 0 ; j < column ;j++) x[i][j] = in.nextInt(); System.out.println();</pre>
-------------	---

	<pre> } System.out.println("Enter the elements of second matrix"); for (i= 0 ; i < row ; i++) { for (j= 0 ; j < column ;j++) y[i][j] = in.nextInt(); System.out.println(); } for (i= 0 ; i < row ; i++) for (j= 0 ; j < column ;j++) total[i][j] = x[i][j] + y[i][j] ; System.out.println("Sum of matrices:"); for (i= 0 ; i < row ; i++) { for (j= 0 ; j < column ;j++) System.out.print(total[i][j]+"\\t"); System.out.println(); } } }</pre>
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> javac matrix.java
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> java matrix
Enter the number of rows:
2
Enter the number columns:
2
Enter the elements of first matrix:
1
2

3
4

Enter the elements of second matrix
5
6

7
8

Sum of matrices:
6      8
10     12
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> █
```

PROGRAM NO: 3

AIM: Add complex numbers.

ALGORITHM

Step1: Define a class “Complexnumbersum” and datamembers are real and imaginary number.

Step2: Define a constructor used to initialize instance variable ‘real’ and ‘img’ in the class

Step3: Define a function Complexnumbersum to add the complex numbers

PROGRAM CODE

Complexnumbersum.java	<pre>public class Complexnumbersum { double real,img; Complexnumbersum(double r,double i){ this.real=r; this.img=i; } public static Complexnumbersum sum(Complexnumbersum x1,Complexnumbersum x2) { Complexnumbersum temp=new Complexnumbersum(0,0); temp.real=x1.real+x2.real; temp.img=x1.img+x2.img; return temp; } public static void main(String[] args) { Complexnumbersum x1=new Complexnumbersum(6.2,5); Complexnumbersum x2=new Complexnumbersum(2.3,4.7); Complexnumbersum temp=sum(x1,x2); System.out.println("Sum is:"+ temp.real+" + " + temp.img+"i") } }</pre>
-----------------------	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> javac Complexnumbersum.java
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> java Complexnumbersum
Sum is:8.5 + 9.7i
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> █
```

PROGRAM NO: 4

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

ALGORITHM

Step1: Define a class 'symmetricmatrix'

Step2: Enter the number of rows and columns of the matrix.

Step3: Enter the elements and store the elements into matrix[][]

Step4: Print the matrix.

Step5: Check that the given matrix is symmetric or not.

if(matrix[i][j] = matrix[j][i])

then 'Symmetric'

Step6: Print the matrix is symmetric or not.

PROGRAM CODE

symmetricmatrix.java

```
import java.util.Scanner;

public class symmetricmatrix
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no. of rows : ");
        int rows = sc.nextInt();
        System.out.println("Enter the no. of columns : ");
        int cols = sc.nextInt();
        int matrix[][] = new int[rows][cols];
        System.out.println("Enter the elements :");

        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                matrix[i][j] = sc.nextInt();
            }
        }
        System.out.println("input matrix is:");
```

```

for (int i = 0; i < rows; i++)
{
    for (int j = 0; j < cols; j++)
    {
        System.out.print(matrix[i][j]+"\\t");
    }

    System.out.println();
}

if(rows != cols)
{
    System.out.println("The given matrix is not a square matrix, so
it can't be symmetric.");
}
else
{
    boolean symmetric = true;

    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            if(matrix[i][j] != matrix[j][i])
            {
                symmetric = false;
                break;
            }
        }
    }

    if(symmetric)
    {
        System.out.println("The given matrix is symmetric");
    }
    else
    {
        System.out.println("The given matrix is not symmetric");
    }
}
}
}

```

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> javac symmetricmatrix.java
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> java symmetricmatrix
Enter the no. of rows :
3
Enter the no. of columns :
3
Enter the elements :
1
7
3
7
4
5
3
5
0
input matrix is:
1      7      3
7      4      5
3      5      0
The given matrix is symmetric
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> █
```

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

Step1: Define a class 'CPU' with data member price.

Step2: Define a class 'Processor' contain data members nocores, manufacturer.

Step3: Define a class 'RAM' contain memory and manufacturer as data members.

Step4: Create objects in corresponding classes and display its details.

PROGRAM CODE

CPU.jav a	<pre>import java.util.Scanner; import java.lang.String; public class CPU { double price; public class Processor{ float nocores; String manufacturer; void processorinfo(float a,String processortname) { nocores=a; manufacturer=processortname; System.out.println("\nInformation of Processor"); System.out.println("No. of cores:"+nocores+"\nManufacturer:" +manufacturer); } } static class RAM{ float memory; String manufacturer; void raminfo(float b,String ram) { memory=b; manufacturer=ram; System.out.println("\nInformation of RAM"); System.out.println("Memory:"+memory+"\nManufacturer:"+manufacturer); } } }</pre>
--------------	--

	<pre>public static void main(String[] args) { CPU obj=new CPU(); CPU.Processor obj1=obj.new Processor(); CPU.RAM obj2=new CPU.RAM(); Scanner sc=new Scanner(System.in); System.out.println("Enter price of CPU"); obj.price=sc.nextInt(); System.out.println("Enter Processor details:"); float a=sc.nextFloat(); Scanner sc1=new Scanner(System.in); String processorname=sc1.nextLine(); System.out.print("Enter RAM details:"); float b=sc.nextFloat(); String ram=sc1.nextLine(); sc.close(); sc1.close(); System.out.println("\nThe price of CPU is "+obj.price); obj1.processorinfo(a, processorname); obj2.raminfo(b, ram); } }</pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> javac CPU.java
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> java CPU
Enter price of CPU
3000
Enter Processor details:
4
intel
Enter RAM details:3
lenovo

The price of CPU is3000.0

Information of Processor
No. of cores:4.0
Manufacturer:intel

Information of RAM
Memory:3.0
Manufacturer:lenovo
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE1> █
```

COURSE OUTCOME 2

PROGRAM NO: 6

AIM: Program to Sort strings

ALGORITHM

Step1: Create a class 'Sort'

Step2: Enter the Strings

Step3: Select the element at first position in the list.

Step4: Compare the selected element with all other elements.

Step5: If any element is found smaller than the selected element, both are swapped.

Step6: Repeat the same procedure till the list is sorted.

PROGRAM CODE

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

	<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.print(str_arr[i]+","); } }</pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

```
Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE2> cmd /C "c:\Users\USER\.vscode\exten
-14.0.2\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=
be326ed0c75fc\redhat.java\jdt_ws\LAB-CYCLE2_4036c0\bin Sort "
Enter the number of strings:
3
Enter the strings:
rasika
aishwarya
devika
String after sorting:
aishwarya,devika,rasika,
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE2> █
```

PROGRAM NO: 7

AIM: Search an element in an array.

