

Experiment No.: 1

Date: 07-03-2023

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.

CO1:

Understand object-oriented concepts and design classes and objects to solve problems.

Procedure:

```
class Product

{

String pcode,pname;

double price;

void details()

{

System.out.println("PRODUCT DETAILS");

System.out.println("Pcode:" +pcode);

System.out.println("Pname:" +pname);

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

}

}

public class ProductDetails

{

public static void main(String[] args)

{

Product p1=new Product();

p1.pcode="D100";
```

```
p1.pname="Basket";

p1.price=100;

System.out.println("\n Product1:");

p1.details();

Product p2=new Product();

p2.pcode="D101";

p2.pname="Bottle";

p2.price=150;

System.out.println("\n Product2:");

p2.details();

Product p3=new Product();

p3.pcode="D102";

p3.pname="Jeans";

p3.price=500;

System.out.println("\n Product3:");

p3.details();

if(p1.price<p2.price && p1.price<p3.price)

{

System.out.println("\n Product1 has the lowest price.");

p1.details();

}

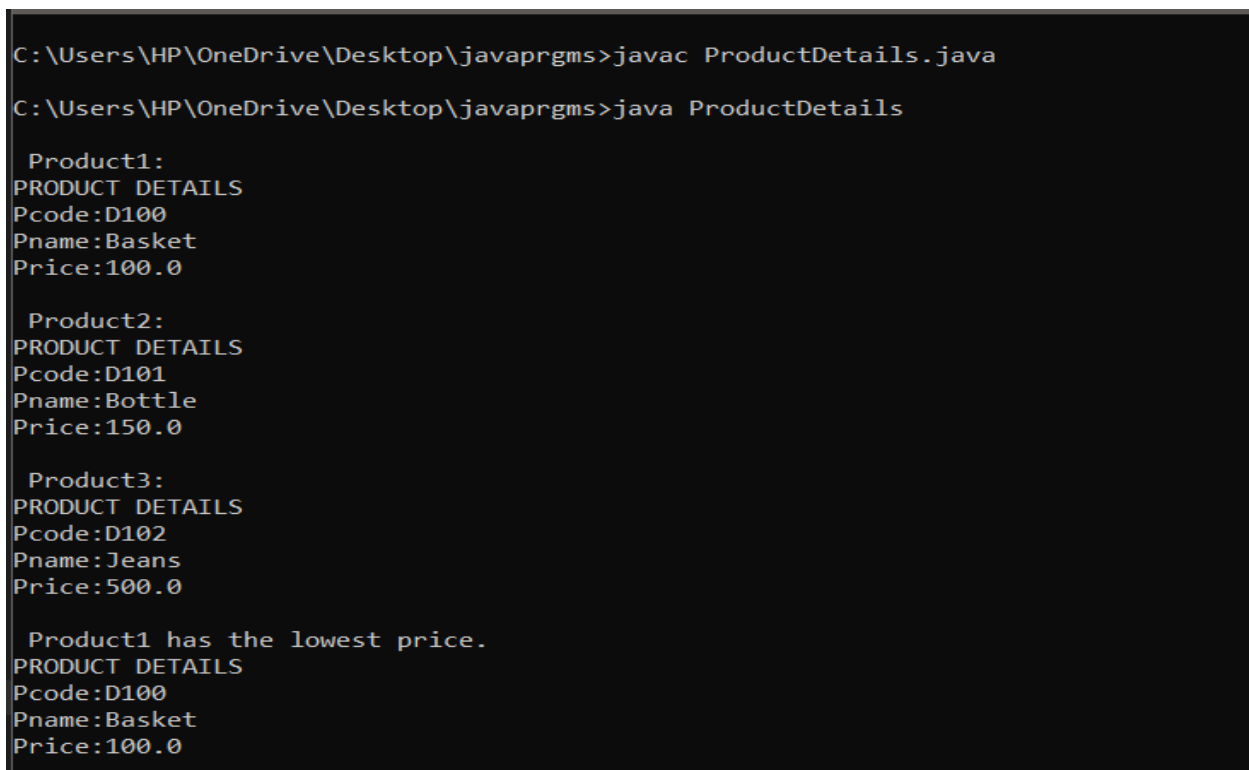
else if(p2.price<p3.price)

{

System.out.println("\n Product2 has the lowest price.");
```

```
p2.details();  
  
}  
  
else  
  
{  
  
System.out.println("\n Product3 has the lowest price.");  
  
p3.details();  
  
}  
  
}  
  
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac ProductDetails.java  
C:\Users\HP\OneDrive\Desktop\javaprgms>java ProductDetails  
  
Product1:  
PRODUCT DETAILS  
Pcode:D100  
Pname:Basket  
Price:100.0  
  
Product2:  
PRODUCT DETAILS  
Pcode:D101  
Pname:Bottle  
Price:150.0  
  
Product3:  
PRODUCT DETAILS  
Pcode:D102  
Pname:Jeans  
Price:500.0  
  
Product1 has the lowest price.  
PRODUCT DETAILS  
Pcode:D100  
Pname:Basket  
Price:100.0
```

Result:

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 2

Date: 09-03-2023

Aim:

Read 2 matrices from the console and perform matrix addition.

CO1:

Understand object-oriented concepts and design classes and objects to solve problems.

Procedure:

```
import java.util.Scanner;

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

        System.out.println("Enter the row and column of first matrix:");
        Integer r1=obj.nextInt();
        Integer c1=obj.nextInt();

        System.out.println("Enter the row and column of second matrix:");
        Integer r2=obj.nextInt();
        Integer c2=obj.nextInt();

        int a[][]=new int[r1][c1];
        int c[][]=new int[r1][c1];
        int b[][]=new int[r2][c2];
        int i,j;
        System.out.println("Enter the elements of first array:");

        for(i=0;i<r1;i++)
```

```
{  
for(j=0;j<c1;j++)  
{  
a[i][j]=obj.nextInt();  
}  
}
```

```
System.out.println("Enter the elements of second array:");  
for(i=0;i<r2;i++)  
{  
for(j=0;j<c2;j++)  
{  
b[i][j]=obj.nextInt();  
}  
}
```

```
System.out.println("The first matrix is:");  
for(i=0;i<r1;i++)  
{  
for(j=0;j<c1;j++)  
{  
System.out.print(a[i][j]+" ");  
}  
System.out.println();  
}
```

```
System.out.println("The second matrix is:");  
for(i=0;i<r2;i++)  
{  
for(j=0;j<c2;j++)  
{  
System.out.print(b[i][j]+" ");  
}
```

```
}  
System.out.println();  
}  
  
if(r1==r2 && c1==c2)  
{  
    for(i=0;i<r1;i++)  
    {  
        for(j=0;j<r1;j++)  
        {  
            c[i][j]=a[i][j]+b[i][j];  
        }  
    }  
  
    System.out.println("The sum of first and second matrix is:");  
    for(i=0;i<r1;i++)  
    {  
        for(j=0;j<c1;j++)  
        {  
            System.out.print(c[i][j]+" ");  
        }  
        System.out.println();  
    }  
  
    else  
    {  
        System.out.println("They are not compatible.");  
    }  
  
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Matadd.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Matadd
Enter the row and column of first matrix:
2
2
Enter the row and column of second matrix:
2
2
Enter the elements of first array:
5
1
2
6
Enter the elements of second array:
4
2
3
1
The first matrix is:
5 1
2 6
The second matrix is:
4 2
3 1
The sum of first and second matrix is:
9 3
5 7
```

Result:

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 3

Date: 09-03-2023

Aim:

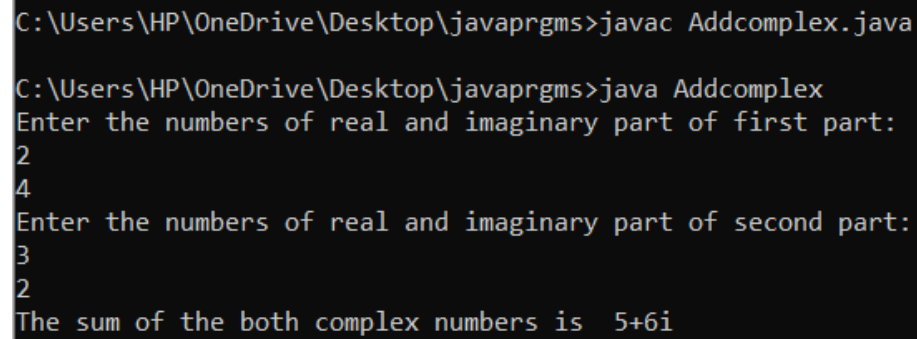
Add complex numbers.

CO1:

Understand object-oriented concepts and design classes and objects to solve problems.

Procedure:

```
import java.util.Scanner;
public class Addcomplex
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the numbers of real and imaginary part of first part:");
        Integer fr=obj.nextInt();
        Integer fi=obj.nextInt();
        System.out.println("Enter the numbers of real and imaginary part of second part:");
        Integer sr=obj.nextInt();
        Integer si=obj.nextInt();
        Integer resultr=fr+sr;
        Integer resulti=fi+si;
        System.out.println("The sum of the both complex numbers is "+resultr+"+" +resulti+"i");
    }
}
```


Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Addcomplex.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Addcomplex  
Enter the numbers of real and imaginary part of first part:  
2  
4  
Enter the numbers of real and imaginary part of second part:  
3  
2  
The sum of the both complex numbers is 5+6i
```

Result:

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 4

Date: 14-03-2023

Aim:

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

CO1:

Understand object-oriented concepts and design classes and objects to solve problems.

Procedure:

```
import java.util.Scanner;

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

        System.out.println("Enter the row and column of first matrix:");

        Integer r1=obj.nextInt();
        Integer c1=obj.nextInt();

        int a[][]=new int[r1][c1];
        int c[][]=new int[r1][c1];

        int i,j;

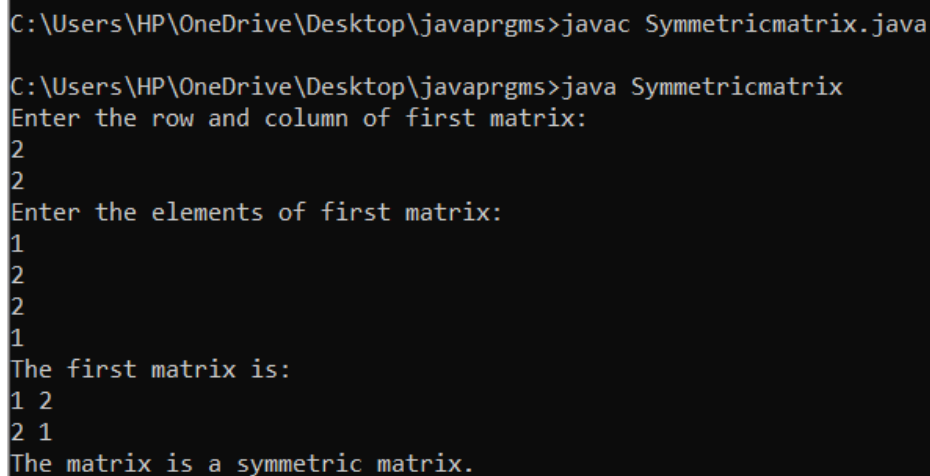
        System.out.println("Enter the elements of first matrix:");

        for(i=0;i<r1;i++)
        {
            for(j=0;j<c1;j++)
```

```
{  
  
a[i][j]=obj.nextInt();  
  
}  
  
}  
  
System.out.println("The first matrix is:");  
  
for(i=0;i<r1;i++)  
{  
for(j=0;j<c1;j++)  
{  
System.out.print(a[i][j]+" ");  
}  
System.out.println();  
}  
  
int flag=1;  
  
for(i=0;i<r1;i++)  
{  
for(j=0;j<c1;j++)  
{  
c[i][j]=a[j][i];  
  
if(c[i][j]!=a[i][j])  
{  
flag=0;  
}  
}  
}
```

```
}  
  
if(flag==1)  
  
{  
  
System.out.println("The matrix is a symmetric matrix.");  
  
}  
  
else  
  
{  
  
System.out.println("The matrix is a not a symmetric matrix.");  
  
}  
  
}  
  
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Symmetricmatrix.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Symmetricmatrix  
Enter the row and column of first matrix:  
2  
2  
Enter the elements of first matrix:  
1  
2  
2  
1  
The first matrix is:  
1 2  
2 1  
The matrix is a symmetric matrix.
```

Result:

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 5

Date: 16-03-2023

Aim:

Program to Sort strings

CO2:

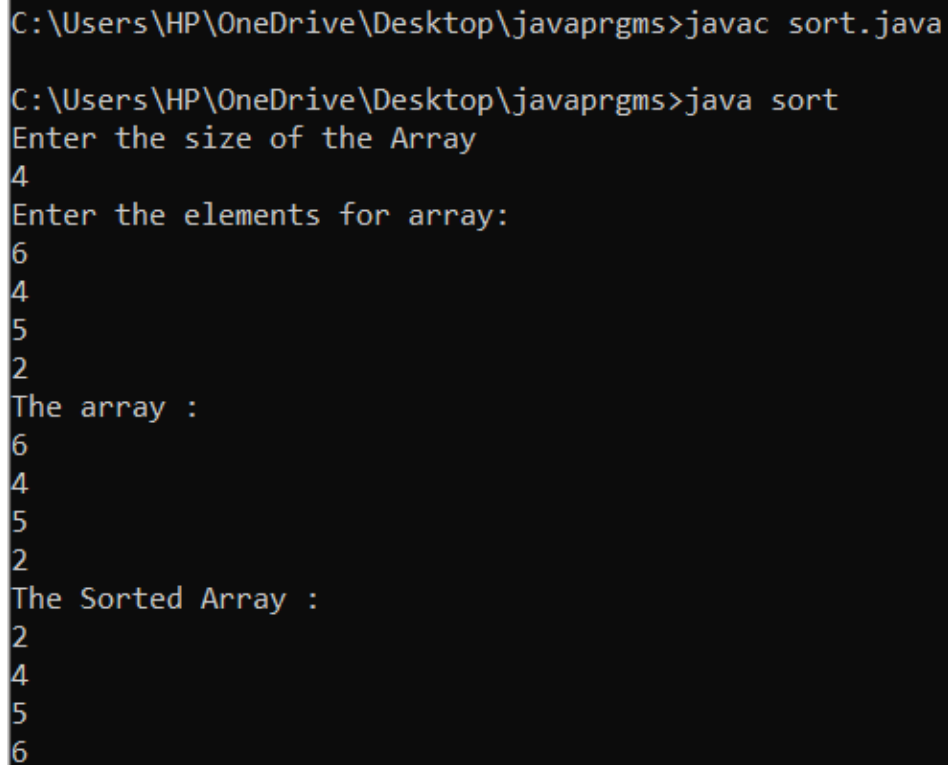
Familiarization and understanding of arrays and strings

Procedure:

```
import java.util.Scanner;
public class Compare
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size of the Array ");
        int l =sc.nextInt();
        String str[] = new String[l];
        int i;
        System.out.println("Enter the elements for array: ");
        for(i=0;i<str.length;i++)
        {
            str[i]=sc.next();
        }
        System.out.println("The array : ");
        for(i=0;i<str.length;i++)
        {
            System.out.println(str[i]);
        }
        String temp;
        int j;
        for(i=0;i<str.length;i++)
        {
            for(j=i+1;j<str.length;j++)
            {
                if(str[i].compareTo(str[j])>0)
                {
                    temp=str[i];
                    str[i]=str[j];
                    str[j]=temp;
                }
            }
        }
    }
}
```

```
System.out.println("The Sorted Array : ");
for(i=0;i<str.length;i++)
{
System.out.println(str[i]);
}
}
```

Output Screenshot:



The screenshot shows a command prompt window with the following text:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac sort.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java sort
Enter the size of the Array
4
Enter the elements for array:
6
4
5
2
The array :
6
4
5
2
The Sorted Array :
2
4
5
6
```

Result:

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 6

Date: 16-03-2023

Aim:

Search an element in an array.

CO2:

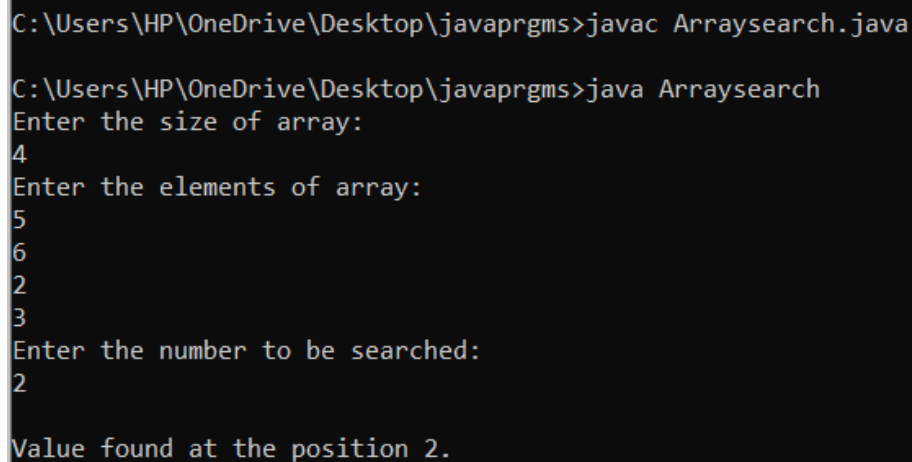
Familiarization and understanding of arrays and strings

Procedure:

```
import java.util.Scanner;
public class Arraysearch
{
public static void main(String[] args)
{
Scanner obj=new Scanner(System.in);
System.out.println("Enter the size of array:");
Integer n=obj.nextInt();
int arr[]=new int[n];
int i,flag=0;
System.out.println("Enter the elements of array:");
for(i=0;i<arr.length;i++)
{
arr[i]=obj.nextInt();
}
System.out.println("Enter the number to be searched:");
Integer item=obj.nextInt();
System.out.println("");
for(i=0;i<arr.length;i++)
{
if(item==arr[i])
{
```

```
System.out.println("Value found at the position "+i+ ".");  
flag=1;  
  
}  
  
}  
if(flag==0)  
{  
System.out.println("The number is not found.");  
}  
  
}  
  
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Arraysearch.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Arraysearch  
Enter the size of array:  
4  
Enter the elements of array:  
5  
6  
2  
3  
Enter the number to be searched:  
2  
  
Value found at the position 2.
```

Result:

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 7

Date: 16-03-2023

Aim:

Perform string manipulations

CO2:

Familiarization and understanding of arrays and strings

Procedure:

```
import java.util.Scanner;
public class Stringmanipulation
{
public static void main(String[] args)
{
Scanner obj=new Scanner(System.in);
System.out.println("Enter the first string:");
String a=obj.nextLine();
System.out.println("The length of the first string is:"+a.length());

System.out.println("Enter the second string:");
String b=obj.nextLine();
System.out.println("The length of the second string is:"+b.length());

System.out.println("The concated string is "+a.concat(b));

System.out.println("The uppercase string is "+a.toUpperCase());
System.out.println("The uppercase string is "+b.toUpperCase());

System.out.println("The lowercase string is "+a.toLowerCase());
```