ALGORITHM

Step1: Create a class 'Search'

Step2: Enter the number of arrays elements.

Step3: Enter the array elements.

Step4: Search for an element.

Step5: If the element is found, display the position of the array found,
else display element not found

PROGRAM CODE

Search.java	<pre>import java.util.Scanner; public class Search { public static void main(String[] args) { int n,key,flag=0; Scanner sc=new Scanner(System.in); System.out.println("Enter the number of array elements: "); n=sc.nextInt(); int[] arr=new int[n]; Scanner sc1=new Scanner(System.in); System.out.println("Enter the array elemnts: "); for(int i=0;i<n;i++) { arr[i]=sc1.nextInt(); } System.out.println("Enter the element to be searched: "); Scanner sc2=new Scanner(System.in); key=sc2.nextInt(); for(int i=0;i<n;i++) { if(arr[i]==key) {</pre>
-------------	--

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

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE2> cmd /C "c:\Users\USER\.vscode\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8 -Djava.class.path=.\bin\Search.class Search "2"
```

Enter the number of array elements:

4

Enter the array elements:

7

4

2

9

Enter the element to be searched:

2

Element 2 is found at 2 position

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE2>
```

PROGRAM NO: 8

AIM: Perform string manipulations

ALGORITHM

Step1: Define a class 'String_manipulation'

Step2: Enter strings.

Step3: Perform string operations.

Step4: Concatenate strings.

Step5: Display the string with uppercase.

Step6: Compare two Strings

PROGRAM CODE

String_manipulation.java	<pre>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("Concatating 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 uppercase: "+str2.toUpperCase());</pre>
--------------------------	---

	<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>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE2> cmd /C "c:\Users\USER\.vscode\extensions\
-14.0.2\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8
be326ed0c75fc\redhat.java\jdt_ws\LAB-CYCLE2_4036c0\bin String_manipulation "
Enter string1 :
hello
Enter string2 :
welcome
Length of string2: 7
Character at position 3 of string1 :l
Concatating string1 and string2 : helloworld
Does string1 contains 'hai': false
Unicode of first character in string1: 104
Compare string1 and string2: 15
String1 in lowercase: hello
String2 in uppercase: WELCOME
String2 after removing space: welcome
String1 after replacing : hello
Is string1 empty?: false

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE2>█
```


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

Step1: Define a class Employee

Step2:Employee having attributes eNo,eName,eSalary.

Step3: Read n employee information.

Step4: Search for an Employee given eNo.

Step5:If the user provided 'eNo' is found, display the details of the corresponding employee.

PROGRAM CODE

Employee.java	<pre>import java.util.Scanner; public class Employee{ 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() { System.out.println("Employee no: "+eNo); System.out.println("Employee name: "+eName); } }</pre>
---------------	---

```

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();
        Employee e[]=new Employee[n];

        for(int i=0;i<n;i++)
        {
            e[i]=new Employee();
            e[i].getinfo();
        }
        System.out.println("The employee details
are:");
        for(int i=0;i<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<n;i++)
        {
            if(no==e[i].eNo)
            {
                e[i].display();
                flag=1;
                break;
            }
        }
        if(flag==0)
        {
            System.out.println("No such
employee");
        }
    }
}

```

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE2> cmd /C "c:\Users\USER\.vsco
-14.0.2\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -Dfile.e
be326ed0c75fc\redhat.java\jdt_ws\LAB-CYCLE2_4036c0\bin Employee "
Enter the no of employees:
3
Enter Employee number:
1
Enter Employee name:
Rasika
Enter Employee salary:
5000
Enter Employee number:
2
Enter Employee name:
Novrin
Enter Employee salary:
8000
Enter Employee number:
3
Enter Employee name:
Anila
Enter Employee salary:
6000
The employee details are:
Employee no: 1
Employee name: Rasika
Salary: 5000.0
Employee no: 2
Employee name: Novrin
Salary: 8000.0
Employee no: 3
Employee name: Anila
Salary: 6000.0
Enter employee no to display details:
1
Employee no: 1
Employee name: Rasika
Salary: 5000.0
```

COURSE OUTCOME 3

PROGRAM NO: 10

AIM: Area of different shapes using overloaded functions

ALGORITHM

Step1: Start

Step2: Define the main class

Step3: Define methods with the same methodname that performs the area operation for each shape

Step4: Display the areas of each shapes

PROGRAM CODE

Area.java	<pre>import java.util.Scanner; public class Area { float area(float a) { float sarea=a*a; System.out.println("Area of Square: "+sarea); return sarea; } float area(float a, float b) { float rarea=a*b; System.out.println("Area of Rectangle: "+rarea); return rarea; } double area(double c) { double carea=3.14*c*c;; System.out.println("Area of Circle: "+carea); return carea; } public static void main(String[] arg) { Area ob= new Area();</pre>
-----------	--

	<pre>Scanner sc= new Scanner(System.in); System.out.println("Enter sides of shapes:"); float a=sc.nextInt(); float b=sc.nextInt(); double c=sc.nextInt(); ob.area(a); ob.area(a,b); ob.area(c); } }</pre>
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

93af1954ad620\redhat.java\jdt_ws\LAB-CYCLE3_4036c1\bin Area "

Enter sides of shapes:

2

3

5

Area of Square: 4.0

Area of Rectangle: 6.0

Area of Circle: 78.5

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3>

PROGRAM NO: 11

AIM: 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

Step1: Create class 'employee' with the provided data members and define the Constructor.

Step2: Create another class 'Teacher' that performs inheritance of employee class and define constructor for the same.

Step3: Create array of objects in the corresponding class.

Step4: Display details for the number of teachers provided.

PROGRAM CODE

Singleinheritance.java	<pre>import java.util.Scanner; class Employee { int empid,salary; String name,address; public Employee() { Scanner sc1 = new Scanner(System.in); System.out.println(" Enter employee id : "); empid = sc1.nextInt(); System.out.println("Enter name : "); name = sc1.next(); System.out.println("Enter salary : "); salary = sc1.nextInt(); System.out.println("Enter address : "); address = sc1.next(); } } class Teacher extends Employee {</pre>
------------------------	--

	<pre> String dept,sub; public Teacher() { Scanner sc1 = new Scanner(System.in); System.out.println("Enter Department : "); dept = sc1.next(); System.out.println("Enter Subject : "); sub = sc1.next(); } public void display() { System.out.println("Emp id : " + empid); System.out.println("Name : " + name); System.out.println("Address: " + address); System.out.println("Salary : " + salary); System.out.println("Department : " + dept); System.out.println("Subject : " + sub); } } public class Singleinheritance { public static void main(String [] args) { int n; Scanner sc2 = new Scanner(System.in); System.out.println("Enter no of teachers : "); n = sc2.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("\nDetails of Employee no : " + (i+1)); obj[i].display(); } } } </pre>
--	--

--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> javac Singleinheritance.java
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> java Singleinheritance
Enter no of teachers :
2
Enter employee id :
1
Enter name :
Rasika
Enter salary :
10000
Enter address :
Kannur
Enter Department :
MCA
Enter Subject :
Java
Enter employee id :
2
Enter name :
Nandana
Enter salary :
8000
Enter address :
Kozhikod
Enter Department :
Mech
Enter Subject :
Mechanics

Details of Employee no : 1
Emp id : 1
Name : Rasika
Address: Kannur
Salary : 10000
Department : MCA
Subject : Java

Details of Employee no : 2
Emp id : 2
Name : Nandana
Address: Kozhikod
Salary : 8000
Department : Mech
Subject : Mechanics
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> █
```

PROGRAM NO: 12

AIM: 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

Step1: Create class 'person' with the provided data members and define the constructor

Step2: Create class 'employee' that performs inheritance of person class and another class 'teacher' that further inherits the properties of its former class.

Step3: Create an array of objects in the corresponding class.

Step4: Display the details of the specific teacher

PROGRAM CODE

Multilevel_inheritance.java	<pre>import java.util.Scanner; class Person { String name,gender,address; int age; public Person() { Scanner sc1 = new Scanner(System.in); System.out.println("Enter Name : "); name = sc1.next(); System.out.println("Enter Age : "); age = sc1.nextInt(); System.out.println("Enter Gender : "); gender = sc1.next(); System.out.println("Enter Address : "); address = sc1.next(); } } class Employee extends Person</pre>
-----------------------------	---

```

{
    int empid,salary;
    String quali,cname;

    public Employee()
    {
        Scanner sc2 = new Scanner(System.in);
        System.out.println("Enter Emp_id : ");
        empid = sc2.nextInt();
        System.out.println("Enter Salary : ");
        salary = sc2.nextInt();
        System.out.println("Enter Qualification : ");
        quali = sc2.next();
        System.out.println("Enter Company Name :
");
        cname = sc2.next();

    }
}

class Teacher extends Employee
{
    String sub,dept;
    int teacher_id;

    public Teacher()
    {
        Scanner sc3 = new Scanner(System.in);
        System.out.println("Enter Teacher_id : ");
        teacher_id = sc3.nextInt();
        System.out.println("Enter Subject: ");
        sub = sc3.next();
        System.out.println("Enter Department : ");
        dept = sc3.next();

    }

    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 : "+quali);
        System.out.println("Company Name :
"+cname);
        System.out.println("Teacher id :
"+teacher_id);
    }
}

```

	<pre> System.out.println("Subject : "+sub); System.out.println("Department : "+dept); } } public class Multilevel_inheritance { public static void main(String args[]) { int n; Scanner sc4 = new Scanner(System.in); System.out.println("Enter number of teachers : "); n = sc4.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("\nDetails of Employee No : " + (i+1)); obj[i].display(); } } } </pre>
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> javac Multilevel_inheritance.java
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> java Multilevel_inheritance
Enter number of teachers :
1
Enter Name :
Rasika
Enter Age :
25
Enter Gender :
Female
Enter Address :
Kannur
Enter Emp_id :
101
Enter Salary :
20000
Enter Qualification :
PG
Enter Company Name :
ABCD
Enter Teacher_id :
1
Enter Subject:
DBMS
Enter Department :
MCA

Details of Employee No : 1
Name : Rasika
Age : 25
Gender : Female
Address : Kannur
Emp id : 101
salary : 20000
Qualification : PG
Company Name : ABCD
Teacher id : 1
Subject : DBMS
Department : MCA
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> █
```

PROGRAM NO: 13

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

ALGORITHM

Step1: Create class 'publisher' and initializes its datamembers.

Step2: Create class book, literature, fiction. Each class inherit from their subsequent previous class and have its own data members.

Step3: Create an array of objects in the corresponding class.

Step4: Display the details of the books required.

PROGRAM CODE

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

	<pre> 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]; for(i=0;i<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<m;i++) { f[i]=new fiction(); </pre>
--	--

	<pre> } System.out.println(" Displaying literature books:\n"); for(i=0;i<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<m;i++) { System.out.println("Displaying details of book no: \t" +(i+1)); f[i].display(); } } </pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

Enter the no of literature books you need to store:

1

Enter publisher name:

schwartz

Enter book name:

hero

Enter Author name:

alex

category:literature

Enter the no of fictional books you need to store:

1

Enter publisher name:

reddit

Enter book name:

sciencfiction

Enter Author name:

reyan

category:fiction

Displaying literature books:

Displaying details of book no1

publishername:schwartz

book name:hero

Author name:alex

Displaying fictional books:

Displaying details of book no: 1

publishername:reddit

book name:sciencfiction

Author name:reyan

PS C:\Users\hp> □

PROGRAM NO: 14

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

ALGORITHM

Step1: Create class Student with datamembers roll,name,mark of each subject.

Step2: Create another class Sports that extends Student with data members Football score and cricket score.

Step3: Create a class Result inherited Student and Sports.

Step4:Display the academic and sports score of a student.

PROGRAM CODE

Finalresult.java	<pre>import java.util.Scanner; class student{ int roll; String name; int phy,eng,maths; student() { Scanner sc1= new Scanner(System.in); System.out.println("Enter the roll number:"); roll =sc1.nextInt(); System.out.println("Enter name:"); name=sc1.next(); System.out.println("Enter physics mark:"); phy =sc1.nextInt(); System.out.println("Enter english mark:"); eng =sc1.nextInt(); System.out.println("Enter maths mark:"); maths =sc1.nextInt(); } } class sports extends student { int fscore,cscore; sports()</pre>
------------------	--

	<pre> { Scanner sc2= new Scanner(System.in); System.out.println("Enter football score:"); fscore=sc2.nextInt(); System.out.println("Enter Cricket score:"); cscore=sc2.nextInt(); } } class Result extends sports { void display() { System.out.println("Academic Details"+"\\n"+"----- -----"); System.out.println("Name : " + name); System.out.println("Roll No : " + roll); System.out.println(""); System.out.println("MARKS" +"\\n" + "-----"); System.out.println("Physics :" + phy); System.out.println("English :" + eng); System.out.println("Maths :" + maths); System.out.println("Total subject mark:"+(phy+eng+maths)); System.out.println(""); System.out.println("SPORTS SCORE" +"\\n"+"----- -----"); System.out.println("Football : " + fscore); System.out.println("Cricket : " + cscore); System.out.println("Total Sports mark:"+(fscore+cscore)); } } public class Finalresult{ public static void main(String[] args) { Result rs =new Result(); rs.display(); } } </pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> javac Finalresult.java
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> java Finalresult
Enter the roll number:
101
Enter name:
John
Enter physics mark:
38
Enter english mark:
32
Enter maths mark:
40
Enter football score:
50
Enter Cricket score:
43
Academic Details
-----
Name : John
Roll No : 101