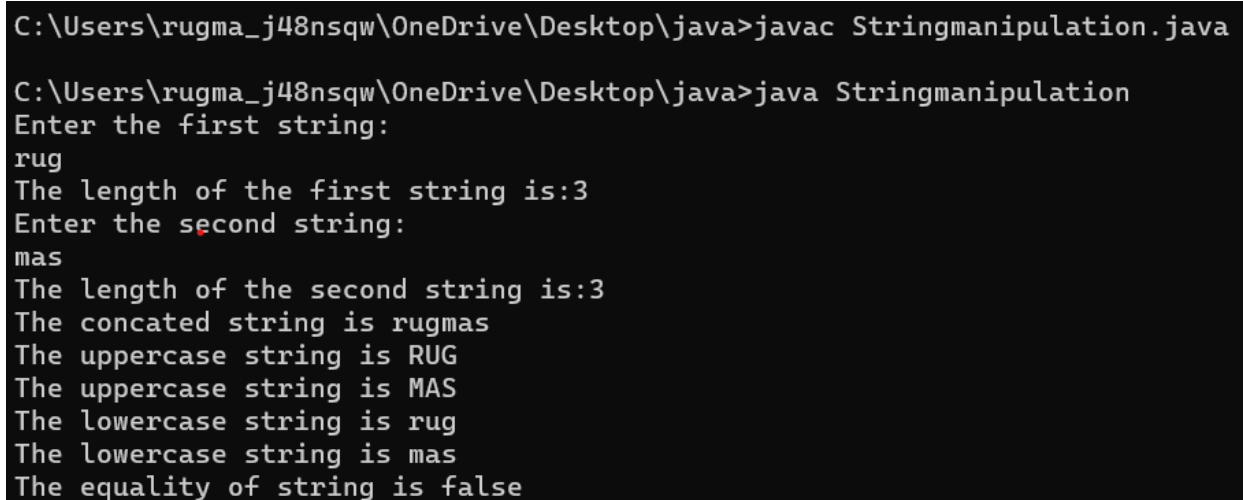
```
System.out.println("The lowercase string is "+b.toLowerCase());
```

```
System.out.println("The equality of string is "+a.equals(b));
```

```
}
```

```
}
```

Output Screenshot:



```
C:\Users\rugma_j48nsqw\OneDrive\Desktop\java>javac Stringmanipulation.java

C:\Users\rugma_j48nsqw\OneDrive\Desktop\java>java Stringmanipulation
Enter the first string:
rug
The length of the first string is:3
Enter the second string:
mas
The length of the second string is:3
The concated string is rugmas
The uppercase string is RUG
The uppercase string is MAS
The lowercase string is rug
The lowercase string is mas
The equality of string is false
```

Result:

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 8

Date: 16-03-2023

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.

CO2:

Familiarization and understanding of arrays and strings

Procedure:

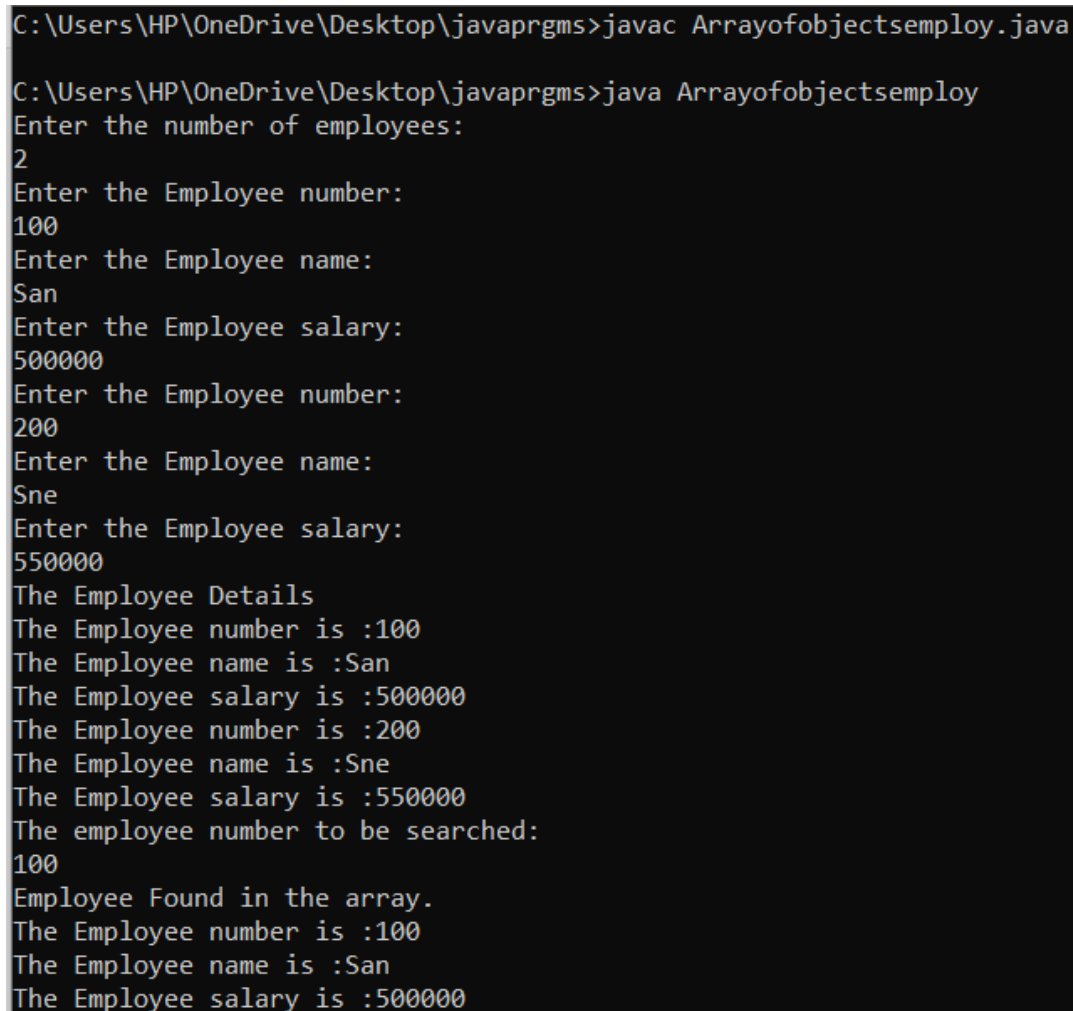
```
import java.util.Scanner;

public class Arrayofobjectsemploy
{
    int eNo,eSalary;
    String eName;
    public void getDetails()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the Employee number:");
        eNo=sc.nextInt();
        System.out.println("Enter the Employee name:");
        eName=sc.next();
        System.out.println("Enter the Employee salary:");
        eSalary=sc.nextInt();
    }
    public void putDetails()
    {
        System.out.println("The Employee number is :"+eNo);
        System.out.println("The Employee name is :"+eName);
        System.out.println("The Employee salary is :"+eSalary);
    }
}
```

```
}  
  
public static void main(String[] args)  
{  
    Scanner sc=new Scanner(System.in);  
    System.out.println("Enter the number of employees:");  
    int n1=sc.nextInt();  
    Arrayofobjectsemploy a[]=new Arrayofobjectsemploy[n1];  
    for(int i=0;i<n1;i++)  
    {  
        a[i]=new Arrayofobjectsemploy();  
        a[i].getDetails();  
    }  
    System.out.println("The Employee Details");  
    for(int i=0;i<n1;i++)  
    {  
        a[i].putDetails();  
    }  
    System.out.println("The employee number to be searched:");  
    int n2=sc.nextInt();  
    int flag=0;  
    for(int i=0;i<n1;i++)  
    {  
        if(a[i].eNo==n2);  
        {  
            System.out.println("Employee Found in the array.");  
            flag=1;  
            a[i].putDetails();  
            break;  
        }  
    }
```

```
}  
if(flag!=1)  
{  
System.out.println("Employee not found.");  
}  
}  
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Arrayofobjectsemploy.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Arrayofobjectsemploy  
Enter the number of employees:  
2  
Enter the Employee number:  
100  
Enter the Employee name:  
San  
Enter the Employee salary:  
500000  
Enter the Employee number:  
200  
Enter the Employee name:  
Sne  
Enter the Employee salary:  
550000  
The Employee Details  
The Employee number is :100  
The Employee name is :San  
The Employee salary is :500000  
The Employee number is :200  
The Employee name is :Sne  
The Employee salary is :550000  
The employee number to be searched:  
100  
Employee Found in the array.  
The Employee number is :100  
The Employee name is :San  
The Employee salary is :500000
```

Result:

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 9

Date: 21-03-2023

Aim:

Area of different shapes using overloaded functions

CO3:

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure:

```
import java.util.Scanner;
public class Methodover
{
    int r;
    double pie=3.14;
    public void area(int r)
    {
        double areaofcir=pie*r*r;
        System.out.print("\n area of circle is:"+areaofcir);
    }
    public void area(double b, double h)
    {
        double areaoftri=1.5*b*h;
        System.out.print("\n area of triangle is:"+areaoftri);
    }
    public void area(int l, int w)
    {
        double areaofrec=l*w;
        System.out.print(" \n area of rectangle is:"+areaofrec);
    }
}
```

```
public void area(float a)
{
    double areaofsqr=a*a;
    System.out.print("\n area of square is:"+areaofsqr);
}

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

    System.out.print("enter the radius:");
    int r=sc.nextInt();
    System.out.print("enter the breadth:");
    int b=sc.nextInt();
    System.out.print("enter the height:");
    int h=sc.nextInt();
    System.out.print("enter the length:");
    int l=sc.nextInt();
    System.out.print("enter the width:");
    int w=sc.nextInt();
    System.out.print("enter the a:");
    int a=sc.nextInt();
    obj.area(r);
    obj.area(b,h);
    obj.area(l,w);
    obj.area(a);
}
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Methodover.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Methodover
enter the radius:2
enter the breadth:4
enter the height:5
enter the length:6
enter the width:3
enter the a:5

area of circle is:12.56
area of rectangle is:20.0
area of rectangle is:18.0
area of circle is:78.5
C:\Users\HP\OneDrive\Desktop\javaprgms>_
```

Result:

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 10

Date: 21-03-2023

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.

CO3:

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure:

```
import java.util.Scanner;

class Employee{
    int empid;
    String Ename;
    int Esalary;
    String Eaddress;
    Employee(int id, String name, int salary , String Add){
        empid=id;
        Ename=name;
        Esalary=salary;
        Eaddress=Add;
    }
    class Teacher extends Employee{
        String department;
        String sub_taught;
        Teacher(int id, String name, int salary , String Add,String dept, String sub){
            super(id,name,salary,Add);
            department=dept;
            sub_taught=sub;
        }
    }
}
```

```
}  
void show(){  
System.out.println("The Employee id:"+empid);  
System.out.println("The Employee name:"+Ename);  
System.out.println("The Employee salary:"+Esalary);  
System.out.println("The Employee address:"+Eaddress);  
System.out.println("The Employee department:"+department);  
System.out.println("The Subject taught:"+sub_taught);  
}}  
public class EmpTeacher{  
public static void main(String args[]){  
Scanner sc = new Scanner(System.in);  
int i;  
System.out.println("Enter the no. of Employees:");  
int l=sc.nextInt();  
Teacher t[]=new Teacher[l];  
for(i=0;i<l;i++){  
System.out.println("Enter the employee id:");  
int empid=sc.nextInt();  
System.out.println("Enter the employee name:");  
String empname= sc.next();  
System.out.println("Enter the employee salary:");  
int empsal=sc.nextInt();  
System.out.println("Enter the employee address:");  
String empaddress=sc.next();  
System.out.println("Enter the employee department:");  
String empdept=sc.next();  
System.out.println("Enter the Subject taught:");  
String empsub=sc.next();  
t[i]= new Teacher(empid,empname,empsal,empaddress,empdept,empsub);  
}  
for(i=0;i<l;i++){
```

```
t[i].show();  
}  
}  
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac EmpTeacher.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java EmpTeacher  
Enter the no. of Employees:  
2  
Enter the employee id:  
100  
Enter the employee name:  
San  
Enter the employee salary:  
500000  
Enter the employee address:  
TREE  
Enter the employee department:  
CS  
Enter the Subject taught:  
Maths  
Enter the employee id:  
200  
Enter the employee name:  
Sis  
Enter the employee salary:  
550000  
Enter the employee address:  
MIST  
Enter the employee department:  
CA  
Enter the Subject taught:  
Accountant  
The Employee id:100  
The Employee name:San  
The Employee salary:500000  
The Employee address:TREE  
The Employee department:CS  
The Subject taught:Maths  
The Employee id:200  
The Employee name:Sis  
The Employee salary:550000  
The Employee address:MIST  
The Employee department:CA  
The Subject taught:Accountant
```

Result:

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 11

Date: 23-03-2023

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.

CO3:

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure:

```
import java.util.*;

class Person
{
    String Pname;
    String Pgender;
    String Paddress;
    int Page;
    Person()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Name:");
        Pname=sc.next();
        System.out.println("Enter the Gender:");
        Pgender=sc.next();
        System.out.println("Enter the Address:");
        Paddress=sc.next();
        System.out.println("Enter the Age:");
        Page=sc.nextInt();
    }
}
```

```
}  
}  
class Employee extends Person  
{  
    int empid;  
    String company_name;  
    String qualification;  
    int salary;  
    Employee()  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the Employee id:");  
        empid=sc.nextInt();  
        System.out.println("Enter the Company name:");  
        company_name=sc.next();  
        System.out.println("Enter the Qualification:");  
        qualification=sc.next();  
        System.out.println("Enter the Salary");  
        salary=sc.nextInt();  
    }  
}  
class Teacher extends Employee  
{  
    String subject;  
    String Department;  
    int Teacherid;  
    Teacher()  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the Subject:");  
        subject=sc.next();  
        System.out.println("Enter the Department:");
```

```
Department=sc.next();
System.out.println("Enter the Teacher id:");
Teacherid=sc.nextInt();
System.out.println("-----");
}
public void show()
{
System.out.println("#####");
System.out.println("The Name: "+Pname);
System.out.println("The Gender:"+Pgender);
System.out.println("The Address:"+Paddress);
System.out.println("The Age:"+Page);
System.out.println("The Employee id: "+empid);
System.out.println("The Company name:"+company_name);
System.out.println("The Qualification:"+qualification);
System.out.println("The Salary:"+salary);
System.out.println("The Teacher id :"+Teacherid);
System.out.println("The Subject:"+subject);
System.out.println("The Department:"+Department);
System.out.println(" ");
}
}
public class Person2
{
public static void main(String[] args)
{
Scanner sc = new Scanner(System.in);
int i;
System.out.println("Enter the no. of Employees:");
int l =sc.nextInt();
Teacher t[]=new Teacher[l];
System.out.println("Enter the Details of "+l+" Employees are: ");
```

```
for(i=0;i<l;i++)  
{  
t[i]= new Teacher();  
}  
for(i=0;i<l;i++)  
{  
t[i].show();  
}  
}  
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Person2.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Person2
Enter the no. of Employees:
2
Enter the Details of 2 Employees are:
Enter the Name:
San
Enter the Gender:
Female
Enter the Address:
MIST
Enter the Age:
23
Enter the Employee id:
100
Enter the Company name:
IBM
Enter the Qualification:
MCA
Enter the Salary
200000
Enter the Subject:
CS
Enter the Department:
CA
Enter the Teacher id:
10
```

```
-----
Enter the Name:
Sen
Enter the Gender:
Male
Enter the Address:
MEND
Enter the Age:
24
Enter the Employee id:
200
Enter the Company name:
SIFY
Enter the Qualification:
MCA
Enter the Salary
250000
Enter the Subject:
CS
Enter the Department:
CA
Enter the Teacher id:
20
-----
```



```
#####  
The Name: San  
The Gender:Female  
The Address:MIST  
The Age:23  
The Employee id: 100  
The Company name:IBM  
The Qualification:MCA  
The Salary:200000  
The Teacher id :10  
The Subject:CS  
The Department:CA
```

```
#####  
The Name: Sen  
The Gender:Male  
The Address:MEND  
The Age:24  
The Employee id: 200  
The Company name:SIFY  
The Qualification:MCA  
The Salary:250000  
The Teacher id :20  
The Subject:CS  
The Department:CA
```

Result:

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 12

Date: 23-03-2023

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.