MARKS
-----
Physics :38
English :32
Maths :40
Total subject mark:110

SPORTS SCORE
-----
Football : 50
Cricket : 43
Total Sports mark:93
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> █
```

PROGRAM NO: 15

AIM: . 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

Step1:Create an interface 'calculation' that has methods to take inputs and compute area and perimeter.

Step2:Create class Circle and Rectangle that implements calculation.

Step3:Display the area and perimeter of circle or rectangle depending upon the choice the user selects.

PROGRAM CODE

Areaperime ter.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() { 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); } }</pre>
------------------------	---

```

    }
}
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();

```

	<pre> break; case 3: obj.perimeter(); break; case 4: return; default: System.out.println("Enter correct choice:"); } } }</pre>
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
1.Input the values
2.Find area
3.Find perimeter
4.Exit
```

```
Enter the choice:
```

```
1
```

```
Enter radius:
```

```
4
```

```
Enter length:
```

```
3
```

```
Enter breadth:
```

```
5
```

```
1.Input the values
2.Find area
3.Find perimeter
4.Exit
```

```
Enter the choice:
```

```
2
```

```
Area of the circle:50.24
```

```
Area of rectangle:15.0
```

```
1.Input the values
2.Find area
3.Find perimeter
4.Exit
```

```
Enter the choice:
```

```
3
```

```
Perimeter of the circle:25.12
```

```
Perimeter of rectangle:16.0
```

```
1.Input the values
2.Find area
3.Find perimeter
4.Exit
```

```
Enter the choice:
```

```
4
```

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3> █
```

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

Step1:Create an interface 'bill' that performs the calculation operations.

Step2:Create class 'details1' that implements the interface bill.

Step3: Create another class that extends first class 'details1'

Step4:Display the net amount by acquiring the data for the specific inputs.

PROGRAM CODE

```
Electricitybill.java
import java.text.SimpleDateFormat;
import java.util.Date;
interface bill
{
    void cal();
}
class details1 implements bill{
    int pid=101,q=2,uprice=25,t1;
    String name1="A";
    @Override
    public void cal()
    {
        t1=q*uprice;
    }
}
class details2 extends details1 {
    int pid2=102,q2=1,uprice2=100,t2;
    String name2="B";
    SimpleDateFormat f=new SimpleDateFormat("dd/MM/yy");
    Date d= new Date();
    public void cal()
```

	<pre> { super.cal(); t2=q2*uprice2; } public void display() { System.out.println("Order No.384\n"); System.out.println("Date: "+f.format(d)); System.out.println("\nProduct Id\tName\t\tQuantity\tunit price\t Total"); System.out.println("----- -----"); System.out.println(pid+"\t"+name1+"\t"+q+"\t"+uprice+"\t\ t"+t1); System.out.println(pid2+"\t"+name2+"\t"+q2+"\t"+uprice2 +"\t"+t2); System.out.println("----- -----"); System.out.println("\t\t\t\tNet.Amount"+" \t"+(t1+t2)); } } public class Electricitybill{ public static void main(String[] args) { details2 obj2=new details2(); obj2.cal(); obj2.display(); } } </pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3\QS_7> javac Electricitybill.java
```

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3\QS_7> java Electricitybill
```

Order No.384

Date: 09/09/21

Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100
Net.Amount				150

```
PS C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE3\QS_7> █
```

COURSE OUTCOME 4

PROGRAM NO: 17

AIM: 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

Step1: Create package named Graphics.

Step2: Inside the Graphics folder, create modules for finding the areas of rectangle, circle, triangle and square.

Step3: Outside the Graphics folder, write a program to access the modules mentioned above and print the output.

PROGRAM CODE

Shape.java	<pre>package Qs1.Graphics; import java.util.Scanner; interface cal { void carea(); void rarea(); void tarea(); void sarea(); } public class Shapes implements cal { Scanner sc=new Scanner(System.in); public void carea() { System.out.println("Enter radius:"); int r=sc.nextInt(); System.out.println("Area of circle:"+(Math.PI*r*r)); } public void rarea() { System.out.println("Enter length:");</pre>
------------	---

	<pre> int l=sc.nextInt(); System.out.println("Enter breadth:"); int b=sc.nextInt(); System.out.println("Area of Rectangle:"+l*b)); } public void tarea() { System.out.println("Enter the base:"); int b = sc.nextInt(); System.out.println("Enter the height:"); int h = sc.nextInt(); System.out.println("Area of the triangle:"+0.5*b*h)); } public void sarea() { System.out.println("Enter the side:"); int s = sc.nextInt(); System.out.println("Area of the square:"+s*s)); } } </pre>
Area.java	<pre> package Qs1; import Qs1.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>

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
Area of different Shapes
-----
Circle
-----
Enter radius:
3
Area of circle:28.274333882308138
Rectangle
-----
Enter length:
3
Enter breadth:
4
Area of Rectangle:12
Triangle
-----
Enter the base:
4
Enter the height:
5
Area of the triangle:10.0
Square
-----
Enter the side:
3
Area of the square:9

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

PROGRAM NO: 18

AIM: 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

Step1: Create a package named Arithmetic.

Step2: Create an interface 'cal' that has the methods to take inputs and compute calculation.

Step3: Inside the Arithmetic we have another module named 'Arithmeticoperation' that implements the interface cal.

Step3: Inside Arithmetic package, create modules to perform addition, subtraction, multiplication and division of two numbers.

Step4: Outside the folder write another program 'Operation' to access the above module and print the output.

PROGRAM CODE

Arithmeticoperation.java	<pre>package Qs2.Arithmetic; import java.util.Scanner; interface cal { void add(); void sub(); void mul(); void div(); } public class Arithmeticoperation 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> package Qs2; import Qs2.Arithmetic.Arithmeticoperation; public class Operation { public static void main(String args[]) { Arithmeticoperation obj=new Arithmeticoperation(); 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(); </pre>

	<pre>System.out.println("\nDivision"+"n"+"-----"); obj.div(); } }</pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
9db25f711b552\redhat.java\jdt_ws\LAB-CYCLE4_4036c2\bin Qs2.Operation "
ARITHMETIC OPERATIONS
```

Addition

Enter first number:

5

Enter second number:

2

Sum=7

Subtraction

Enter first number:

10

Enter second number:

5

Difference=5

Multiplication

Enter first number:

6

Enter second number:

2

Product=12

Division

Enter first number:

20

Enter second number:

2

Quotient=10

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>

PROGRAM NO: 19

AIM: Write a user defined exception class to authenticate the user name and password.

ALGORITHM

Step1: Create a class named UsernameException that inherits Exception class with a constructor that calls Exception class constructor and pass error message.

Step2: Create a class named passwordException that inherits Exception class with a constructor that calls Exception class constructor and pass error message.

Step3: Create a class Userauthentication, Then inside the main(), Enter username and password.

Step4: 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

Step5: Catch block is used to handle the Exception by declaring the type of exception with in the parameter.

PROGRAM CODE

Userauthentication.java	<pre>package Qs_3; import java.util.Scanner; class passwordException extends Exception { private static final long serialVersionUID = 1L; passwordException(String s){ super(s); } } class UsernameException extends Exception { private static final long serialVersionUID = 1L;</pre>
-------------------------	---

	<pre> 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("/") </pre>
--	--

	<pre> 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); } } } </pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4> cmd /C "c:\Users\USER\.vscode\extensions
-14.0.2\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=U
9db25f711b552\redhat.java\jdt_ws\LAB-CYCLE4_4036c2\bin Qs3.Userauthentication "
Please enter the Username and Password
-----
ENTER USERNAME:Rasikavv
ENTER PASSWORD:Rasika@2468
Login Successfully!!!

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4> cmd /C "c:\Users\USER\.vscode\exten
-14.0.2\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding:
9db25f711b552\redhat.java\jdt_ws\LAB-CYCLE4_4036c2\bin Qs3.Userauthentication
Please enter the Username and Password
-----
ENTER USERNAME:Rasikavv
ENTER PASSWORD:rasika
Exception Occurred. . Qs3.passwordException: Please enter a strong password

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4> cmd /C "c:\Users\USER\.vscode\extensions\vscjava.vscod
-14.0.2\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8 -cp C:\Users\US
9db25f711b552\redhat.java\jdt_ws\LAB-CYCLE4_4036c2\bin Qs3.Userauthentication "
Please enter the Username and Password
-----
ENTER USERNAME:Rasika
ENTER PASSWORD:rasika1234
Exception Occurred. . Qs3.passwordException: Password should contain at least one special character

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4> cmd /C "c:\Users\USER\.vscode\extensions'  
-14.0.2\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8  
9db25f711b552\redhat.java\jdt_ws\LAB-CYCLE4_4036c2\bin Qs3.Userauthentication "  
Please enter the Username and Password
```

ENTER USERNAME:Rasi

ENTER PASSWORD:Rasika@1234

Exception Occurred. . Qs3.UsernameException: Username must be atmost 5 characters!

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>█

PROGRAM NO: 20

AIM: Find the average of N positive integers, raising a user defined exception for each negative input.

ALGORITHM

Step1:Create a class 'NegException' that inherits Exception class with a constructor inside which we call the Exception class constructor and pass error message.

Step2:Create a class 'Average' ,Inside the main(), Read the limit of array.

Step3:Inside the try block read the array and check if any element is less than 0.

Step4: If true ,throw NegException with appropriate message.

Step5:Calculate the average of the array and print it.

Step6:Inside the catch exception print 'NEGATIVE EXCEPTION OCCURRED'.

PROGRAM CODE

Average.java	<pre>package Qs4; 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 n numbers:"); int n=sc.nextInt(); for(i=1;i<=n;i++) { try { System.out.println("Enter number"+i);</pre>
--------------	---

	<pre>int a=sc.nextInt(); if(a<0) { i--; throw new NegException("Negative numbers not allowed, Try again"); } else { sum=sum+a; } } catch(NegException e) { System.out.println("NEGATIVE EXCEPTION OCCURED:"+e); } } avg=sum/n; System.out.println("Average is "+avg); sc.close(); } }</pre>
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.19041.329]

(c) 2020 Microsoft Corporation. All rights reserved.

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4> cmd /C "c:\Users\USER\.vscode\extensions\
:\Program Files\Java\jdk-14.0.2\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessage
g\Code\User\workspaceStorage\a2c047bfde05f988eca9db25f711b552\redhat.java\jdt_ws\LAB
Enter n numbers:
```

4

Enter number1

1

Enter number2

-6

NEGATIVE EXCEPTION OCCURED:Qs4.NegException: Negative numbers not allowed, Try again

Enter number2

2

Enter number3

3

Enter number4

4

Average is 2.5

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

PROGRAM NO: 21

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

ALGORITHM

Step1: Create a class 'ThreadA' that inherits Thread class with member function as run()

Step2: Inside the run(), Print the multiplication table for 5.

Step2: Create a class named ThreadB that inherits Thread class with member function run()

Step3: Inside run(), Print the prime numbers up to limit of user's choice

Step4: Inside the main(), create an object for the classes and call start() using each object to begin the execution of thread.

Step5: Call sleep() to stop the execution of the current thread.

PROGRAM CODE

Threadprogram.java	<pre>package Qs_5; import java.util.*; public class Threadprogram { 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>
--------------------	--

	<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); } } } } } </pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

```
scjava.vscode-java-debug-0.34.0\scripts\launcher.bat "C:\Programs -Dfile.encoding=UTF-8 -cp C:\Users\USER\AppData\Roaming\CodeB-CYCLE4_4036c2\bin Qs5.Threadprogram "  
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:
```

```
20
```

```
Prime numbers between 1 and 20 are:
```

```
2  
3  
5  
7  
11  
13  
17  
19
```

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

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: Create a class named ThreadB that implements Runnable interface with function run()

Step 2: Inside run(), we read the limit for printing even numbers and print it using for loop.

Step 3: Create another class ThreadA that implements Runnable interface with function run().

Step 4: Inside run(), Initialise n1 as 0, n2 as 1 and n3 as 0.