CO3:

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure:

```
import java.util.Scanner;

class Publisher
{
    String Pubname;
    Publisher()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the publisher: ");
        Pubname=sc.next();
    }
}

class books extends Publisher
{
    String booktype ;
    books()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the type of the book Literature or Fiction: ");
        booktype= sc.next();
    }
}

class Literature extends books
```

```
{
String title , author ;
int price;
Literature()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the title of the literature book: ");
title= sc.next();
System.out.println("Enter the author of the literature book: ");
author= sc.next();
System.out.println("Enter the price of the literature book: ");
price= sc.nextInt();
}
void display()
{
System.out.println("*****");
System.out.println("The Details of the Literature Book!!");
System.out.println("Publisher Name: "+Pubname);
System.out.println("Type of the Book: "+booktype );
System.out.println("Title: "+title);
System.out.println("Author: "+author);
System.out.println("Price: "+price);
System.out.println("*****");
System.out.println(" ");
}
}
class Fiction extends books
{
String title1 , author1 ;
int price1;
Fiction()
{
```

```
Scanner sc = new Scanner(System.in);
System.out.println("Enter the title of the Fiction book: ");
title1= sc.next();
System.out.println("Enter the author of the Fiction book: ");
author1= sc.next();
System.out.println("Enter the price of the Fiction book: ");
price1= sc.nextInt();
}
void display1()
{
System.out.println("*****");
System.out.println("The Details of the Fiction Book!!");
System.out.println("Publisher Name: "+Pubname);
System.out.println("Type of the Book: "+booktype );
System.out.println("Title: "+title1);
System.out.println("Author: "+author1);
System.out.println("Price: "+price1);
System.out.println("*****");
System.out.println(" ");
}
}
public class Book
{
public static void main(String[] args)
{
Scanner sc=new Scanner(System.in);
int i;
System.out.print("Enter number of entries for Literature:");
int ln=sc.nextInt();
Literature l[]=new Literature[ln];
for(i=0;i<ln;i++)
{
```

```
l[i]=new Literature();
}
for(i=0;i<ln;i++)
{
l[i].display();
}

System.out.print(" ");
System.out.print("Enter number of entries for Fiction:");
int fn=sc.nextInt();
Fiction f[]=new Fiction[fn];
for(i=0;i<fn;i++)
{
f[i]=new Fiction();
}
for(i=0;i<fn;i++)
{
f[i].display1();
}
}
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Book.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Book
Enter number of entries for Literature:1
Enter the publisher:
MS
Enter the type of the book Literature or Fiction:
Literature
Enter the title of the literature book:
Poem
Enter the author of the literature book:
LL
Enter the price of the literature book:
555
*****
The Details of the Literature Book!!
Publisher Name: MS
Type of the Book: Literature
Title: Poem
Author: LL
Price: 555
*****

Enter number of entries for Fiction:1
Enter the publisher:
SM
Enter the type of the book Literature or Fiction:
Fiction
Enter the title of the Fiction book:
MagicScience
Enter the author of the Fiction book:
Raz
Enter the price of the Fiction book:
899
*****
The Details of the Fiction Book!!
Publisher Name: SM
Type of the Book: Fiction
Title: MagicScience
Author: Raz
Price: 899
*****
```

Result:

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 13

Date: 28-03-2023

Aim:

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

C03:

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure:

```
import java.util.Scanner;

interface Student
{
    public void student_get();
    public void student_display();
}

interface Sports
{
    public void sport_get();
    public void sport_display();
}

class Result implements Student,Sports
{
    String name , sports ;
    double rollno , mark1, mark2, mark3,total , pos;
    double percentage;
    public void student_get(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name of the student: ");
        name = sc.next();
        System.out.println("Enter the Roll No of the student: ");
        rollno = sc.nextDouble();
    }
}
```

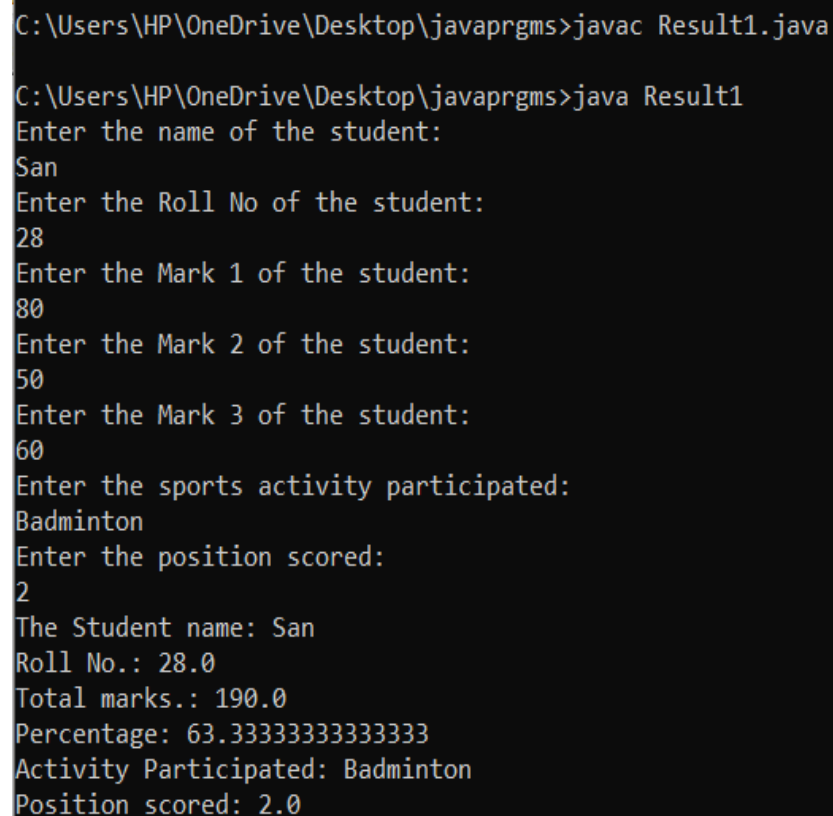
```
System.out.println("Enter the Mark 1 of the student: ");
mark1 = sc.nextDouble();
System.out.println("Enter the Mark 2 of the student: ");
mark2 = sc.nextDouble();
System.out.println("Enter the Mark 3 of the student: ");
mark3 = sc.nextDouble();
total = mark1+mark2+mark3;
percentage = total/300*100;
}
public void student_display()
{
System.out.println("The Student name: "+name);
System.out.println("Roll No.: "+rollno);
System.out.println("Total marks.: "+total);
System.out.println("Percentage: "+percentage);
}
public void sport_get()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the sports activity participated: ");
sports = sc.next();
System.out.println("Enter the position scored:");
pos = sc.nextInt();
}
public void sport_display()
{
System.out.println("Activity Participated: "+sports);
System.out.println("Position scored: "+pos);
}

}
```



```
public class Result1
{
    public static void main(String[] args)
    {
        Result obj = new Result();
        obj.student_get();
        obj.sport_get();
        obj.student_display();
        obj.sport_display();
    }
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Result1.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Result1
Enter the name of the student:
San
Enter the Roll No of the student:
28
Enter the Mark 1 of the student:
80
Enter the Mark 2 of the student:
50
Enter the Mark 3 of the student:
60
Enter the sports activity participated:
Badminton
Enter the position scored:
2
The Student name: San
Roll No.: 28.0
Total marks.: 190.0
Percentage: 63.33333333333333
Activity Participated: Badminton
Position scored: 2.0
```

Result:

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 14

Date: 28-03-2023

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.

CO3:

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure:

```
import java.util.Scanner;

interface prototype
{
    public void get_data();
    public void display();
    public void area();
    public void perimeter();
}

class Circle implements prototype
{
    int r ;
    double area,perimeter;
    public void get_data()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the radius of the circle: ");
        r=sc.nextInt();
    }
    public void area()
    {
        area=3.14*r*r;
```

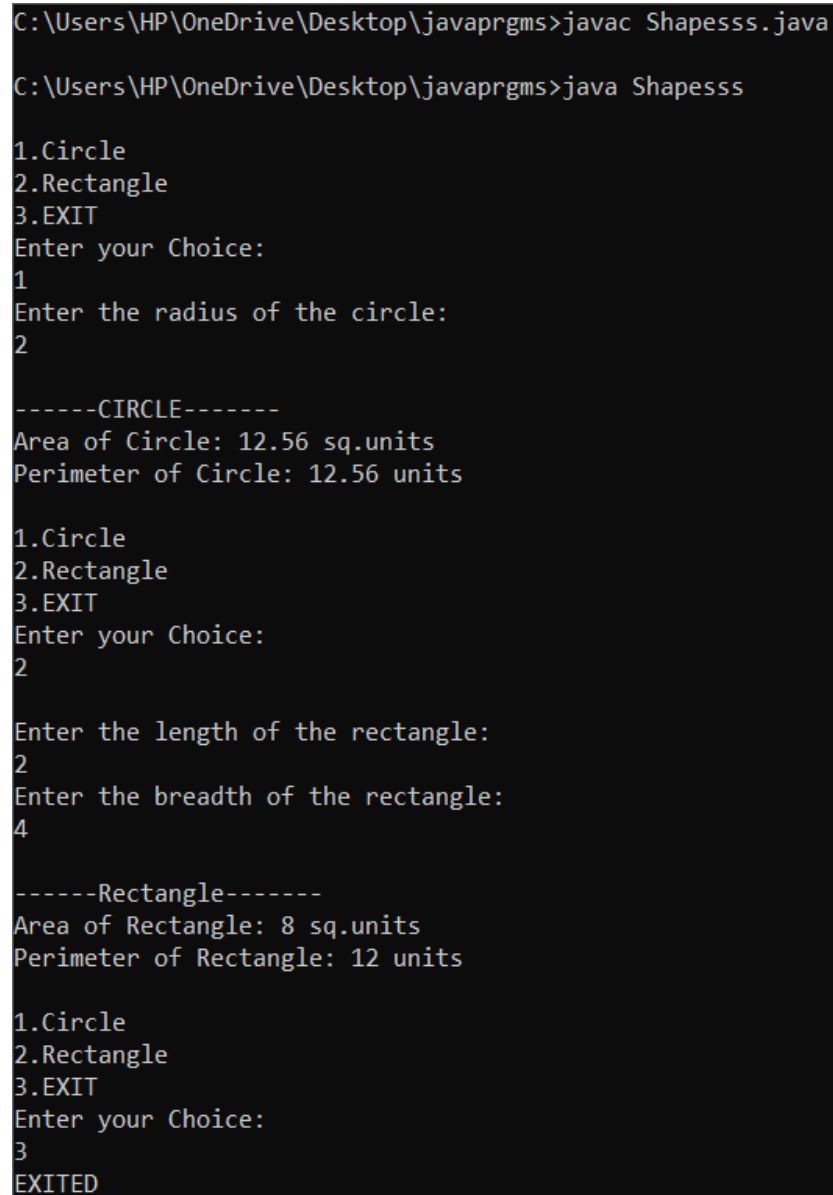
```
}  
public void perimeter()  
{  
    perimeter=2*3.14*r;  
}  
public void display()  
{  
    System.out.println(" ");  
    System.out.println("-----CIRCLE-----");  
    System.out.println("Area of Circle: "+area+" sq.units");  
    System.out.println("Perimeter of Circle: "+perimeter+" units");  
}  
}  
class Rectangle implements prototype  
{  
    int l,b,area1,perimeter1;  
    public void get_data(){  
        System.out.println(" ");  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the length of the rectangle: ");  
        l=sc.nextInt();  
        System.out.println("Enter the breadth of the rectangle: ");  
        b=sc.nextInt();  
    }  
    public void area()  
    {  
        area1=l*b;  
    }  
    public void perimeter()  
    {  
        perimeter1=2*(l+b);  
    }  
}
```

```
public void display()
{
    System.out.println(" ");
    System.out.println("-----Rectangle-----");
    System.out.println("Area of Rectangle: "+area1+" sq.units");
    System.out.println("Perimeter of Rectangle: "+perimeter1+" units");
}
}

public class Shapes
{
    public static void main(String[] args)
    {
        int ch,u=0;
        Scanner sc = new Scanner(System.in);
        Circle obj =new Circle();
        Rectangle obj1=new Rectangle();
        do{
            System.out.println("\n1.Circle\n2.Rectangle\n3.EXIT ");
            System.out.println("Enter your Choice:");
            ch=sc.nextInt();
            switch(ch){
                case 1: obj.get_data();
                    obj.area();
                    obj.perimeter();
                    obj.display();
                    break;
                case 2:obj1.get_data();
                    obj1.area();
                    obj1.perimeter();
                    obj1.display();
                    break;
                case 3:System.out.println("EXITED");
```

```
System.exit(0);  
}  
}  
while(ch!=3);  
}  
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Shapesss.java  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Shapesss  
  
1.Circle  
2.Rectangle  
3.EXIT  
Enter your Choice:  
1  
Enter the radius of the circle:  
2  
  
-----CIRCLE-----  
Area of Circle: 12.56 sq.units  
Perimeter of Circle: 12.56 units  
  
1.Circle  
2.Rectangle  
3.EXIT  
Enter your Choice:  
2  
  
Enter the length of the rectangle:  
2  
Enter the breadth of the rectangle:  
4  
  
-----Rectangle-----  
Area of Rectangle: 8 sq.units  
Perimeter of Rectangle: 12 units  
  
1.Circle  
2.Rectangle  
3.EXIT  
Enter your Choice:  
3  
EXITED
```

Result:

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 15

Date: 28-03-2023

Aim:

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

Order No. ,Date :

ProductId	Name	Quantity	unitprice	Total
-----------	------	----------	-----------	-------

101	A	2	25	50
-----	---	---	----	----

102	B	1	100	100
-----	---	---	-----	-----

Net.Amount 150

CO3:

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

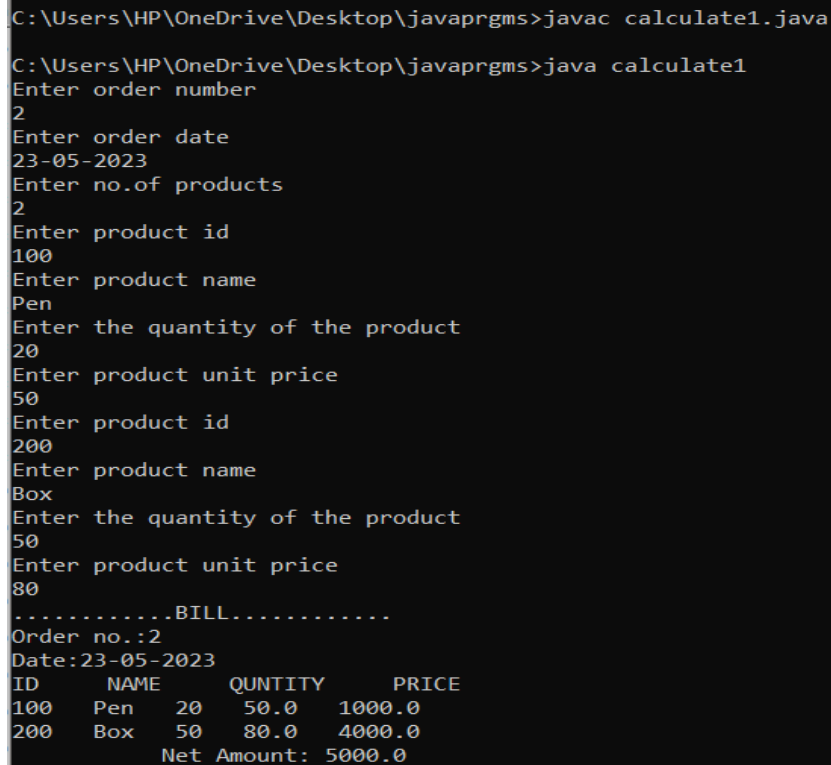
Procedure:

```
import java.util.*;
interface calculate
{
    public void calc();
}
class bill implements calculate
{
    String date,name;
    int qu,id;
    float uprice,total;
    Scanner sc= new Scanner(System.in);
    public void get()
    {
        System.out.println("Enter product id");
        id=sc.nextInt();
        System.out.println("Enter product name");
        name=sc.next();
    }
}
```

```
System.out.println("Enter the quantity of the product");
qu=sc.nextInt();
System.out.println("Enter product unit price");
uprice=sc.nextFloat();
}
public void calc()
{
total=qu*uprice;
}
public void disp()
{
System.out.println(id+" "+name+" "+qu+" "+uprice+" "+total);
}
}
public class calculate1
{
public static void main(String[] args)
{
int n,i,o;
float net=0;
String date;
Scanner sc= new Scanner(System.in);
System.out.println("Enter order number");
o=sc.nextInt();
System.out.println("Enter order date");
date=sc.next();
System.out.println("Enter no.of products");
n=sc.nextInt();
bill b[]=new bill[n];
for(i=0;i<n;i++)
{
b[i]=new bill();
```

```
b[i].get();
b[i].calc();
}
System.out.println(".....BILL.....");
System.out.println("Order no.:"+o);
System.out.println("Date:"+date);
System.out.println("ID   NAME   QANTITY   PRICE");
for(i=0;i<n;i++)
{
b[i].disp();
net=net+b[i].total;
}
System.out.println("        Net Amount: "+net);
}
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac calculate1.java
C:\Users\HP\OneDrive\Desktop\javaprgms>java calculate1
Enter order number
2
Enter order date
23-05-2023
Enter no.of products
2
Enter product id
100
Enter product name
Pen
Enter the quantity of the product
20
Enter product unit price
50
Enter product id
200
Enter product name
Box
Enter the quantity of the product
50
Enter product unit price
80
.....BILL.....
Order no.:2
Date:23-05-2023
ID      NAME      QANTITY      PRICE
100    Pen       20    50.0    1000.0
200    Box       50    80.0    4000.0
        Net Amount: 5000.0
```

Result:

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 16

Date: 04-04-2023

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.

C04:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure://shapes.java

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

```
public class shapes implements Area  
{  
    double lr,lb,ra,th,tb,ta,sa,s,cr,cc;  
    public void getRect()  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the length of the rectangle: ");  
        lr = sc.nextInt();  
        System.out.println("Enter the breadth of the rectangle: ");  
        lb = sc.nextInt();  
    }  
    public void Rectangle()  
    {  
        ra = lr*lb;  
        System.out.println("Area of Rectangle is "+ra);  
    }  
    public void getTri()  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the height of the Triangle: ");  
        th = sc.nextInt();  
        System.out.println("Enter the base of the Triangle: ");  
        tb = sc.nextInt();  
    }  
    public void Triangle()  
    {  
        ta = 0.5*th*tb;
```

```
System.out.println("Area of Triangle angle is "+ta);
}
public void getSqr()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the Side of the Square: ");
s = sc.nextInt();
}
public void Square()
{
sa = s*s;
System.out.println("Area of Square is: "+sa);
}

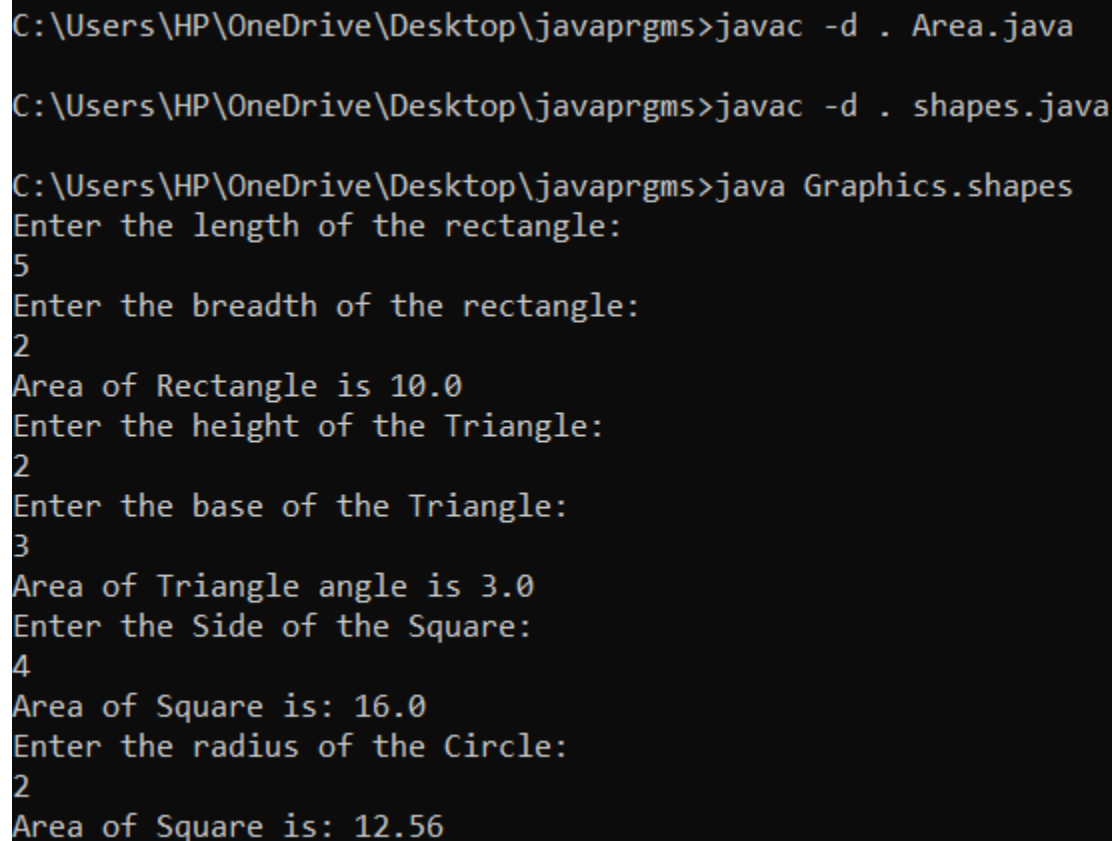
public void getCrl()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the radius of the Circle: ");
cc = sc.nextInt();
}
public void Circle()
{
cr = 3.14*cc*cc;
System.out.println("Area of Square is: "+cr);
}

}
public static void main(String[] args)
{
shapes o= new shapes();
o.getRect();
o.Rectangle();
o.getTri();
o.Triangle();
o.getSqr();
o.Square();
o.getCrl();
o.Circle();
}
}
```

//Area.java

```
package Graphics;
interface Area
{
public void Rectangle();
public void Triangle();
```

```
public void Square();  
public void Circle();  
public void getRect();  
public void getTri();  
public void getSqr();  
public void getCr1();  
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac -d . Area.java  
C:\Users\HP\OneDrive\Desktop\javaprgms>javac -d . shapes.java  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Graphics.shapes  
Enter the length of the rectangle:  
5  
Enter the breadth of the rectangle:  
2  
Area of Rectangle is 10.0  
Enter the height of the Triangle:  
2  
Enter the base of the Triangle:  
3  
Area of Triangle angle is 3.0  
Enter the Side of the Square:  
4  
Area of Square is: 16.0  
Enter the radius of the Circle:  
2  
Area of Square is: 12.56
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 17

Date: 04-04-2023

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.

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

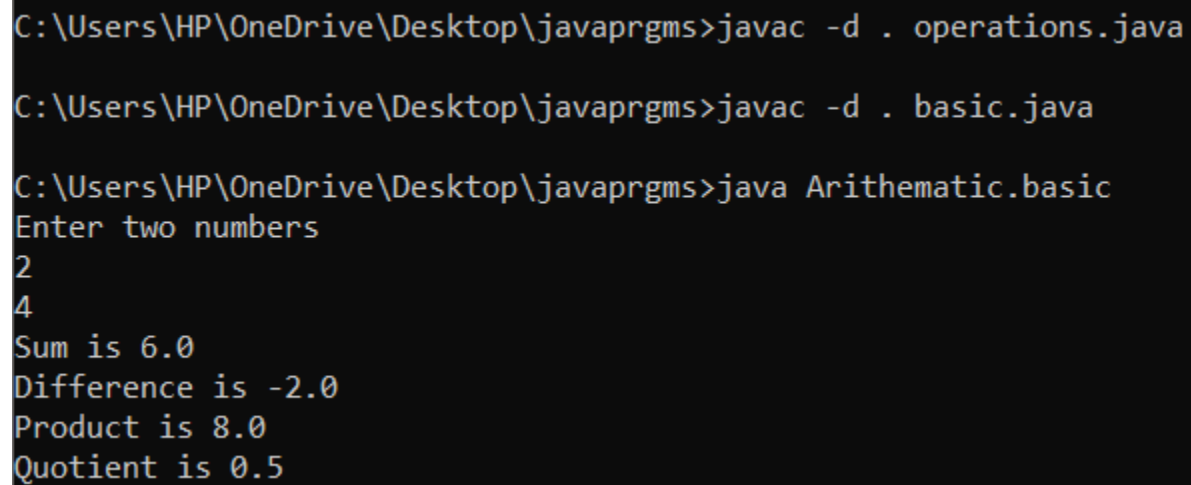
//operations.java

```
package Arithmetic;
interface operations
{
    public void input();
    public void add();
    public void subtract();
    public void multiply();
    public void division();
}
```

//basic.java

```
package Arithmetic;
import java.util.*;
public class basic implements operations
{
    double a,b,ad,dif,mult,div;
    public void input()
    {
        Scanner ab=new Scanner(System.in);
        System.out.println("Enter two numbers");
        a=ab.nextInt();
        b=ab.nextInt();
    }
}
```

```
public void add()
{
    ad=a+b;
    System.out.println("Sum is "+ad);
}
public void subtract()
{
    dif=a-b;
    System.out.println("Difference is "+dif);
}
public void multiply()
{
    mult=a*b;
    System.out.println("Product is "+mult);
}
public void division()
{
    div=a/b;
    System.out.println("Quotient is "+div);
}
public static void main(String[] args)
{
    basic o=new basic();
    o.input();
    o.add();
    o.subtract();
    o.multiply();
    o.division();
}
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac -d . operations.java
C:\Users\HP\OneDrive\Desktop\javaprgms>javac -d . basic.java
C:\Users\HP\OneDrive\Desktop\javaprgms>java Arithmetic.basic
Enter two numbers
2
4
Sum is 6.0
Difference is -2.0
Product is 8.0
Quotient is 0.5
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 18

Date: 11-04-2023

Aim:

Write a user defined exception class to authenticate the username and password.

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.Scanner;

class UsernameException extends Exception
{
    public UsernameException(String msg)
    {
        super(msg);
    }
}

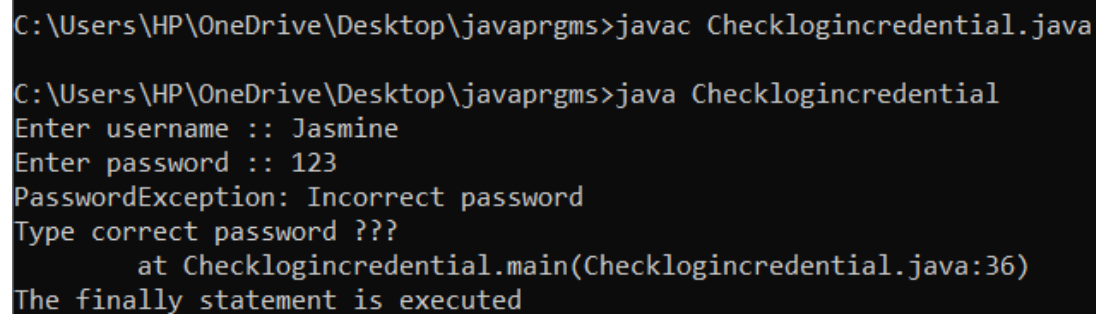
class PasswordException extends Exception
{
    public PasswordException(String msg)
    {
        super(msg);
    }
}

public class CheckLoginCredential
```

```
{  
  
public static void main(String[] args)  
  
{  
  
Scanner s = new Scanner(System.in);  
  
String username, password;  
  
System.out.print("Enter username :: ");  
  
username = s.nextLine();  
  
System.out.print("Enter password :: ");  
  
password = s.nextLine();  
  
int length = username.length();  
  
try  
  
{  
  
if(length < 6)  
  
throw new UsernameException("Username must be greater than 6 characters ???");  
  
else if(!password.equals("hello"))  
  
throw new PasswordException("Incorrect password\nType correct password ???");  
  
else  
  
System.out.println("Login Successful !!!");  
  
}  
  
catch (UsernameException u)  
  
{  
  
u.printStackTrace();  
  
}  
  
catch (PasswordException p)
```



```
{  
  
p.printStackTrace();  
  
}  
  
finally  
  
{  
  
System.out.println("The finally statement is executed");  
  
}  
  
}  
  
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Checklogincredential.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Checklogincredential  
Enter username :: Jasmine  
Enter password :: 123  
PasswordException: Incorrect password  
Type correct password ???  
    at Checklogincredential.main(Checklogincredential.java:36)  
The finally statement is executed
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 19

Date: 11-04-2023

Aim:

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

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

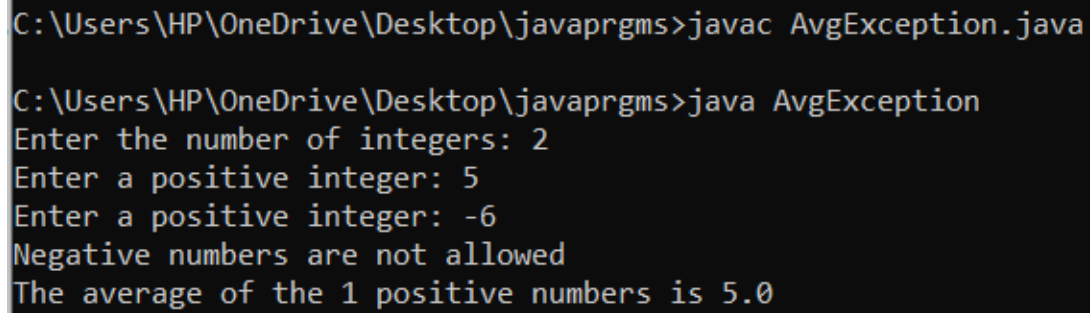
```
import java.util.*;

class NegativeNumberException extends Exception
{
    public NegativeNumberException(String message)
    {
        super(message);
    }
}

class AvgException
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of integers: ");
        int n = scanner.nextInt();

        int sum = 0;
        int count = 0;
        for (int i = 0; i < n; i++)
        {
            try
            {
```

```
System.out.print("Enter a positive integer: ");
int num = scanner.nextInt();
if (num < 0)
{
    throw new NegativeNumberException("Negative numbers are not allowed");
}
sum += num;
count++;
}
catch (NegativeNumberException e)
{
    System.out.println(e.getMessage());
}
}
if (count == 0)
{
    System.out.println("No positive numbers were entered");
}
else
{
    double average = (double) sum / count;
    System.out.println("The average of the " + count + " positive numbers is " + average);
}
}
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac AvgException.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java AvgException
Enter the number of integers: 2
Enter a positive integer: 5
Enter a positive integer: -6
Negative numbers are not allowed
The average of the 1 positive numbers is 5.0
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 20

Date: 11-04-2023

Aim:

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

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.Scanner;
class Multi extends Thread
{
public void run()
{
int num=5;
System.out.printf("___MULTIPLICATION TABLE OF 5___");
for(int i=1;i<=10;i++)
{
System.out.printf("%d * %d = %d\n",num,i,num*i);
}
}
}
class Prime extends Thread
{
public void run()
{
int i,j,flag;
Scanner sc=new Scanner(System.in);
System.out.println("___TO GENERATE First N PRIME NUMBERS___");
System.out.println("Enter the Limit");
int N=sc.nextInt();
System.out.println("Prime numbers between 1 and "+N+" are: ");
```

```
for( i=1;i<=N;i++)
{
if(i==0 || i==1)
continue;
flag=1;
for(j=2;j<=i/2;j++)
{
if(i%j==0)
{
flag=0;
break;
}
}
if(flag==1)
System.out.println(i+" ");
}
}
}
public class ThreatC
{
public static void main(String[] args) throws InterruptedException
{
Multi a = new Multi();
a.start();
a.sleep(200);
Prime b=new Prime();
b.start();
b.sleep(200);
}
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac ThreatC.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java ThreatC
__MULTIPLICATION TABLE OF 5__5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
__TO GENERATE First N PRIME NUMBERS__
Enter the Limit
5
Prime numbers between 1 and5 are:
2
3
5
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 21

Date: 13-04-2023

Aim:

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

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.Scanner;

class Fibonacci implements Runnable
{
    public void run()
    {
        int first=0,second=1,next=0;
        Scanner sc= new Scanner(System.in);
        System.out.println("___TO GENERATE FIBONACCI SERIES___");
        System.out.println("Enter the number of terms required ");
        int n =sc.nextInt();
        System.out.println("Series generated");
        for(int i=0;i<=n;i++)
        {
            System.out.println(first+" ");
            next=first+second;
            first=second;
            second=next;
        }
    }
}

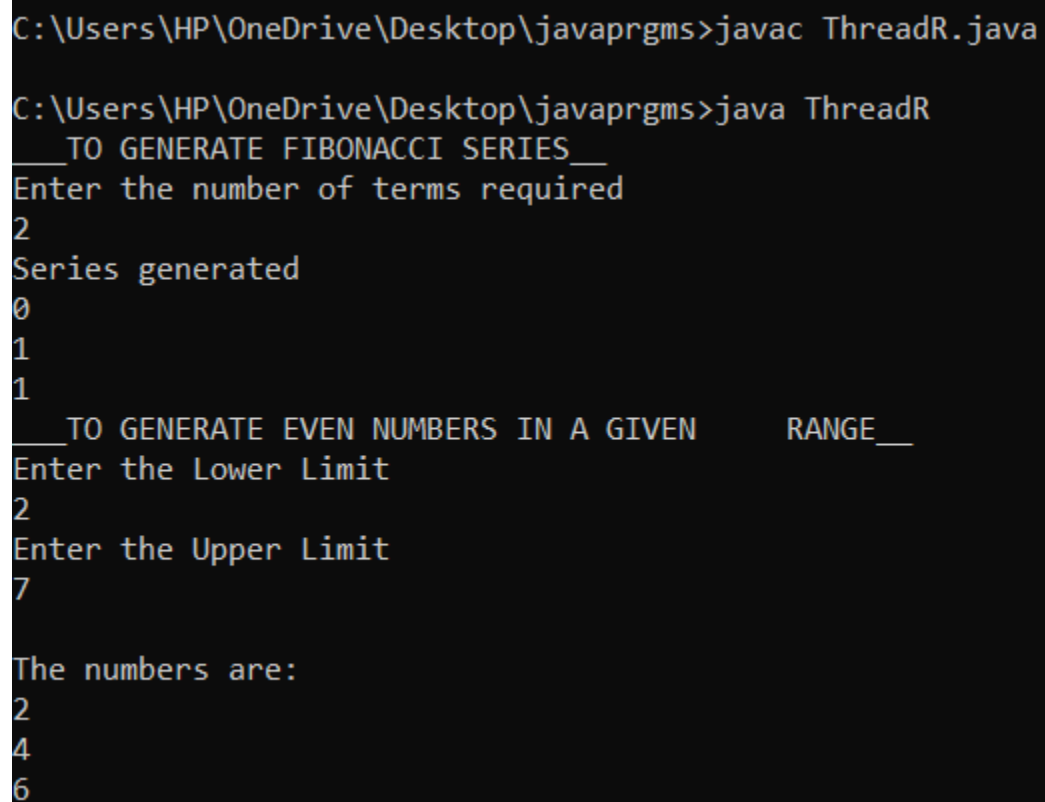
class EvenNo implements Runnable
```

```
{
public void run()
{
int lower , upper;
Scanner sc=new Scanner(System.in);
    System.out.println("___TO GENERATE EVEN NUMBERS IN A GIVEN
    RANGE___");
System.out.println("Enter the Lower Limit");
lower=sc.nextInt();
System.out.println("Enter the Upper Limit");
upper=sc.nextInt();
System.out.println(" ");
System.out.println("The numbers are: ");
for(int i=lower;i<=upper;i++)
{
if(i%2==0)
{
System.out.println(i+" ");
i++;
}
else{
break;
}
}
}
}

public class ThreadR
{
public static void main(String[] args) throws InterruptedException
{
Fibonacci obj = new Fibonacci();
Thread a =new Thread(obj);
```

```
a.start();
a.sleep(2000);
EvenNo obj1=new EvenNo();
Thread b=new Thread(obj1);
b.start();
b.sleep(1000);
}
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac ThreadR.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java ThreadR
__TO GENERATE FIBONACCI SERIES__
Enter the number of terms required
2
Series generated
0
1
1
__TO GENERATE EVEN NUMBERS IN A GIVEN RANGE__
Enter the Lower Limit
2
Enter the Upper Limit
7

The numbers are:
2
4
6
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 22

Date: 13-04-2023

Aim:

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

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

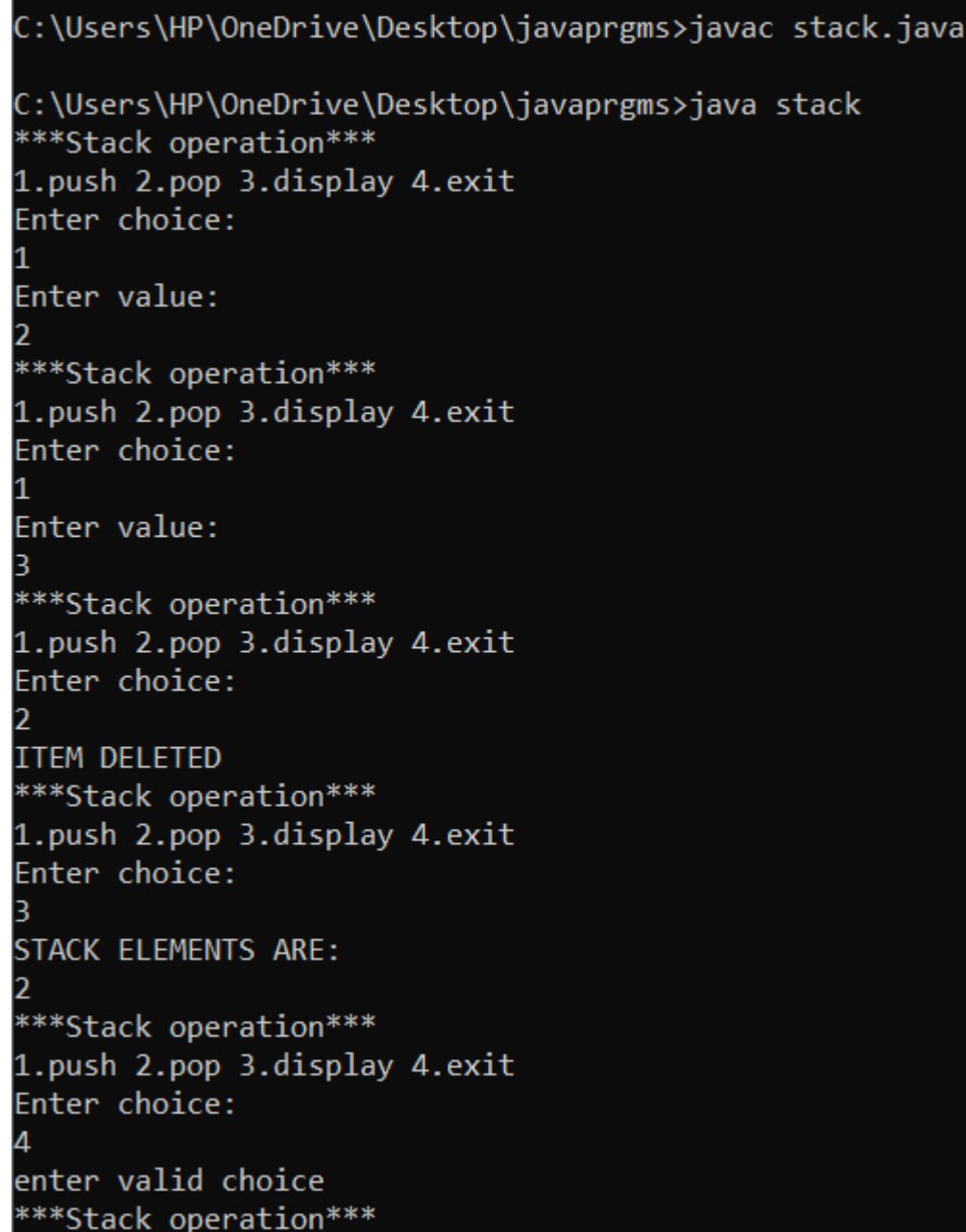
```
import java.util.*;
class stack1
{
int top=-1,n=10,i;
int a[]=new int[n];

void push()
{
if(top==n-1)
{
System.out.println("STACK OVERFLOW");
}
else
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter value:");
i=sc.nextInt();
top=top+1;
a[top]=i;
}
}
void pop()
{
if(top== -1)
{
System.out.println("STACK UNDERFLOW");
}
else
{
top=top-1;
System.out.println("ITEM DELETED");
}
```

```
}  
}  
void display()  
{  
System.out.println("STACK ELEMENTS ARE: ");  
for(int i=top;i>=0;i--)  
{  
System.out.println(a[i]);  
}  
}  
public static void main(String[] args)  
{  
int n=10,ch;  
Scanner sc = new Scanner(System.in);  
stack1 s=new stack1();  
do{  
System.out.println("***Stack operation***");  
System.out.println("1.push 2.pop 3.display 4.exit");  
System.out.println("Enter choice:");  
ch=sc.nextInt();  
switch (ch)  
{  
case 1:  
{  
s.push();  
break;  
}  
case 2:  
{  
s.pop();  
break;  
}  
case 3:  
{  
s.display();  
break;  
}  
  
default:  
{  
System.out.println("enter valid choice");  
}  
}  
  
}  
while(ch !=0);
```

```
}  
  
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac stack.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java stack  
***Stack operation***  
1.push 2.pop 3.display 4.exit  
Enter choice:  
1  
Enter value:  
2  
***Stack operation***  
1.push 2.pop 3.display 4.exit  
Enter choice:  
1  
Enter value:  
3  
***Stack operation***  
1.push 2.pop 3.display 4.exit  
Enter choice:  
2  
ITEM DELETED  
***Stack operation***  
1.push 2.pop 3.display 4.exit  
Enter choice:  
3  
STACK ELEMENTS ARE:  
2  
***Stack operation***  
1.push 2.pop 3.display 4.exit  
Enter choice:  
4  
enter valid choice  
***Stack operation***
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 23

Date: 18-04-2023

Aim:

Using generic method perform Bubble sort.