Step 5: Check if n<0, if true, print "Enter a positive number" else goto step 7

Step 6: Repeat step 8 to 11 until n3>n

Step 7: Print n1

Step 8: n3=n1+n2

Step 9: n1=n2

Step 10: n2=n3

Step 11: Create object e of even and create an object 'a' of Thread with its parameterized constructor passing e as parameter

Step 12: Call start() using 'a'

Step 13: Do the same for class odd with Thread object 'b' and call start() using b

PROGRAM CODE

Threadtest.java	<pre>package Qs6; import java.util.*; public class Threadtest { public static void main(String args[]) throws InterruptedException { ThreadA at=new ThreadA();</pre>
-----------------	---

```

Thread a=new Thread(at);
ThreadB bt=new ThreadB();
Thread b=new Thread(bt);
a.start();
a.sleep(200);
b.start();
b.sleep(200);
}

}
class ThreadA implements Runnable
{
    public void run()
    {
        int i,f=1,s=1,t,n=10;
        System.out.println("\nFibonacci Series"+"\\n"+"-----
        -----");
        System.out.println(f);
        System.out.println(s);
        for(i=3;i<=n;i++)
        {
            t=f+s;
            System.out.println(t);
            f=s;
            s=t;
        }
        System.out.println("\\n");
    }
}
class ThreadB implements Runnable{
    public void run()
    {
        int i;
        System.out.println("Print Even numbers"+"\\n"+"-----
        -----");
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the limit");
        int n2=sc.nextInt();
        System.out.println("Print even numbers between 1 to
        "+n2+":");
        for(i=1;i<=n2;i++)
        {
            if(i%2==0)
            {
                System.out.println(i);
            }
        }
    }
}
}

```

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
s -Dfile.encoding=UTF-8 -cp C:\Users\USER\AppData\Roami  
B-CYCLE4_4036c2\bin Qs6.Threadtest "
```

```
Fibonacci Series
```

```
-----  
1  
1  
2  
3  
5  
8  
13  
21  
34  
55
```

```
Print Even numbers
```

```
-----  
Enter the limit  
20  
Print even numbers between 1 to 20:  
2  
4  
6  
8  
10  
12  
14  
16  
18  
20
```

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>█
```


PROGRAM NO: 23

AIM: Producer/Consumer using ITC

ALGORITHM

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.

PROGRAM CODE

ProducerConsumer.java	<pre>package Qs7; import java.util.ArrayList; import java.util.List; class Producer extends Thread { List<Integer> list; public Producer(List<Integer> list) { 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; } @Override public void run(){ try{ while(true){ synchronized(list){ if(list.size()==0) list.wait();</pre>
-----------------------	---

	<pre> 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(); } }</pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

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
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

PROGRAM NO: 24

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

ALGORITHM

Step 1: Create a class named Stackop and declare an array and top of stack ,a function named stackoperation().

Step 2: Inside stackoperation(), give choices to push,pop and display the stack

Step 3: If the choice is 1, then check whether the stack is full, else add an element into the stack.

Step 4: If the choice is 2, then check whether the stack is empty, else delete an element into the stack.

Step 5: If the choice is 3, then check whether the stack is empty, else print all the elements in the stack.

Step 6: If the choice is greater than 4, then print “Invalid choice”.

Step 7: Inside the main(), create an object of class ‘s’ and call the stackoperation() function.

PROGRAM CODE

Stack.java	<pre>package Qs8; import java.util.Scanner; class Stackop { int a[] = new int[20]; int top=-1,ch,item,i; Scanner sc = new Scanner(System.in); public void stackoperation() { System.out.println("Enter the size of the array : "); int n=sc.nextInt(); do { System.out.println("\nCHOICES : "+"\\n"+"-----"); System.out.println("\\n" + "1.PUSH"+"\\n" + "2.POP" + "\\n" + "3.DISPLAY" + "\\n" + "4.Exit"); System.out.println("\\n Enter your choice : ");</pre>
------------	---

```

ch=sc.nextInt();
switch(ch)
{
case 1: if(top >=n-1)
        {
            System.out.println("STACK OVERFLOW");
        }
        else
        {
            System.out.println("Enter the element :");
            item =sc.nextInt();
            top=top+1;
            a[top]=item;
        }
        break;
case 2 : if(top<0)
        {
            System.out.println("STACK UNDERFLOW");
        }
        else
        {
            a[top]='\0';
            top=top-1;
        }
        break;
case 3:
        if(top < 0)
        {
            System.out.println("STACK IS EMPTY");
        }
        else
        {
            System.out.println("\nSTACK"+"\\n"+"-----");
            for(i=top;i>=0;i--)
            {
                System.out.println(a[i]);
            }
        }
        break;

case 4 : return;
default : System.out.println("\\n Invalid choice");
}
}
while(ch!=4);
}

class Stack
{

```

	<pre> public static void main(String[] args) { Stackop s =new Stackop(); s.stackoperation(); } } </pre>
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```

Enter the size of the array :
5

```

```

CHOICES :
-----

```

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.Exit

```

Enter your choice :
1
Enter the element :
10

```

```

CHOICES :
-----

```

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.Exit

```

Enter your choice :
3

```

```

STACK
-----
10

```

CHOICES :

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.Exit

Enter your choice :

3

STACK

40
30
20
10

CHOICES :

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.Exit

Enter your choice :

2

CHOICES :

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.Exit

Enter your choice :

3

STACK

30
20
10

PROGRAM NO: 25

AIM: Using generic method perform Bubble sort

ALGORITHM

Step 1: Read number of numbers(N) to sort.

Step 2: Read the numbers

Step 3: Repeat step 5 for i=0 to N-1

Step 4: Repeat for j=i+1 to N

Step 5: Check if array[i] > array[j],

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

Step 7: Print the sorted array

PROGRAM CODE

Bubblesort.java	<pre>package Qs9; import java.util.*; public class Bubblesort { 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 < n; i++) array[i] = sc.nextInt(); for (i = 0; i < (n - 1); i++) { for (j = 0; j < n - i - 1; j++) {</pre>
-----------------	--

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

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4> cmd /C "c:\Users\USER\I
C:\Program Files\Java\jdk-14.0.2\bin\java.exe" -XX:+ShowCodeDetail:
ing\Code\User\workspaceStorage\02c047bfde05f988eca9db25f711b552\re
Enter the number of integers to sort:

5

Enter 5 numbers:

35

12

40

10

30

Sorted list of numbers:

10

12

30

35

40

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4> █

PROGRAM NO: 26

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

ALGORITHM

Step 1: Create an object of the class ArrayList.

Step 2: Adding elements to the object of ArrayList using method add() and display.

Step 3: Remove elements of object of ArrayList using method remove() and display .

Step 4: Sort elements of object of ArrayList using method sort() and display.

Step 5: Getting object of list which is present at the specified index using method get() and display.

Step 6: Check an element is present in the list using the method contains() and display True or False

Step 7: Clear List using the method clear().

PROGRAM CODE

ArrayListOp.java	<pre>package Qs10; import java.util.*; public class ArrayListOp { public static void main(String[] args) { ArrayList<String> obj =new ArrayList<String>(); System.out.println("Adding elements to the ArrayList:"+ "\n"+"-----"); obj.add("Rasika"); obj.add("Anila"); obj.add("Novrin"); obj.add("Ananya"); obj.add("Sindoori"); obj.add("Sree"); //Displaying elements System.out.println("ArrayList:"+ "\n"+"----- "); for(String str:obj) System.out.println(str); } }</pre>
------------------	---