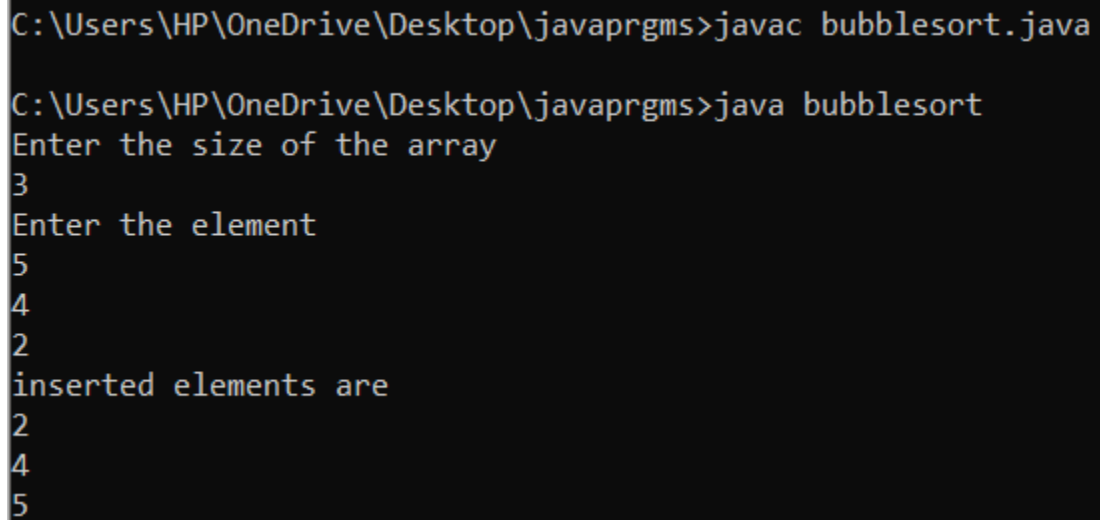
CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.*;
class bubblesort
{
public static void main(String [] args)
{
Scanner sc= new Scanner(System.in);
int i,j;
bubblesort obj=new bubblesort ();
System.out.println("Enter the size of the array");
int n=sc.nextInt();
int arr[]=new int[n];
System.out.println("Enter the element");
for(i=0;i<n;i++)
{
arr[i]=sc.nextInt();
}
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
if(arr[i]>arr[j])
{
int temp=arr[i];
arr[i]=arr[j];
arr[j]=temp;
}
}
}
System.out.println("inserted elements are");
for(i=0;i<n;i++)
{
```

```
System.out.println(arr[i]);  
}  
}  
}
```

Output Screenshot:

A screenshot of a Windows command prompt window showing the execution of a Java program. The user enters the command to compile and then run a file named 'bubblesort.java'. The program prompts for the size of the array (3) and then for three elements (5, 4, 2). It then displays the sorted elements: 2, 4, 5.

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac bubblesort.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java bubblesort  
Enter the size of the array  
3  
Enter the element  
5  
4  
2  
inserted elements are  
2  
4  
5
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 24

Date: 18-04-2023

Aim:

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

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.ArrayList;
import java.util.Comparator;
import java.util.Scanner;
import java.util.Collections;
public class ArrayListDemo
{
    public static void insert(ArrayList<String> list)
    {
        Scanner Snr = new Scanner(System.in);
        System.out.print("Enter the string to be added: ");
        String val = Snr.next();
        list.add(val);
        System.out.println("Value inserted successfully");
    }
    public static void deleteAll(ArrayList<String> list)
    {
        list.clear();
        System.out.println("ArrayList successfully cleared");
    }
    public static void find(ArrayList<String> list)
    {
        Scanner Snr = new Scanner(System.in);
        System.out.print("Enter the index to get: ");
        int val = Snr.nextInt();
        System.out.println(list.get(val));
    }
    public static void delete(ArrayList<String> list)
    {
        Scanner Snr = new Scanner(System.in);
        System.out.print("Enter the index to delete: ");
```

```
int val = Snr.nextInt();
list.remove(val);
System.out.println("Value successfully removed");
}
public static void len(ArrayList<String> list)
{
System.out.println("Length of Array List: " + list.size());
}
public static void sortlist(ArrayList<String> list)
{
list.sort(Comparator.naturalOrder());
System.out.println("Sorted Array List: " + list);
}
public static void sortRev(ArrayList<String> list)
{
list.sort(Comparator.reverseOrder());
System.out.println("Reverse Order Sorted Array List: " + list);
}
public static void max(ArrayList<String> list)
{
String large = list.get(0);
int len = large.length();
for(int i=1; i<list.size(); i++)
{
String cand = list.get(i);
if(len < cand.length())
{
large = list.get(i);
len = large.length();
}
}
System.out.println("Largest String: " + large);
}
public static void display(ArrayList<String> list)
{
System.out.println(list);
}
public static void main(String[] args)
{
ArrayList<String> list = new ArrayList<String>();
Scanner Snr = new Scanner(System.in);
int choice=0;
while(choice!=10)
{
System.out.println("");
System.out.println(":: ARRAYLIST ::");
```

```
System.out.println("1. Insert String");
System.out.println("2. Clear List");
System.out.println("3. Get Specific Element by Index");
System.out.println("4. Remove Specific Element by Index");
System.out.println("5. Length of List");
System.out.println("6. Sort Array List");
System.out.println("7. Reverse Order Sort of Array List");
System.out.println("8. Get Longest value from List");
System.out.println("9. Display");
System.out.println("10. Exit");
System.out.print("Enter your choice: ");
choice= Snr.nextInt();
switch(choice)
{
case 1:
insert(list);
break;
case 2:
deleteAll(list);
break;
case 3:
find(list);
break;
case 4:
delete(list);
break;
case 5:
len(list);
break;
case 6:
sortlist(list);
break;
case 7:
sortRev(list);
break;
case 8:
max(list);
break;
case 9:
display(list);
break;
case 10:
System.exit(0);
break;
default:
System.out.println("Enter a valid choice!");
```

```
}  
}  
}  
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac ArrayListDemo.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java ArrayListDemo  
  
:: ARRAYLIST ::  
1. Insert String  
2. Clear List  
3. Get Specific Element by Index  
4. Remove Specific Element by Index  
5. Length of List  
6. Sort Array List  
7. Reverse Order Sort of Array List  
8. Get Longest value from List  
9. Display  
10. Exit  
Enter your choice: 1  
Enter the string to be added: San  
Value inserted successfully  
  
:: ARRAYLIST ::  
1. Insert String  
2. Clear List  
3. Get Specific Element by Index  
4. Remove Specific Element by Index  
5. Length of List  
6. Sort Array List  
7. Reverse Order Sort of Array List  
8. Get Longest value from List  
9. Display  
10. Exit  
Enter your choice: 1  
Enter the string to be added: List  
Value inserted successfully  
  
:: ARRAYLIST ::  
1. Insert String  
2. Clear List  
3. Get Specific Element by Index  
4. Remove Specific Element by Index  
5. Length of List  
6. Sort Array List  
7. Reverse Order Sort of Array List  
8. Get Longest value from List  
9. Display  
10. Exit
```

```
Enter your choice: 3
Enter the index to get: 0
San

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 9
[San, List]

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 8
Largest String: List

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 5
Length of Array List: 2

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 6
Sorted Array List: [List, San]
```

```
:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 7
Reverse Order Sorted Array List: [San, List]

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 4
Enter the index to delete: 1
Value successfully removed

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 9
[San]

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 10
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 25

Date: 18-04-2023

Aim:

Program to remove all the elements from a linked list.

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.LinkedList;
import java.util.*;
public class LinkedListDemo
{
    public static void main(String[] args)
    {
        LinkedList<String> list = new LinkedList<String>();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no of linked list: ");
        int n=sc.nextInt();
        for(int i=0;i<n;i++)
        {
            System.out.println("Enter the items to the linked list");
            String item=sc.next();
            list.add(item);
        }
        System.out.println("Original LinkedList: "+list);
        list.clear();
        System.out.println("After clearing Linkedlist:" +list);
    }
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac LinkedListDemo.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java LinkedListDemo
Enter the no of linked list:
4
Enter the items to the linked list
25
Enter the items to the linked list
32
Enter the items to the linked list
16
Enter the items to the linked list
48
Original LinkedList: [25, 32, 16, 48]
After clearing Linkedlist:[]
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 26

Date: 27-04-2023

Aim:

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

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.*;
public class StackDemo
{
    public static void main(String[] args)
    {
        Stack<String> stack = new Stack<String>();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of elemnts of Stack : ");
        int n=sc.nextInt();
        for(int i=0;i<n;i++)
        {
            System.out.println("Enter the items");
            String list=sc.next();
            stack.add(list);
        }
        System.out.println("The items of the stack are: ");
        System.out.println(stack);

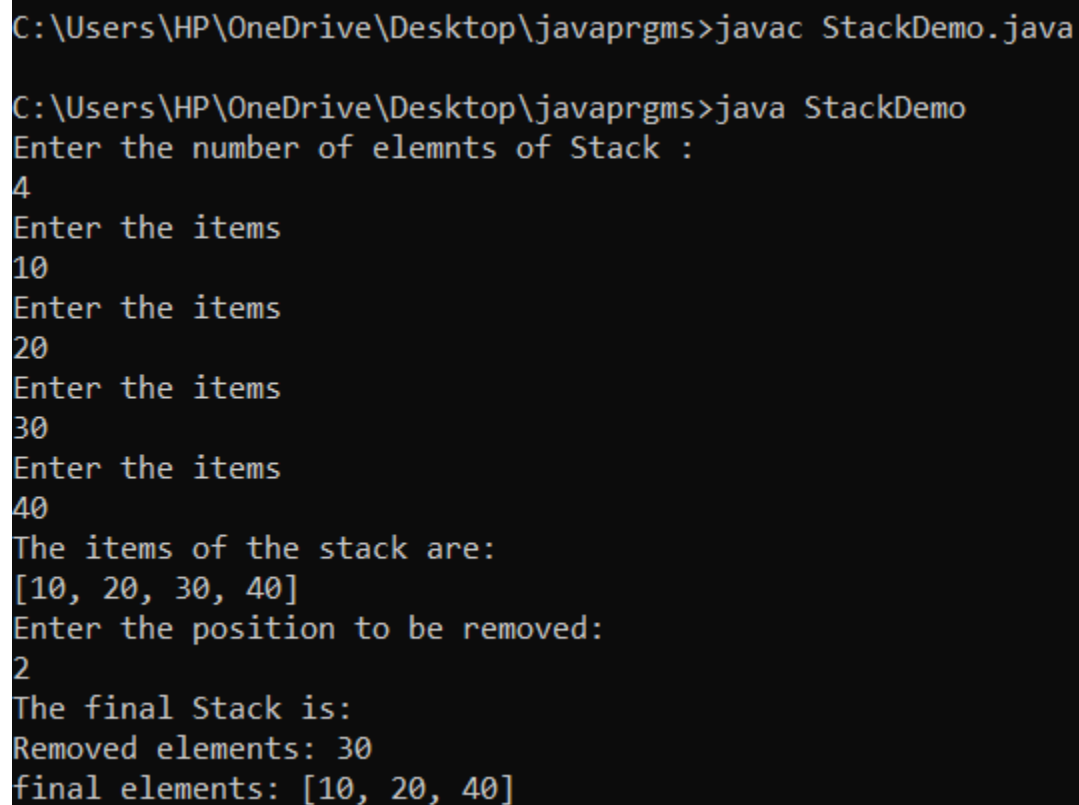
        System.out.println("Enter the position to be removed: ");
        int remove=sc.nextInt();

        //removing the elements
        String rem_ele=stack.remove(remove);
```



```
System.out.println("The final Stack is: ");  
System.out.println("Removed elements: "+rem_ele);  
System.out.println("final elements: "+stack);  
}  
}
```

Output Screenshot:



The screenshot shows a command prompt window with the following text:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac StackDemo.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java StackDemo  
Enter the number of elemnts of Stack :  
4  
Enter the items  
10  
Enter the items  
20  
Enter the items  
30  
Enter the items  
40  
The items of the stack are:  
[10, 20, 30, 40]  
Enter the position to be removed:  
2  
The final Stack is:  
Removed elements: 30  
final elements: [10, 20, 40]
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 27

Date: 01-06-2023

Aim:

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

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

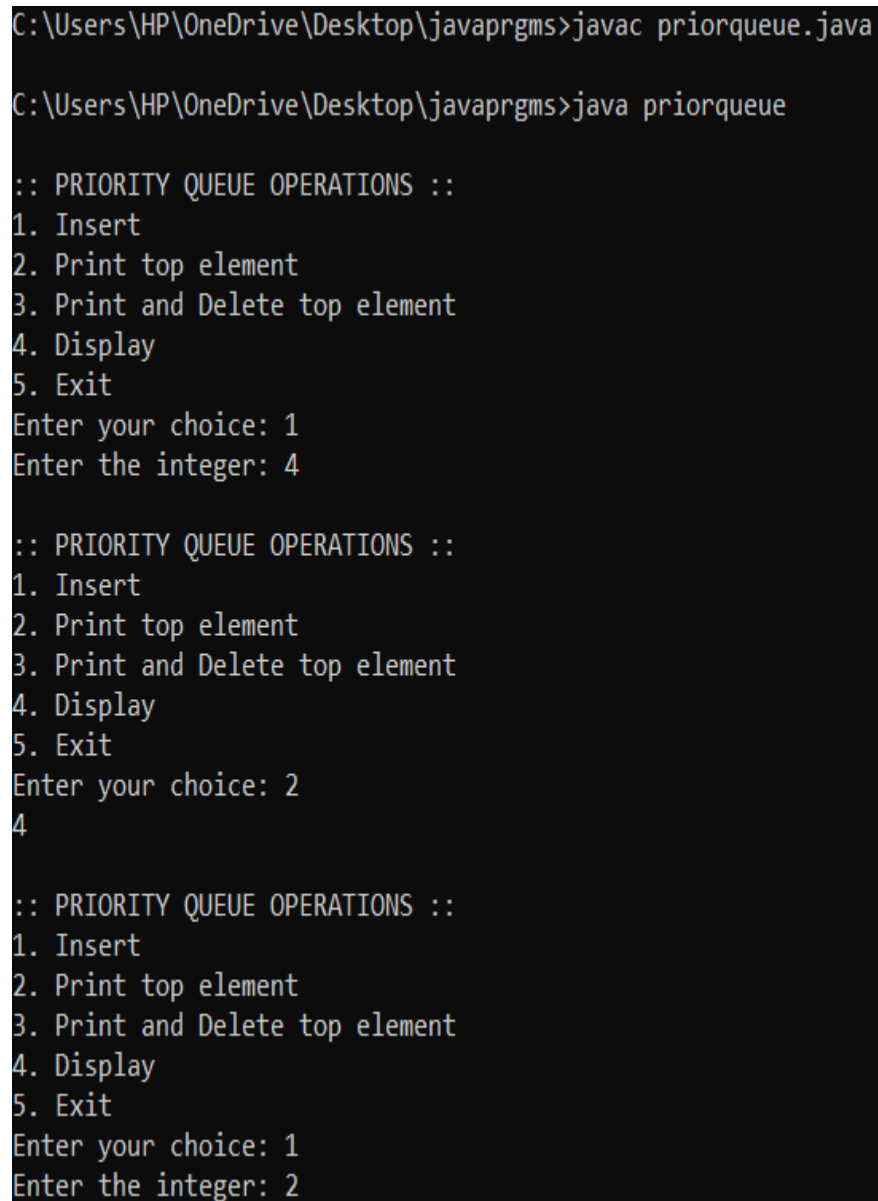
```
import java.util.*;
import java.util.PriorityQueue;
public class priorqueue
{
public static void insert(PriorityQueue<Integer> pq)
{
Scanner Snr = new Scanner(System.in);
System.out.print("Enter the integer: ");
int num = Snr.nextInt();
pq.add(num);
}
public static void last(PriorityQueue<Integer> pq)
{
System.out.println(pq.peek());
}
public static void del(PriorityQueue<Integer> pq)
{
System.out.println(pq.poll() + " removed");
}
public static void display(PriorityQueue<Integer> pq)
{
System.out.println("Priority Queue: " + pq);
}
```

```
}  
public static void main(String[] args)  
{  
    Scanner Snr =new Scanner(System.in);  
    PriorityQueue<Integer> pq = new PriorityQueue<Integer>();  
    int choice = 0;  
    while(choice != 5){  
        System.out.println(" ");  
        System.out.println(":: PRIORITY QUEUE OPERATIONS ::");  
        System.out.println("1. Insert");  
        System.out.println("2. Print top element");  
        System.out.println("3. Print and Delete top element");  
        System.out.println("4. Display");  
        System.out.println("5. Exit");  
        System.out.print("Enter your choice: ");  
        choice = Snr.nextInt();  
        switch(choice){  
            case 1:  
                insert(pq);  
                break;  
            case 2:  
                last(pq);  
                break;  
            case 3:  
                del(pq);  
                break;  
            case 4:  
                display(pq);  
                break;  
            case 5:  
                System.exit(0);  
                break;
```