```

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

//Sorting the ArrayList
Collections.sort(obj);
System.out.println("ArrayList after Sorting"+"\\n"+"-----
-----");
for(String str:obj)
    System.out.println(str);

//Remove "Sindoori" from ArrayList
obj.remove("Sindoori");

//Removes element from a given index
obj.remove(1);

//Displaying Elements
System.out.println("ArrayList after remove
operation"+"\\n"+"-----");
for(String str:obj)
    System.out.println(str);

//Displaying Final ArrayList
System.out.println("Final ArrayList"+"\\n"+"-----
-----");
for(String str:obj)
    System.out.println(str);

//Returns the object of list which is present at the specified
index
System.out.println("Object at the index 2:"+"\\n"+"-----
-----"+"\\n"+obj.get(2));

//Check whether the object is in the ArrayList
System.out.println("Check whether the given element is
present in the ArrayList"+"\\n"+"-----
");
System.out.println("Rasika is in the
ArrayList:"+obj.contains("Rasika"));
System.out.println("Riya is in the
ArrayList:"+obj.contains("Riya"));

//Add object into the ArrayList
obj.add(4,"Anu");

//Display the ArrayList after add
System.out.println("ArrayList after add operation"+"\\n"+"-----
-----");
for(String str:obj)

```

	<pre>System.out.println(str); //Removing all the elements in the ArrayList obj.clear(); System.out.println("ArrayList after clear method:"+obj); } }</pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
Adding elements to the ArrayList:
-----
ArrayList:
-----
Rasika
Anila
Novrin
Ananya
Sindoori
Sree
Size of the ArrayList
-----
6
ArrayList after Sorting
-----
Ananya
Anila
Novrin
Rasika
Sindoori
Sree
ArrayList after remove operation
-----
Ananya
Novrin
Rasika
Sree
Final ArrayList
-----
Ananya
Novrin
Rasika
Sree
Object at the index 2:
-----
Rasika
Check whether the given element is present in the ArrayList
-----
Rasika is in the ArrayList:true
Riya is in the ArrayList:false
ArrayList after add operation
-----
Ananya
Novrin
Rasika
Sree
Anu
ArrayList after clear method:[]

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

PROGRAM NO: 27

AIM: Program to remove all the elements from a linked list

ALGORITHM

Step 1: Create an object of the class LinkedList.

Step 2: Adding elements to the linked list using method add().

Step 3: Remove all the elements of linkedlist using method clear().

Step 4: Display linkedlist.

PROGRAM CODE

LinkedList_remove.java	<pre>package Qs11; import java.util.*; class LinkedList_remove { public static void main(String[] args) { LinkedList<String> list=new LinkedList<String>(); Scanner sc=new Scanner(System.in); System.out.print("how many Students do you want to add: "); int num=sc.nextInt(); for(int i=0;i<num;i++) { System.out.print("Add Student: "); String s=sc.next(); list.add(s); } System.out.println(); System.out.println("LinkedList After adding Students: "); Iterator<String> itr=list.iterator(); while(itr.hasNext()){ System.out.println(itr.next()); } list.clear(); System.out.println("Linked list After removing Students: " + list); } }</pre>
------------------------	--

	}
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL

```
Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4> cmd /C "c:\Users\USER\.vscode\
C:\Program Files\Java\jdk-14.0.2\bin\java.exe" -agentlib:jdwp=transport=dt
etailsInExceptionMessages -Dfile.encoding=UTF-8 -cp C:\Users\USER\AppData\
52\redhat.java\jdt_ws\LAB-CYCLE4_4036c2\bin Qs11.LinkedList_remove "
how many Students do you want to add: 5
Add Student: Rasika
Add Student: Nandana
Add Student: Novrin
Add Student: Ananya
Add Student: Anila

LinkedList After adding Students:
Rasika
Nandana
Novrin
Ananya
Anila
Linked list After removing Students: []

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

PROGRAM NO: 28

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

ALGORITHM

Step 1: Create an object of the class Stack.

Step 2: Adding elements to the stack using method add().

Step 3: Remove the element of stack at position 'pos' using method remove(pos).

Step 4: Display removed element and Stack after removal.

PROGRAM CODE

Stack_remove.java

```
package Qs12;
import java.util.*;
public class Stack_remove {
    public static void main(String[] args)
    {
        Stack<Integer> st=new Stack<Integer>();
        Scanner sc=new Scanner(System.in);
        System.out.print("how many Elements do you want to add:
");
        int num=sc.nextInt();
        for(int i=0;i<num;i++)
        {
            System.out.print("Add element: ");
            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);
    }
}
```

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\USER\Desktop\OOPS-LAB> cmd /C "c:\Users\USER\.vscode\e:
Files\Java\jdk-14.0.2\bin\java.exe" -agentlib:jdwp=transport=dt
eptionMessages -Dfile.encoding=UTF-8 -cp C:\Users\USER\AppData\
ava\jdt_ws\OOPS-LAB_18a71c5b\bin Stack_remove "
how many Elements do you want to add: 5
Add element: 10
Add element: 12
Add element: 25
Add element: 32
Add element: 50
Original Stack:[10, 12, 25, 32, 50]
Enter the index to be removed:
3
Removed Element is:32

Stack after remove:
[10, 12, 25, 50]

C:\Users\USER\Desktop\OOPS-LAB>
```

PROGRAM NO: 29

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

ALGORITHM

Step 1: Create an object of the class PriorityQueue.

Step 2: Adding elements to the PriorityQueue using method add().

Step 3: Display PriorityQueue.

PROGRAM CODE

Priority_queue.java	<pre>package Qs13; import java.util.*; public class Priority_queue { public static void main(String[] args) { PriorityQueue<String> queue=new PriorityQueue<String>(); 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<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");</pre>
---------------------	---

	<pre> Iterator<String> itr2=queue.iterator(); while(itr2.hasNext()){ System.out.println(itr2.next()); } } } </pre>
--	--

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```

C:\Program Files\Java\jdk-14.0.2\bin\java.exe" -age
etailsInExceptionMessages -Dfile.encoding=UTF-8 -cp
52\redhat.java\jdt_ws\LAB-CYCLE4_4036c2\bin Qs13.Pr
Enter Number Of elements
5
Enter the elements
10
20
30
40
50
head:10
head:10
Iterating the queue elements

10
20
30
40
50
After removing two elements

30
40
50

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>

```

PROGRAM NO: 30

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

ALGORITHM

Step 1: Create an object of the class Deque.