default:

```
System.out.println("Enter a valid choice!");  
}  
}  
}  
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac priorqueue.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java priorqueue  
  
:: PRIORITY QUEUE OPERATIONS ::  
1. Insert  
2. Print top element  
3. Print and Delete top element  
4. Display  
5. Exit  
Enter your choice: 1  
Enter the integer: 4  
  
:: PRIORITY QUEUE OPERATIONS ::  
1. Insert  
2. Print top element  
3. Print and Delete top element  
4. Display  
5. Exit  
Enter your choice: 2  
4  
  
:: PRIORITY QUEUE OPERATIONS ::  
1. Insert  
2. Print top element  
3. Print and Delete top element  
4. Display  
5. Exit  
Enter your choice: 1  
Enter the integer: 2
```

```
:: PRIORITY QUEUE OPERATIONS ::
1. Insert
2. Print top element
3. Print and Delete top element
4. Display
5. Exit
Enter your choice: 4
Priority Queue: [2, 4]

:: PRIORITY QUEUE OPERATIONS ::
1. Insert
2. Print top element
3. Print and Delete top element
4. Display
5. Exit
Enter your choice: 3
2 removed

:: PRIORITY QUEUE OPERATIONS ::
1. Insert
2. Print top element
3. Print and Delete top element
4. Display
5. Exit
Enter your choice: 4
Priority Queue: [4]

:: PRIORITY QUEUE OPERATIONS ::
1. Insert
2. Print top element
3. Print and Delete top element
4. Display
5. Exit
Enter your choice: 5
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 28

Date: 01-06-2023

Aim:

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

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

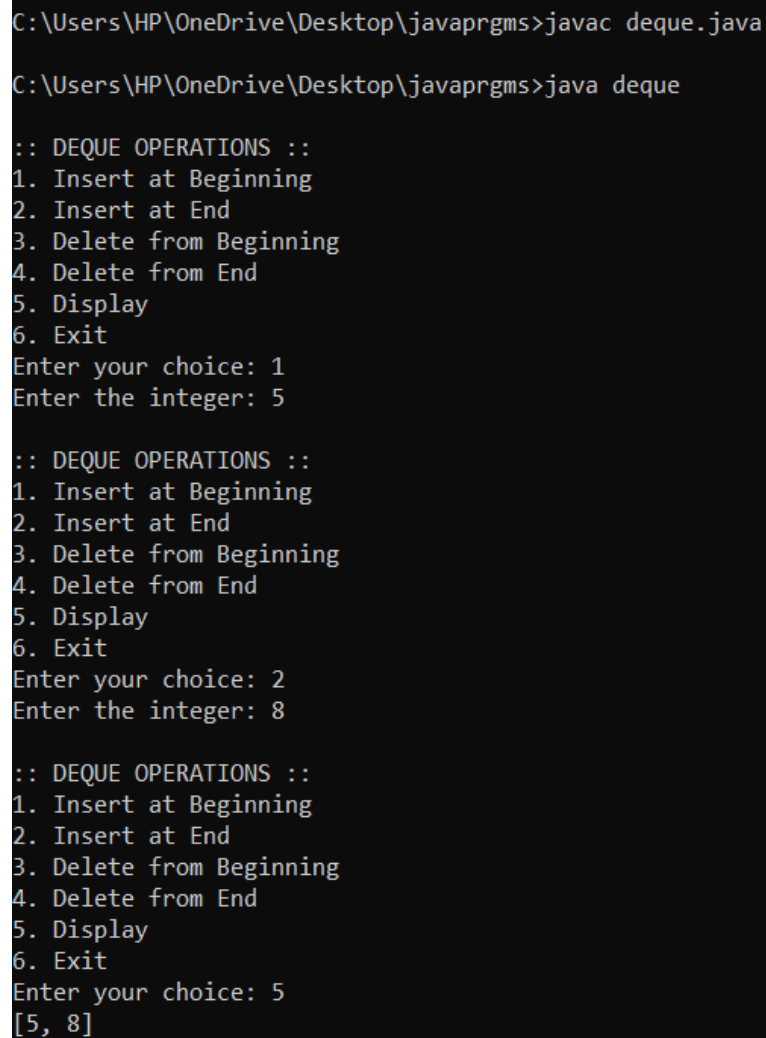
Procedure:

```
import java.util.*;
import java.util.ArrayDeque;
import java.util.Deque;
public class deque{
    public static void insertBeg(Deque<Integer> dq){
        Scanner Snr = new Scanner(System.in);
        System.out.print("Enter the integer: ");
        int num = Snr.nextInt();
        dq.addFirst(num);
    }
    public static void insertEnd(Deque<Integer> dq){
        Scanner Snr = new Scanner(System.in);
        System.out.print("Enter the integer: ");
        int num = Snr.nextInt();
        dq.addLast(num);
    }
    public static void delBeg(Deque<Integer> dq){
        System.out.println(dq.removeFirst() + " removed");
    }
    public static void delEnd(Deque<Integer> dq){
        System.out.println(dq.removeLast() + " removed");
    }
}
```

```
public static void main(String[] args){
Scanner Snr =new Scanner(System.in);
Deque<Integer> dq = new ArrayDeque<Integer>();
int choice = 0;
while(choice != 6){
System.out.println(" ");
System.out.println(":: DEQUE OPERATIONS ::");
System.out.println("1. Insert at Beginning");
System.out.println("2. Insert at End");
System.out.println("3. Delete from Beginning");
System.out.println("4. Delete from End");
System.out.println("5. Display");
System.out.println("6. Exit");
System.out.print("Enter your choice: ");
choice = Snr.nextInt();
switch(choice){
case 1:
insertBeg(dq);
break;
case 2:
insertEnd(dq);
break;
case 3:
delBeg(dq);
break;
case 4:
delEnd(dq);
break;
case 5:
System.out.println(dq);
break;
case 6:
```

```
System.exit(0);  
  
break;  
  
default:  
  
System.out.println("Enter a valid choice!");  
  
}  
  
}  
  
}  
  
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac deque.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java deque  
  
:: DEQUE OPERATIONS ::  
1. Insert at Beginning  
2. Insert at End  
3. Delete from Beginning  
4. Delete from End  
5. Display  
6. Exit  
Enter your choice: 1  
Enter the integer: 5  
  
:: DEQUE OPERATIONS ::  
1. Insert at Beginning  
2. Insert at End  
3. Delete from Beginning  
4. Delete from End  
5. Display  
6. Exit  
Enter your choice: 2  
Enter the integer: 8  
  
:: DEQUE OPERATIONS ::  
1. Insert at Beginning  
2. Insert at End  
3. Delete from Beginning  
4. Delete from End  
5. Display  
6. Exit  
Enter your choice: 5  
[5, 8]
```



```
:: DEQUE OPERATIONS ::  
1. Insert at Beginning  
2. Insert at End  
3. Delete from Beginning  
4. Delete from End  
5. Display  
6. Exit
```

Enter your choice: 1

Enter the integer: 6

```
:: DEQUE OPERATIONS ::  
1. Insert at Beginning  
2. Insert at End  
3. Delete from Beginning  
4. Delete from End  
5. Display  
6. Exit
```

Enter your choice: 5

[6, 5, 8]

```
:: DEQUE OPERATIONS ::  
1. Insert at Beginning  
2. Insert at End  
3. Delete from Beginning  
4. Delete from End  
5. Display  
6. Exit
```

Enter your choice: 3

6 removed

```
:: DEQUE OPERATIONS ::  
1. Insert at Beginning  
2. Insert at End  
3. Delete from Beginning  
4. Delete from End  
5. Display  
6. Exit
```

Enter your choice: 4

8 removed

```
:: DEQUE OPERATIONS ::  
1. Insert at Beginning  
2. Insert at End  
3. Delete from Beginning  
4. Delete from End  
5. Display  
6. Exit
```

Enter your choice: 5

[5]

```
:: DEQUE OPERATIONS ::  
1. Insert at Beginning  
2. Insert at End  
3. Delete from Beginning  
4. Delete from End  
5. Display  
6. Exit
```

Enter your choice: 6

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 29

Date: 08-06-2023

Aim:

Program to demonstrate the creation of Set object using the LinkedHashSet class.

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.*;
import java.util.LinkedHashSet;
import java.util.Set;
public class set
{
    public static void insert(Set<Integer> st)
    {
        Scanner Snr = new Scanner(System.in);
        System.out.print("Enter the integer: ");
        int line = Snr.nextInt();
        st.add(line);
    }
    public static void del(Set<Integer> st)
    {
        Scanner Snr = new Scanner(System.in);
        System.out.print("Enter the position: ");
        int pos = Snr.nextInt();
        pos = pos-1;
        System.out.println(st.remove(pos) + " removed");
    }
    public static void display(Set<Integer> st)
    {
```

```
System.out.println("Stack: " + st);
}
public static void delAll(Set<Integer> st)
{
    st.clear();
    System.out.println("Set successfully cleared");
}
public static void main(String[] args)
{
    Set<Integer> st = new LinkedHashSet<Integer>();
    Scanner Snr = new Scanner(System.in);
    int choice1 = 0, choice2 = 0, choice3 = 0;
    while(choice1 != 5)
    {
        System.out.println(":: SET OPERATIONS ::");
        System.out.println("1. Insert");
        System.out.println("2. Delete");
        System.out.println("3. Display");
        System.out.println("4. Clear All");
        System.out.println("5. Exit");
        System.out.print("Enter your choice: ");
        choice1 = Snr.nextInt();
        switch(choice1){
            case 1:
                insert(st);
                break;
            case 2:
                del(st);
                break;
            case 3:
                display(st);
                break;
```

case 4:

delAll(st);

break;

case 5:

System.exit(0);

break;

default:

System.out.println("Enter a valid choice");

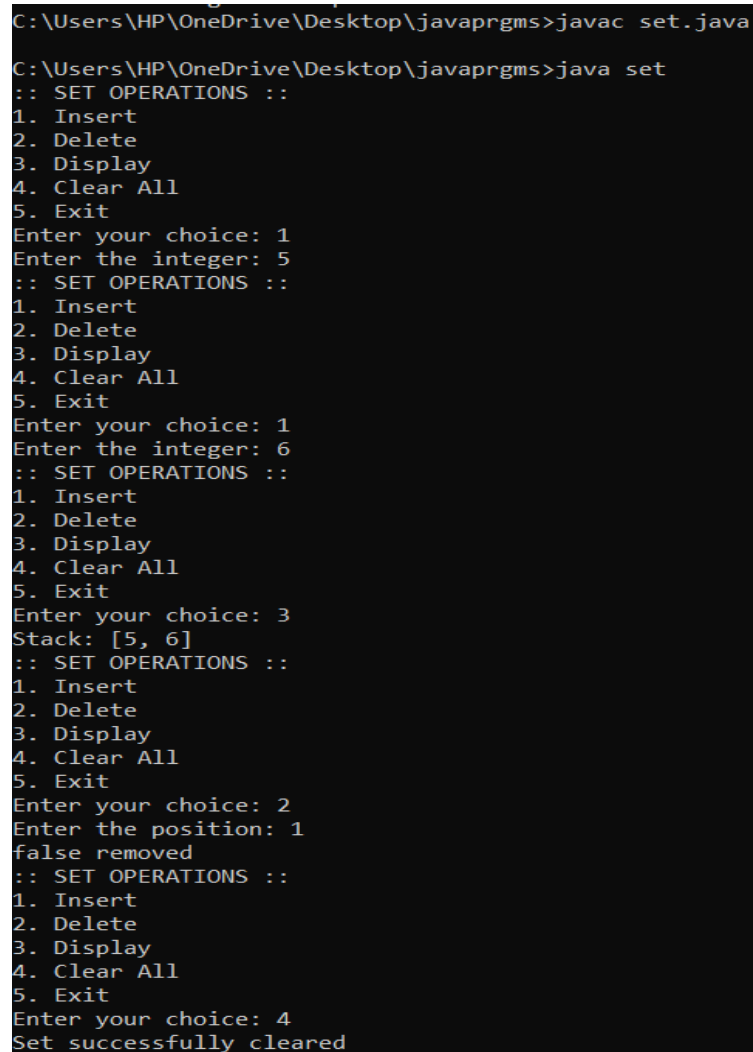
}

}

}

}

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac set.java
C:\Users\HP\OneDrive\Desktop\javaprgms>java set
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 1
Enter the integer: 5
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 1
Enter the integer: 6
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 3
Stack: [5, 6]
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 2
Enter the position: 1
false removed
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 4
Set successfully cleared
```

```
:: SET OPERATIONS ::  
1. Insert  
2. Delete  
3. Display  
4. Clear All  
5. Exit  
Enter your choice: 3  
Stack: []  
:: SET OPERATIONS ::  
1. Insert  
2. Delete  
3. Display  
4. Clear All  
5. Exit  
Enter your choice: 5
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 30

Date: 08-06-2023

Aim:

Write a Java program to compare two hash set.

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.*;

public class Mainn
{
    public static void main(String[] args)
    {
        int n;
        String str;
        HashSet<String> set1= new HashSet<String>();
        System.out.println("HashSet 1");
        System.out.println("Enter No. of countries:");
        Scanner sc=new Scanner(System.in);
        n=sc.nextInt();
        System.out.println("Enter the name of countries:");
        Scanner sc1=new Scanner(System.in);
        for(int i=0;i<n;i++)
        {
            str=sc1.nextLine();
            set1.add(str);
        }

        System.out.println("HashSet 2");
        HashSet<String> set2= new HashSet<String>();
```

```
System.out.println("Enter No. of countries:");
n=sc.nextInt();

System.out.println("Enter the name of countries:");
for(int i=0;i<n;i++)
{
    str=sc1.nextLine();
    set2.add(str);
}
System.out.println("Set1:"+set1);
System.out.println("Set2:"+set2);

HashSet<String> a= new HashSet<String>(set1);
a.addAll(set2);

System.out.println("Union of country set:"+a);
HashSet<String> b= new HashSet<String>(set1);
b.retainAll(set2);

System.out.println("Intersection of country set:"+b);
HashSet<String> c= new HashSet<String>(set1);
c.removeAll(set2);

System.out.println("Difference of country set:"+c);
}

}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Mainn.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Mainn
HashSet 1
Enter No. of countries:
2
Enter the name of countries:
London
Canada
HashSet 2
Enter No. of countries:
3
Enter the name of countries:
Korea
Japan
China
Set1:[Canada, London]
Set2:[Japan, China, Korea]
Union of country set:[Canada, Japan, China, London, Korea]
Intersection of country set:[]
Difference of country set:[Canada, London]
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 31

Date: 13-06-2023

Aim:

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

CO4:

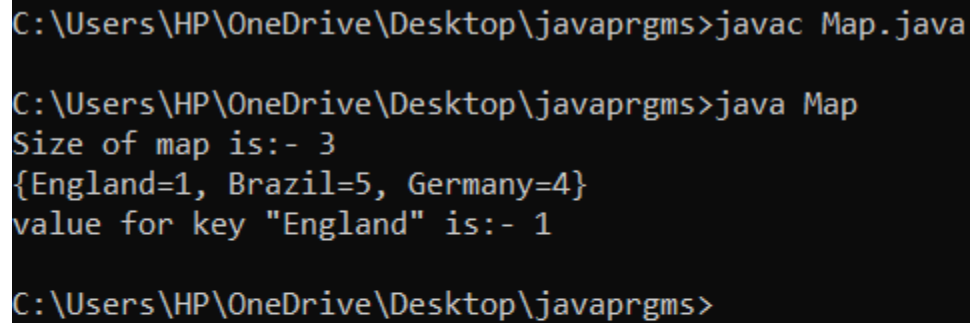
Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.HashMap;

public class Map
{
    public static void main(String[] args)
    {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("Germany", 4);
        map.put("England", 1);
        map.put("Brazil", 5);
        System.out.println("Size of map is:- " + map.size());
        // Printing elements in object of Map
        System.out.println(map);
        // Checking if a key is present and if
        // present, print value by passing
        // random element
        if (map.containsKey("England"))
        {
            // Mapping
            Integer a = map.get("England");
            // Printing value for the corresponding key
            System.out.println("value for key" + " \"England\" is:- " + a);
        }
    }
}
```

```
}  
}  
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Map.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Map  
Size of map is:- 3  
{England=1, Brazil=5, Germany=4}  
value for key "England" is:- 1  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 32

Date: 13-06-2023

Aim:

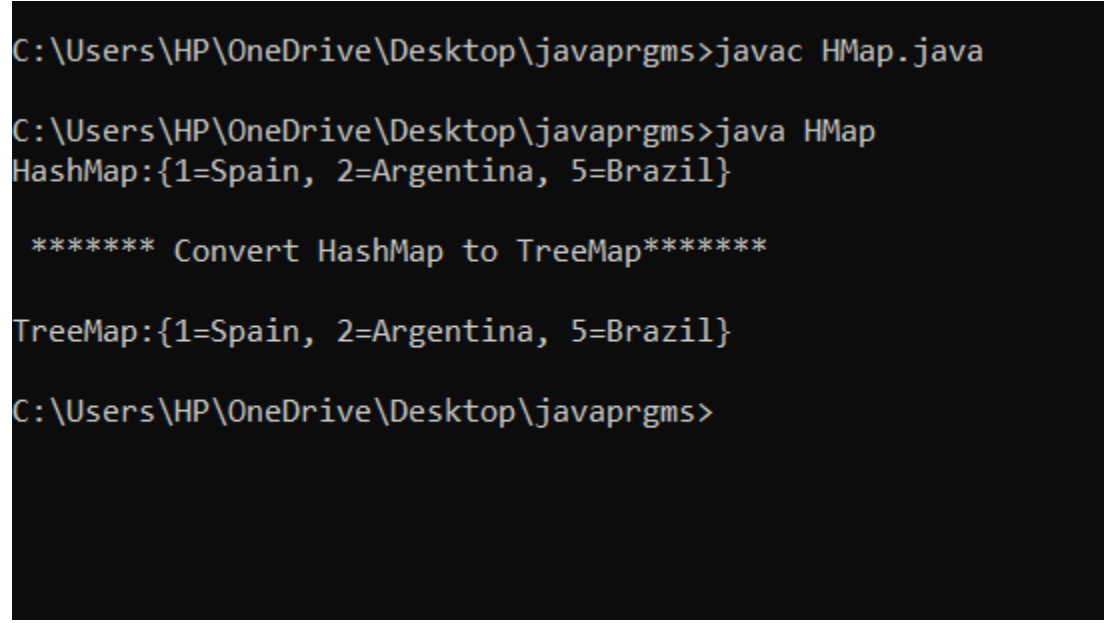
Program to Convert HashMap to TreeMap.

CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure:

```
import java.util.*;
import java.util.Map;
public class HMap
{
public static void main(String[] args)
{
// TODO Auto-generated method stub
Map<Integer,String> hm=new LinkedHashMap<>();
hm.put(1,"England");
hm.put(1,"Spain");
hm.put(2,"France");
hm.put(5,"Brazil");
hm.put(2,"Argentina");
System.out.println("HashMap:"+hm);
Map<Integer,String> tm=new TreeMap<>(hm);
System.out.println("\n ***** Convert HashMap to TreeMap*****\n");
System.out.println("TreeMap:"+tm);
}
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac HMap.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java HMap
HashMap:{1=Spain, 2=Argentina, 5=Brazil}

***** Convert HashMap to TreeMap*****

TreeMap:{1=Spain, 2=Argentina, 5=Brazil}

C:\Users\HP\OneDrive\Desktop\javaprgms>
```

Result:

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 33

Date: 20-06-2023

Aim:

Program to draw Circle, Rectangle, Line in Applet.

CO5:

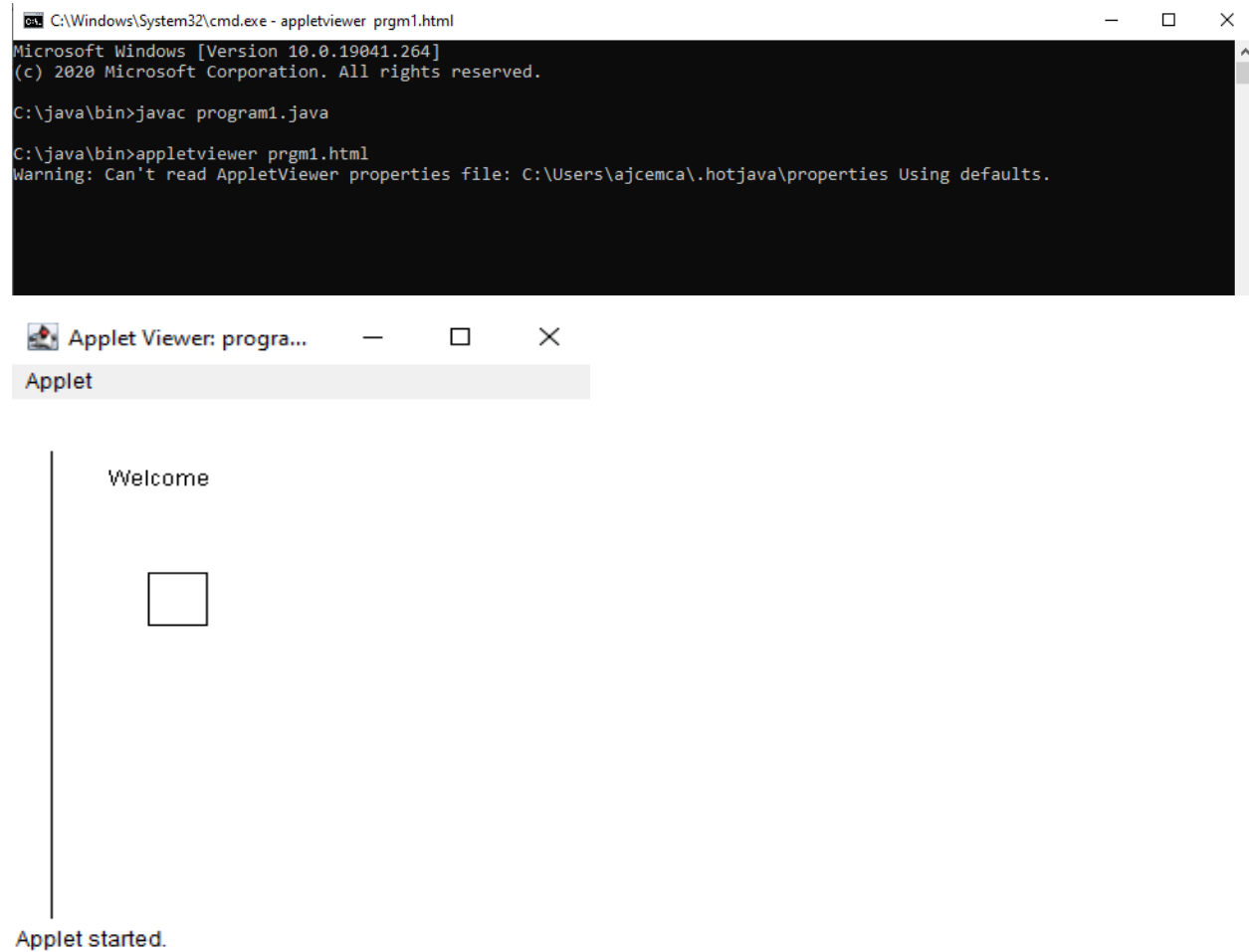
Design applications using files and network concepts

Procedure:**program1.java**

```
import java.applet.Applet;
import java.awt.Graphics;
public class program1 extends Applet
{
public void paint(Graphics g)
{
g.drawString("Welcome",50,50);
g.drawLine(20,30,20,300);
g.drawRect(70,100,30,30);
}
}
```

prgm1.html

```
<html>
<body>
<applet code="program1.class" width="300" height="300">
</applet>
</body>
</html>
```

Output Screenshot:

The screenshot shows two windows. The top window is a command prompt titled "C:\Windows\System32\cmd.exe - appletviewer prgm1.html". It displays the following text:

```
Microsoft Windows [Version 10.0.19041.264]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\java\bin>javac program1.java

C:\java\bin>appletviewer prgm1.html
Warning: Can't read AppletViewer properties file: C:\Users\ajcemca\.hotjava\properties Using defaults.
```

The bottom window is titled "Applet Viewer: progra...". It has a title bar with standard Windows window controls. Below the title bar, the word "Applet" is displayed. The main content area shows the text "Welcome" and a small square icon. At the bottom left of the window, the text "Applet started." is visible.

Result:

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 34

Date: 20-06-2023

Aim:

Program to find maximum of three numbers using AWT.

CO5:

Design applications using files and network concepts

Procedure:**maxThree.java**

```
import java.awt.*;

import java.awt.Graphics;

import java.applet.*;

import java.awt.event.*;

public class maxThree extends Applet implements ActionListener

{

    Label l1,l2,l3,l4;

    TextField tf1,tf2,tf3,tf4;

    Button btn;

    public void init()

    {

        l1= new Label("Number 1:");

        tf1=new TextField();

        l2=new Label("Number 2:");

        tf2=new TextField();

        l3=new Label("Number 3:");

        tf3=new TextField();
```

```
l4 = new Label("MAXIMUM:");

tf4= new TextField();

btn= new Button("Submit");

setLayout(null);

l1.setBounds(450,50,70,20);

tf1.setBounds(520,50,100,20);

l2.setBounds(450,80,70,20);

tf2.setBounds(520,80,100,20);

l3.setBounds(450,110,70,20);

tf3.setBounds(520,110,100,20);

l4.setBounds(450,140,70,20);

tf4.setBounds(520,140,100,20);

btn.setBounds(450,290,80,30);

add(l1);

add(l2);

add(l3);

add(l4);

add(tf1);

add(tf2);

add(tf3);

add(tf4);

add(btn);

btn.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{
```



```
if(e.getSource()==btn)

{

float n1,n2,n3,max;

n1=Float.parseFloat(tf1.getText());

n2=Float.parseFloat(tf2.getText());

n3=Float.parseFloat(tf3.getText());

if(n1>n2 && n1>n3)

{

max=n1;

}

else if(n2>n1 && n2>n3)

{

max=n2;

}

else

{

max=n3;

}

tf4.setText(String.valueOf(max));

}

}

/*public static void main(String[] args)

{

maximum m = new maximum();

m.setTitle("Maximum");

m.setSize(150,150);
```

```
m.setVisible(true);
```

```
 */
```

```
}
```

maxThree.html

```
<html>
```

```
<head>
```

```
<title>Maximum of Three</title>
```

```
</head>
```

```
<body>
```

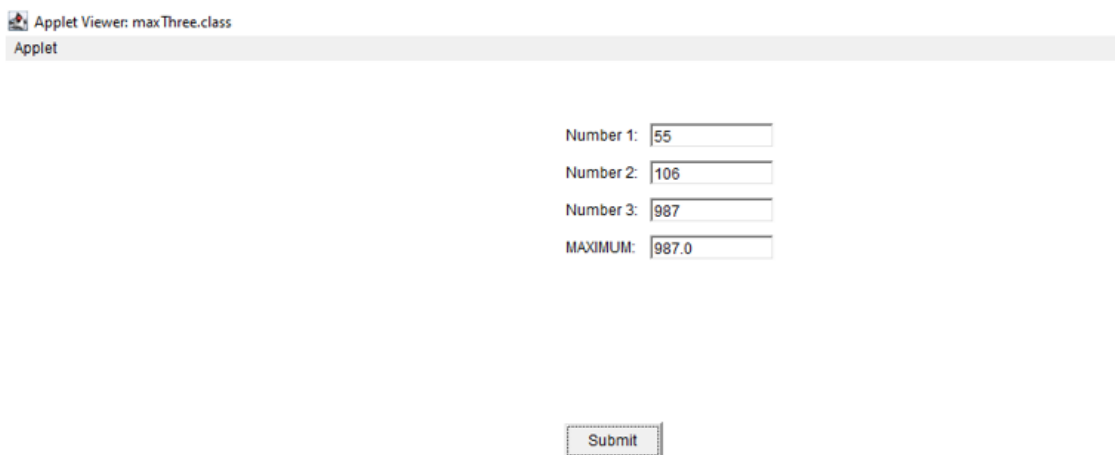
```
<applet code="maxThree.class" width="400" height="400">
```

```
</applet>
```

```
</body>
```

```
</html>
```

Output Screenshot:



Result:

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 35

Date: 22-06-2023

Aim:

Find the percentage of marks obtained by a student in 3 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.

CO5:

Design applications using files and network concepts

Procedure:

markface.java

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class markface extends Applet implements ActionListener
{
    Label label1,label2,label3;
    TextField t1,t2,t3;
    Button btn;
    public void init()
    {
        label1=new Label("Mark1:");
        add(label1);
        t1=new TextField(20);
        add(t1);
        label2=new Label("Mark2:");
        add(label2);
        t2=new TextField(20);
        add(t2);
        label3=new Label("Mark3:");
        add(label3);
```

```
t3=new TextField(20);
add(t3);
btn=new Button("Enter");
add(btn);
btn.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==btn)
{
int mark1,mark2,mark3,total,percent;
mark1= Integer.parseInt(t1.getText());
mark2= Integer.parseInt(t2.getText());
mark3= Integer.parseInt(t3.getText());
total=mark1+mark2+mark3;
percent=((total*100)/300);
t3.setText(""+percent);
Graphics g=getGraphics();
int percentage= Integer.parseInt(t3.getText());
if(percentagem>50)
{
g.drawOval(80,70,150,150);
g.fillOval(120,120,15,15);
g.fillOval(170,120,15,15);
g.drawArc(130,150,50,50,180,180);
}
else
{
g.drawOval(80,70,150,150);
g.fillOval(120,120,15,15);
g.fillOval(170,120,15,15);
```

```
g.drawArc(120,180,60,70,0,180);  
}  
}  
}  
}
```

markfa.html

```
<html>  
<body>  
<applet code="markface.class" width="300" height="300">  
</applet>  
</body>  
</html>
```

Output Screenshot:

```
D:\java\bin>javac markface.java  
D:\java\bin>appletviewer markfa.html
```



Result:

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 36

Date: 22-06-2023

Aim:

Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

CO5:

Design applications using files and network concepts

Procedure:**HouseApplet.java**

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class HouseApplet extends Applet implements MouseListener
{
    int a,b;

    public void init()
    {
        addMouseListener( this);
    }

    public void paint(Graphics g)
    {
        int x[]={ 150,300,225};
        int y[]={ 150,150,25};
        g.drawPolygon(x,y,3);

        g.setColor(Color.green);
        g.fillPolygon(x,y,3);
        g.drawRect(150,150,150,200);//House
```

```
g.setColor(Color.gray);
g.fillRect(150,150,150,200);
g.drawRect(200,200,50,150);//Door
g.setColor(Color.cyan);
g.fillRect(200,200,50,150);
if(a>200 && a<300 && b>200 && b<300)
{
g.setColor(Color.blue);
g.fillRect(200, 200, 50, 150);
}
}
public void mouseClicked(MouseEvent e)
{
}
public void mouseEntered(MouseEvent e)
{
}
@Override
public void mouseExited(MouseEvent e) {
}
public void mousePressed(MouseEvent e)
{
a=e.getX();
b=e.getY();
repaint();
}
public void mouseReleased(MouseEvent e)

{
}
}
```

index.html

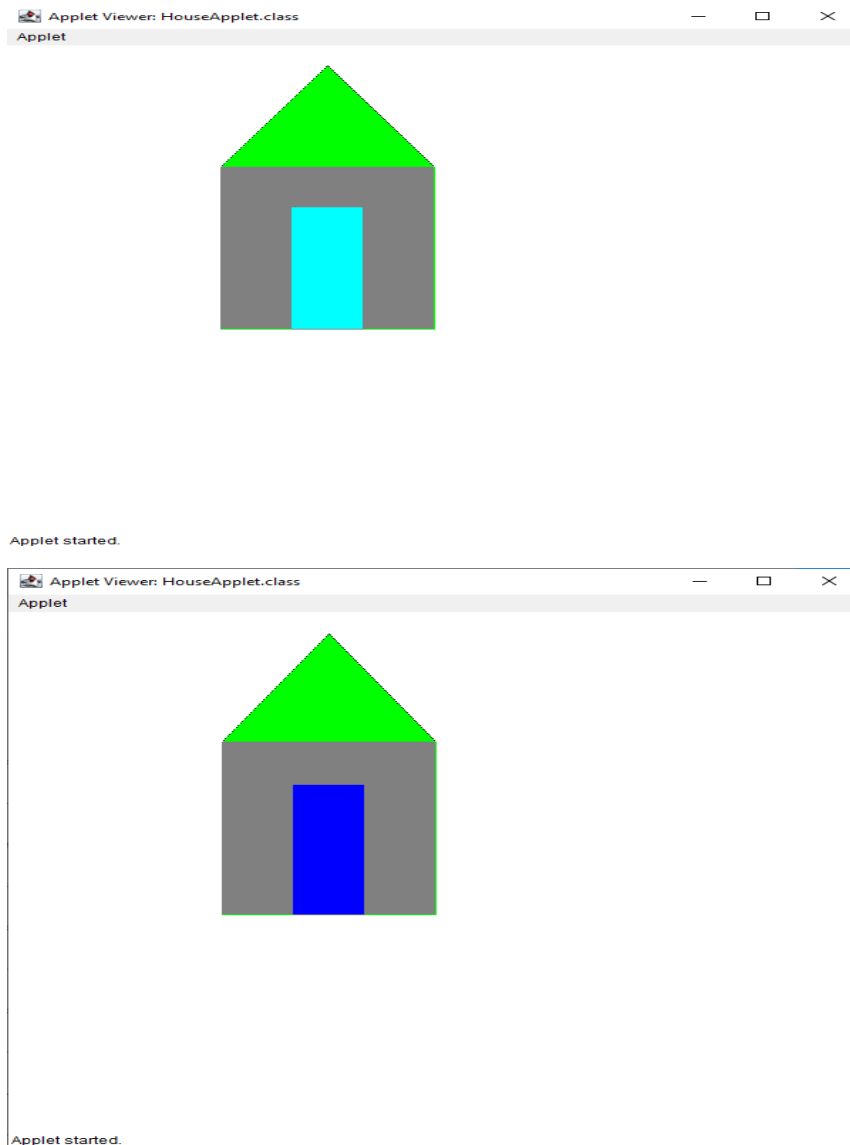
```
<html>
```

```
<body>
```

```
<applet code="HouseApplet.class" width="600" height="600"></applet>
```

```
</body>
```

```
</html>
```

Output Screenshot:**Result:**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 37

Date: 27-06-2023

Aim:

Develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user's choice.

CO5:

Design applications using files and network concepts

Procedure:**chshape.java**

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class chshape extends Applet implements ItemListener

{
    Choice ch=new Choice();
    int n;
    public void init()
    {

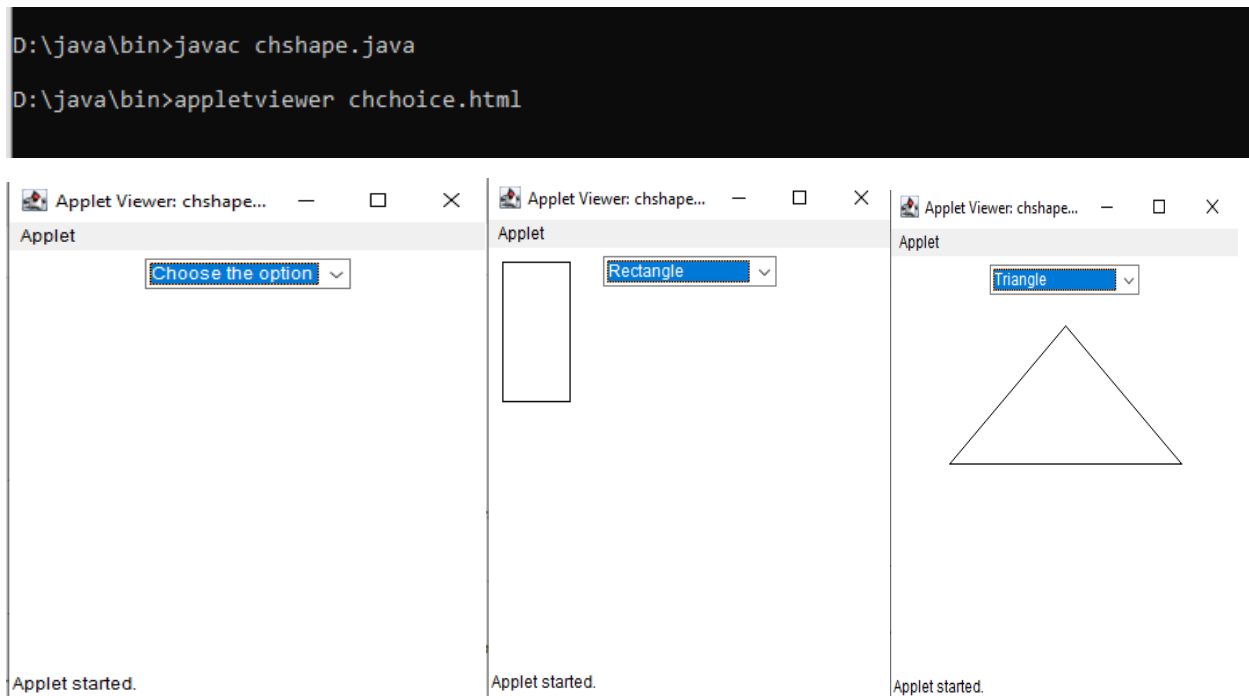
        ch.addItem("Choose the option");
        ch.addItem("Rectangle");
        ch.addItem("Triangle");
        ch.addItem("Square");
        ch.addItem("Circle");
        ch.addItemListener(this);
        add(ch);
    }

    public void paint(Graphics g)
```

```
{
if(n==0)
{
}
if(n==1)
{
g.drawRect(10,10,50,100);
}
if(n==2)
{
int[] xPoints={ 150,50,250};
int[] zPoints={ 50,150,150};
g.drawPolygon(xPoints,zPoints,3);
}
if(n==3)
{
g.drawRect(100,100,50,50);
}
if(n==4)
{
g.drawOval(80,70,150,150);
}
}
public void itemStateChanged(ItemEvent e)
{
n=ch.getSelectedIndex();
repaint();
}
}
```

choice.html

```
<html>
<body>
<applet code="chshape.class" width="300" height="300">
</applet>
</body>
</html>
```

Output Screenshot:**Result:**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 38

Date: 04-07-2023

Aim:

Develop a program to handle all mouse events and window events.

CO5:

Design applications using files and network concepts

Procedure:

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

public class MEvt_WEvt extends Frame implements MouseMotionListener, MouseListener
{
    static Label label1, label2, label3, label4, label5;

    public MEvt_WEvt()
    {
        setTitle("All Mouse Events and Window Events");
        setSize(900, 300);
        setLayout(new FlowLayout());
        addMouseMotionListener(this);
        addMouseListener(this);

        Panel p = new Panel();
        Panel p1 = new Panel();

        Label l1 = new Label("MouseMotionListener events:");
        Label l2 = new Label("MouseListener events:");

        label1 = new Label("no event");
        label2 = new Label("no event");
        label3 = new Label("no event");
        label4 = new Label("no event");
        label5 = new Label("no event");

        p.add(l1);
        p.add(label1);
        p.add(label2);
        p1.add(l2);
```

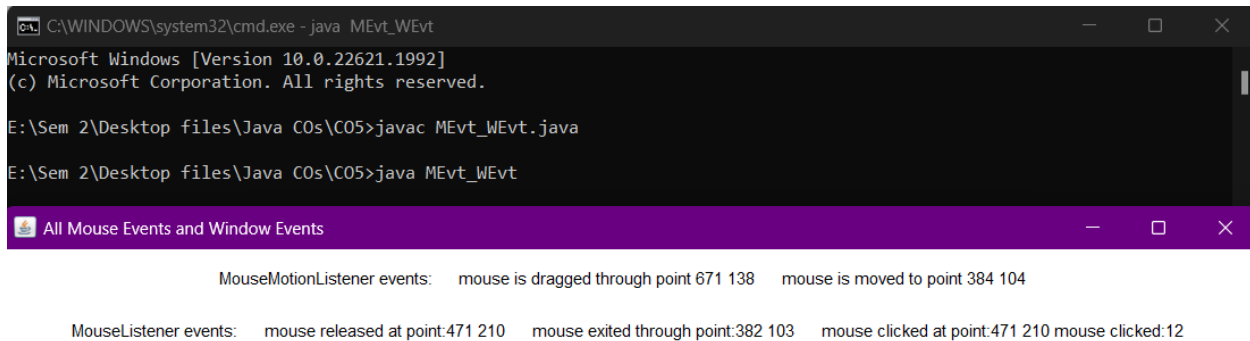
```
p1.add(label3);
p1.add(label4);
p1.add(label5);
add(p);
add(p1);
addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
});
setVisible(true);
}

public static void main(String[] args)
{
    new MEvt_WEvt();
}
public void mouseDragged(MouseEvent e)
{
    label1.setText("mouse is dragged through point " + e.getX() + " " + e.getY());
}
public void mouseMoved(MouseEvent e)
{
    label2.setText("mouse is moved to point " + e.getX() + " " + e.getY());
}
public void mousePressed(MouseEvent e)
{
    label3.setText("mouse pressed at point:" + e.getX() + " " + e.getY());
}
public void mouseReleased(MouseEvent e)
{
    label3.setText("mouse released at point:" + e.getX() + " " + e.getY());
}
public void mouseExited(MouseEvent e)
{
    label4.setText("mouse exited through point:" + e.getX() + " " + e.getY());
}
public void mouseEntered(MouseEvent e)
{
    label4.setText("mouse entered at point:" + e.getX() + " " + e.getY());
}
public void mouseClicked(MouseEvent e)
{

```

```
label5.setText("mouse clicked at point:" + e.getX() + " " + e.getY() + " mouse clicked:" +  
e.getClickCount());  
}  
}
```

Output Screenshot:



Result:

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 39

Date: 20-07-2023

Aim:

Write a program to write to a file, then read from the file and display the contents on the console.

CO6:

Design applications using files and networking concepts.

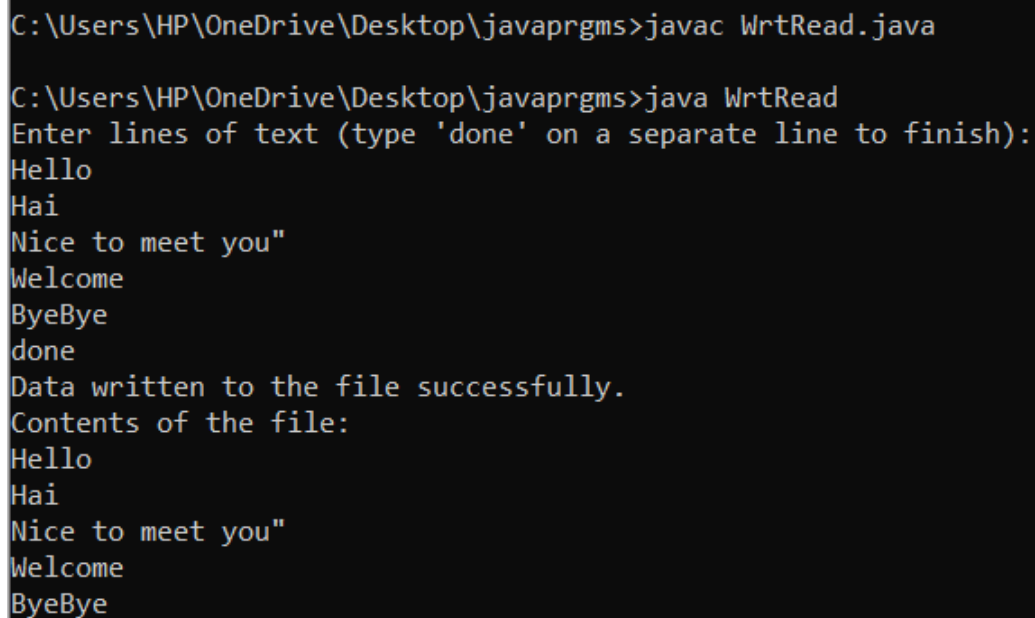
Procedure:

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

public class WrtRead
{
    public static void main(String[] args)
    {
        try
        {
            FileWriter writer = new FileWriter("output.txt");
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter lines of text (type 'done' on a separate line to finish):");
            String inputLine;
            while (!(inputLine = scanner.nextLine()).equalsIgnoreCase("done"))
            {
                writer.write(inputLine + "\n");
            }
            writer.close();
            System.out.println("Data written to the file successfully.");
            FileReader reader = new FileReader("output.txt");
            BufferedReader bufferedReader = new BufferedReader(reader);
            String line;
```

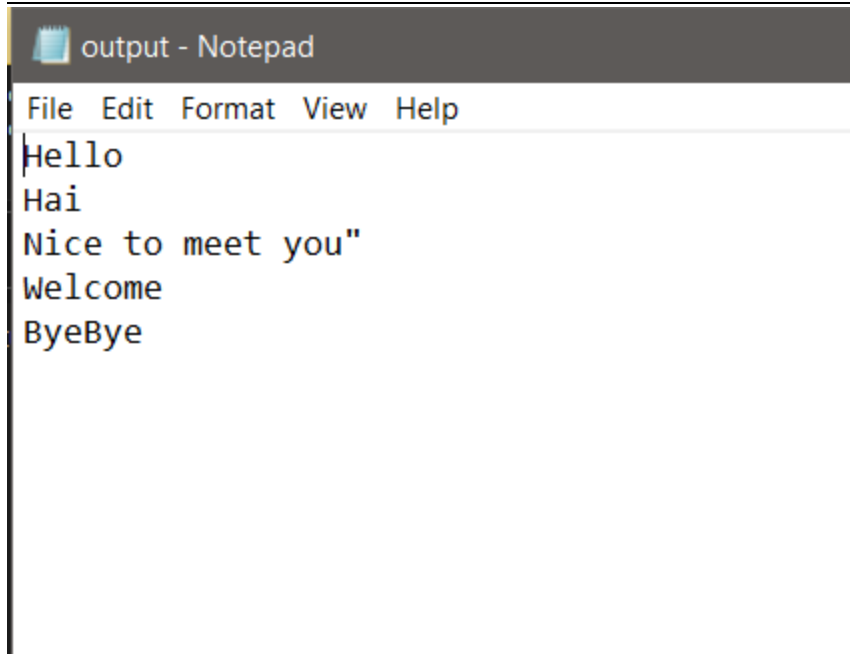
```
System.out.println("Contents of the file:");
while ((line = bufferedReader.readLine()) != null)
{
    System.out.println(line);
}
bufferedReader.close();
scanner.close();
}
catch (IOException e)
{
    e.printStackTrace();
}
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac WrtRead.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java WrtRead
Enter lines of text (type 'done' on a separate line to finish):
Hello
Hai
Nice to meet you"
Welcome
ByeBye
done
Data written to the file successfully.
Contents of the file:
Hello
Hai
Nice to meet you"
Welcome
ByeBye
```

```
output - Notepad
File Edit Format View Help
Hello
Hai
Nice to meet you"
Welcome
ByeBye
```

Result:

The program was executed and the result was successfully obtained. Thus CO6 was obtained.

Experiment No.: 40

Date: 20-07-2023

Aim:

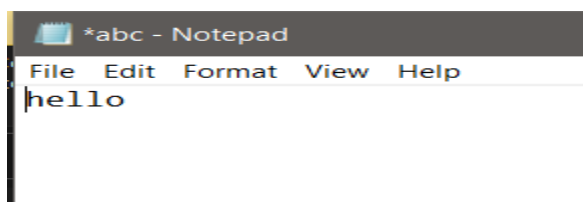
Write a program to copy one file to another.

CO6:

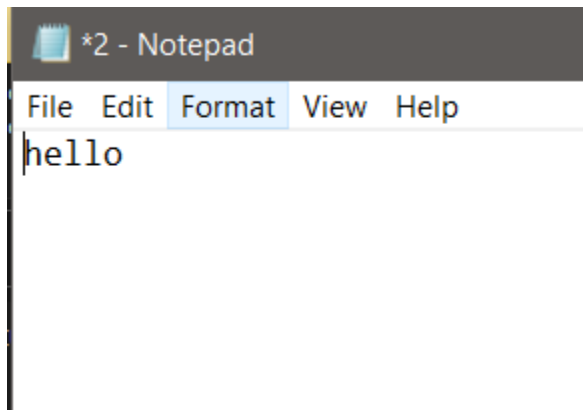
Design applications using files and networking concepts.

Procedure:

```
import java.io.*;
public class Pgm45
{
    public static void main(String[] args) throws Exception {
        {
            FileInputStream fileinput=new FileInputStream("abc.txt") ;
            FileOutputStream fileoutput=new FileOutputStream("2.txt");
            int i;
            while ((i=fileinput.read()) !=-1)
            {
                fileoutput.write(i);
            }
            System.out.println("successfully copied");
        }
    }
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Pgm45.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Pgm45  
successfully copied
```

**Result:**

The program was executed and the result was successfully obtained. Thus CO6 was obtained.

Experiment No.: 41

Date: 22-07-2023

Aim:

Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

CO6:

Design applications using files and networking concepts.

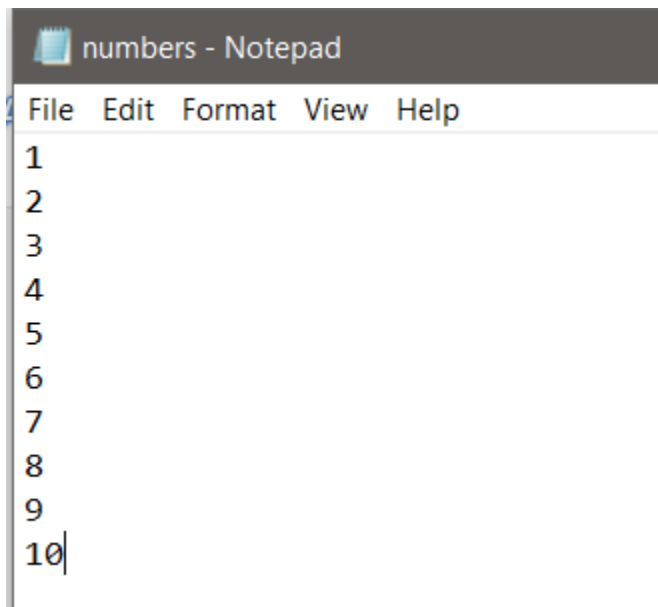
Procedure:

```
import java.io.*;

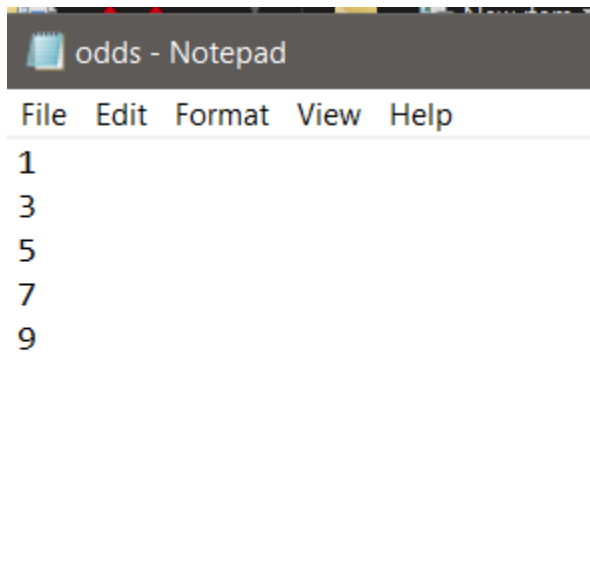
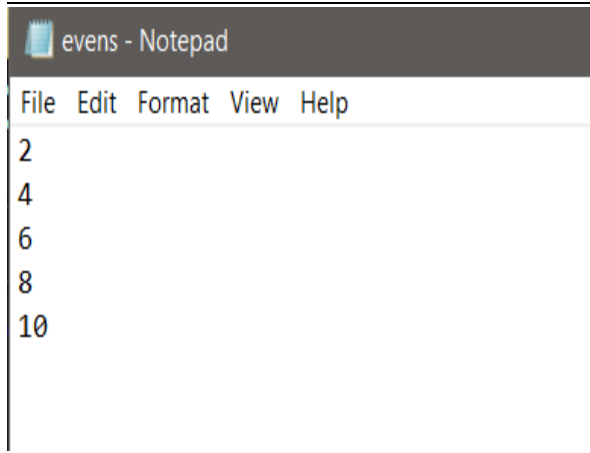
public class OddEven
{
    public static void main(String[] args)
    {
        try
        {
            FileReader reader = new FileReader("numbers.txt");
            BufferedReader bufferedReader = new BufferedReader(reader);
            FileWriter evenWriter = new FileWriter("evens.txt");
            FileWriter oddWriter = new FileWriter("odds.txt");
            String line;
            while ((line = bufferedReader.readLine()) != null)
            {
                int number = Integer.parseInt(line);
                if (number % 2 == 0)
                {
                    evenWriter.write(number + "\n");
                } else
                {
                    oddWriter.write(number + "\n");
                }
            }
        }
    }
}
```

```
}  
bufferedReader.close();  
evenWriter.close();  
oddWriter.close();  
  
System.out.println("Even and odd numbers separated and copied successfully.");  
}  
catch (IOException | NumberFormatException e)  
{  
e.printStackTrace();  
}  
}  
}
```

Output Screenshot:



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac OddEven.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java OddEven  
Even and odd numbers separated and copied successfully.
```

**Result:**

The program was executed and the result was successfully obtained. Thus CO6 was obtained.

Experiment No.: 42

Date: 27-07-2023

Aim:

Client Server communication using DatagramSocket – UDP

CO6:

Design applications using files and networking concepts.

Procedure:**Client.java**

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class Client
{
    public static void main(String[] args)
    {
        try
        {
            DatagramSocket socket = new DatagramSocket();
            InetAddress serverAddress = InetAddress.getByName("localhost");
            int serverPort = 9876;
            Scanner scanner = new Scanner(System.in);
            while (true)
            {
                byte[] sendData;
                System.out.print("Client: ");
                String clientMessage = scanner.nextLine();
                sendData = clientMessage.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);
                socket.send(sendPacket);

                if (clientMessage.equals("done"))
                {
                    System.out.println("Client is exiting...");
                    break;
                }
            }
        }
    }
}
```

```
byte[] receiveData = new byte[1024];
DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
socket.receive(receivePacket);
    String serverMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());
    System.out.println("Server: " + serverMessage);

    if (serverMessage.equals("done"))
    {
        System.out.println("Server is exiting...");
        break;
    }
}
socket.close();
scanner.close();
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```

Server.java

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
public class Server
{
    public static void main(String[] args)
    {
        try
        {
            DatagramSocket socket = new DatagramSocket(9876);
            byte[] receiveData = new byte[1024];
            byte[] sendData;
            while (true)
            {
```



```
DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
socket.receive(receivePacket);
String clientMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());
System.out.println("Client: " + clientMessage);
if (clientMessage.equals("done"))
{
    System.out.println("Server is exiting...");
    break;
}
System.out.print("Server: ");
String serverMessage = System.console().readLine();
sendData = serverMessage.getBytes();
DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
receivePacket.getAddress(), receivePacket.getPort());
socket.send(sendPacket);
if (serverMessage.equals("done"))
{
    System.out.println("Server is exiting...");
    break;
}
}
socket.close();
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```

Output Screenshot:

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Server.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Server  
Client: Hello  
Server: Hai  
Client: Hai  
Server: done  
Server is exiting...
```

```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Client.java  
  
C:\Users\HP\OneDrive\Desktop\javaprgms>java Client  
Client: Hello  
Server: Hai  
Client: Hai  
Server: done  
Server is exiting...
```

Result:

The program was executed and the result was successfully obtained. Thus CO6 was obtained.