Step 2: Adding elements to the queue using method add().

Step 3: Removing elements of queue using method pop().

Step 4: Display Queue.

PROGRAM CODE

Dq.java	<pre>package Qs14; import java.util.*; public class Dq { public static void main(String[] args) { int choice; String element; Deque<String> deque= new LinkedList<String>(); 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();</pre>
---------	--

	<pre> 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"); 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); } } </pre>
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4> cmd
.\2\bin\java.exe" -agentlib:jdwp=transport=dt_so
\USER\AppData\Roaming\Code\User\workspaceStorag
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:
rasika
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:
anila
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:
novrin
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:
[rasika, anila, novrin]
```


ENTER THE CHOICE:

5

PRINT ELEMENTS:

[rasika, anila, novrin]

- 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:

[anila, novrin]

- 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:

4

ELEMENT DELETED FROM THE LAST 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:

[anila]

PROGRAM NO: 31

AIM: Program to demonstrate the creation of Set object using the LinkedHashSet class

ALGORITHM

Step 1: Create an object of the class Set.

Step 2: Adding elements to the HashSet using method add().

Step 3: Display LinkedHashSet.

PROGRAM CODE

Createset.java	<pre>package Qs15; import java.util.LinkedHashSet; import java.util.Scanner; import java.util.Set; import java.util.Iterator; public class Createset { public static void main(String[] args) { Set<Integer> s= new LinkedHashSet<Integer>(); int n; Scanner sc=new Scanner(System.in); System.out.println("Enter the number of elements to be added to set: "); n=sc.nextInt(); System.out.println("Enter set elements: "); while(n!=0) { int x=sc.nextInt(); s.add(x); n--; } System.out.println("Printing set:"+s); System.out.println("Enter the element to be deleted:"); int d=sc.nextInt();</pre>
----------------	--

	<pre> if(s.remove(d)) { System.out.println("Set after removal:"+s); } else { System.out.println("Element not found in the given set!"); } System.out.println("Iterating over set: "); Iterator<Integer> i = s.iterator(); while (i.hasNext()) { System.out.println(i.next()); } }</pre>
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

Enter the number of elements to be added to set:

5

Enter set elements:

10

20

30

40

50

Printing set:[10, 20, 30, 40, 50]

Enter the element to be deleted:

30

Set after removal:[10, 20, 40, 50]

Iterating over set:

10

20

40

50

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>

PROGRAM NO: 32

AIM: Write a Java program to compare two hash set

ALGORITHM

Step 1: Create two objects of the class LinkedHashset.

Step 2: Adding elements to the two objects of LinkedHashSet using method add().

Step 3: Compare two hash sets.

Step4: Performs union, intersection, difference operation of two sets.

PROGRAM CODE

Hashset_compare.java	<pre>package Qs16; import java.util.HashSet; import java.util.Scanner; import java.util.Set; public class Hashset_compare { public static void main(String[] args) { Set<Integer> s1=new HashSet<Integer>(); Set<Integer> s2=new HashSet<Integer>(); int n1,n2; Scanner sc=new Scanner(System.in); System.out.println("Enter the no of element in first set:"); n1=sc.nextInt(); System.out.println("Enter the elements of first set:"); for(int i=0;i<n1;i++) { int x=sc.nextInt(); s1.add(x); } System.out.println("Enter the no of element in Second set:"); n2=sc.nextInt(); System.out.println("Enter the elements of Second set:"); for(int i=0;i<n2;i++) {</pre>
----------------------	--

	<pre> int y=sc.nextInt(); s2.add(y); } System.out.println("HashSet1: "+s1); System.out.println("HashSet2: "+s2); System.out.println("\n****COMPARING TWO HASH SET****\n"); Set<Integer> union = new HashSet<Integer>(s1); union.addAll(s2); System.out.print("Union of the two Set"); System.out.println(union); Set<Integer> intersection = new HashSet<Integer>(s1); intersection.retainAll(s2); System.out.print("Intersection of the two Set"); System.out.println(intersection); Set<Integer> difference = new HashSet<Integer>(s1); difference.removeAll(s2); System.out.print("Difference of set1 from set2"); System.out.println(difference); } } </pre>
--	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
aming\Code\User\workspaceStorage\a2c047bfde05f988eca9db2
Enter the no of element in first set:
4
Enter the elements of first set:
10
12
30
45
Enter the no of element in Second set:
4
Enter the elements of Second set:
25
12
60
100
HashSet1: [10, 12, 45, 30]
HashSet2: [100, 25, 12, 60]

****COMPARING TWO HASH SET****

Union of the two Set[100, 25, 10, 12, 60, 45, 30]
Intersection of the two Set[12]
Difference of set1 from set2[10, 45, 30]

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

PROGRAM NO: 33

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

ALGORITHM

Step 2:Initialization of a Map.

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_interface.java	<pre>package Qs17; import java.util.*; public class Map_interface { public static void main(String[] args) { Map<Integer, String> map = new HashMap<>(); map.put(1,"Mumbai"); map.put(2, "Chennai"); map.put(3, "Kochi"); System.out.println("\nPrinting initial map: "+map); map.put((3),"Hyderabad"); map.put((2), "Delhi"); System.out.println("Updated map: "+map); map.remove(1); System.out.println("Map after removal: "+map); } }</pre>
--------------------	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

```
Printing initial map: {1=Mumbai, 2=Chennai, 3=Kochi}  
Updated map: {1=Mumbai, 2=Delhi, 3=Hyderabad}  
Map after removal: {2=Delhi, 3=Hyderabad}
```

```
C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>
```

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 put() method of treeMap

Step 4: Return the formed TreeMap

PROGRAM CODE

Hashmap_to_treemap.java	<pre>package Qs18; import java.util.HashMap; import java.util.Map; import java.util.TreeMap; public class Hashmap_to_treemap { public static void main(String[] args) { Map<String, String>map = new HashMap<String, String>(); map.put("1", "One"); map.put("2", "Two"); map.put("3", "Three"); map.put("4", "Four"); map.put("5", "Five"); map.put("6", "Six"); map.put("7", "Seven"); map.put("8", "Eight"); Map<String, String> converted = new TreeMap<String, String>(map); System.out.println("\nMap after conversion (HashMap to TreeMap): \n "+converted); } }</pre>
-------------------------	---

RESULT: The above program is successfully executed and obtained the output.

OUTPUT

Map after conversion (HashMap to TreeMap):

{1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight}

C:\Users\USER\Desktop\OOPS-LAB\LAB-CYCLE4>