
Experiment: 01**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

```
import java.util.Scanner;

public class Product {

    int pcode;

    String pname;

    int price;

    public void ask() {

        Scanner cin = new Scanner(System.in);

        System.out.println("Enter product code: ");

        pcode = cin.nextInt();

        System.out.println("Enter product name: ");

        pname = cin.next();

        System.out.println("Enter product price: ");

        price = cin.nextInt();

    }

}
```

```
public void printProductId() {  
    System.out.println("Product code is: " + pcode);  
    System.out.println("Product name is: " + pname);  
    System.out.println("Product price is: " + price);  
}  
  
public static void main(String[] args) {  
    Product p1 = new Product();  
    Product p2 = new Product();  
    Product p3 = new Product();  
    p1.ask();  
    p2.ask();  
    p3.ask();  
    p1.printProductId();  
    p2.printProductId();  
    p3.printProductId();  
    if(p1.price<p2.price && p1.price <p3.price)  
    {  
        System.out.println(p1.price+"is lowest price");  
    }  
    else if(p2.price<p1.price && p2.price<p3.price)  
    {  
        System.out.println(p2.price+"is lowest price");  
    }  
}
```

```
    else{  
        System.out.println(p3.price+"is lowest price");  
    } }}
```

Output

```
C:\Users\nehaa\Downloads>java Product.java  
Enter product code:  
222  
Enter product name:  
pen  
Enter product price:  
2  
Enter product code:  
22  
Enter product name:  
pencil  
Enter product price:  
2  
Enter product code:  
43  
Enter product name:  
scale  
Enter product price:  
10  
Product code is: 222  
Product name is: pen  
Product price is: 2  
Product code is: 22  
Product name is: pencil  
Product price is: 2  
Product code is: 43  
Product name is: scale  
Product price is: 10  
10is lowest price
```

Result

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

Experiment: 02**Aim:**

Read 2 matrices from the console and perform matrix addition.

CO 1:

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

Procedure

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

```
public class addmatrix{
```

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

```
    {
```

```
        int val,n,i,j,r,c;
```

```
        int arr[][]=new int[50][50];
```

```
        int arr1[][]=new int[50][50];
```

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

```
        System.out.println("enter the size of row :");
```

```
        r=obj.nextInt();
```

```
        System.out.println(" enter the size of the column :");
```

```
        c=obj.nextInt();
```

```
        System.out.println("enter the elments of first array:");
```

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

```
        {
```

```
            for(j=0;j<c;j++)
```

```
        {  
            arr[i][j]=obj.nextInt();  
        }  
    }  
    System.out.println("the first array elements are :");  
    for(i=0;i<r;i++)  
    {  
        for(j=0;j<c;j++)  
        {  
            System.out.print(arr[i][j]+" ");  
        }  
        System.out.println();  
    }  
    System.out.println("enter the elements of second array :");  
    for(i=0;i<r;i++)  
    {  
        for(j=0;j<c;j++)  
        {  
            arr1[i][j]=obj.nextInt();  
        }  
    }  
    System.out.println("the second array elements are :");  
    for(i=0;i<r;i++)  
    {  
        for(j=0;j<c;j++)
```

```
        {  
            System.out.print(arr1[i][j]+" ");  
        }  
        System.out.println();  
    }  
    System.out.println("the sum of array elements is :");  
    for(i=0;i<r;i++)  
    {  
        for(j=0;j<c;j++)  
        {  
            System.out.print(arr[i][j]+arr1[i][j]+" ");  
        }  
        System.out.println();  
    }  
}
```

Output

```
C:\Users\nehaa\Downloads>javac addmatrix.java
C:\Users\nehaa\Downloads>java addmatrix
enter the size of row :
2
enter the size of the column :
2
enter the elements of first array:
2
1
2
3
the first array elements are :
2 1
2 3
enter the elements of second array :
2
1
2
1
the second array elements are :
2 1
2 1
the sum of array elements is :
4 2
4 4
```

Result

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

Experiment: 03**Aim :**

Add complex numbers

CO 1:

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

Procedure

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

```
public class complex
```

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

```
    {    Scanner scan=new Scanner(System.in);
```

```
        System.out.println("Enter the real part of the first number : ");
```

```
        int real1=scan.nextInt();
```

```
        System.out.println("enter the imaginary part of the first number : ");
```

```
        int img1=scan.nextInt();
```

```
        System.out.println("Enter the real part of the second number : ");
```

```
        int real2=scan.nextInt();
```

```
        System.out.println("enter the imaginary part of the second number : ");
```

```
        int img2=scan.nextInt();
```

```
        System.out.println("The sum of first complex number is"+real1+"+"+img1+"i");
```

```
        System.out.println("The sum of second complex number is"+real2+"+"+img2+"i");
```

```
    }}
```


Output

```
C:\MCA NOTES\S2 notes\OOP Lab\java C01>javac complex.java

C:\MCA NOTES\S2 notes\OOP Lab\java C01>java complex
Enter the real part of the first number :
1
enter the imaginary part of the first number :
2
Enter the real part of the second number :
3
enter the imaginary part of the second number :
2
The sum of first complex number is1+2i
The sum of second complex number is3+2i
```

Result

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

Experiment : 04**Aim :**

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

CO 1:

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

Procedure

```
import java.util.Scanner;
```

```
public class matrixsymmetry
```

```
{
```

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

```
    {
```

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

```
        System.out.println("Enter the row and col 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 elements to first array");
```

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

```
        {
```

```
            for(j=0;j<c1;j++)
```

```
        {  
            a[i][j]=obj.nextInt();  
        }  
  
    System.out.println("Printing 2d first Matrix");  
    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

```
C:\MCA NOTES\S2 notes\OOP Lab\java C01>javac matrixsymmetry.java  
C:\MCA NOTES\S2 notes\OOP Lab\java C01>java matrixsymmetry  
Enter the row and col of first matrix  
3  
3  
enter elements to first array  
2  
2  
2  
2  
2  
2  
2  
2  
2  
2  
2  
Printing 2d first Matrix  
2 2 2  
2 2 2  
2 2 2  
The matrix is a Symmetric Matrix
```

Result

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

Experiment : 05**Aim :**

Program to Sort strings

CO 2:

Implement arrays and strings

Procedure

```
import java.util.Scanner;

public class sortstring {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the size of the Array :");

        int l = sc.nextInt();

        String str[] = new String[l];

        int i;

        System.out.println("Enter the String Elements");

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

            str[i] = sc.next();

        }

        String temp;

        int j;

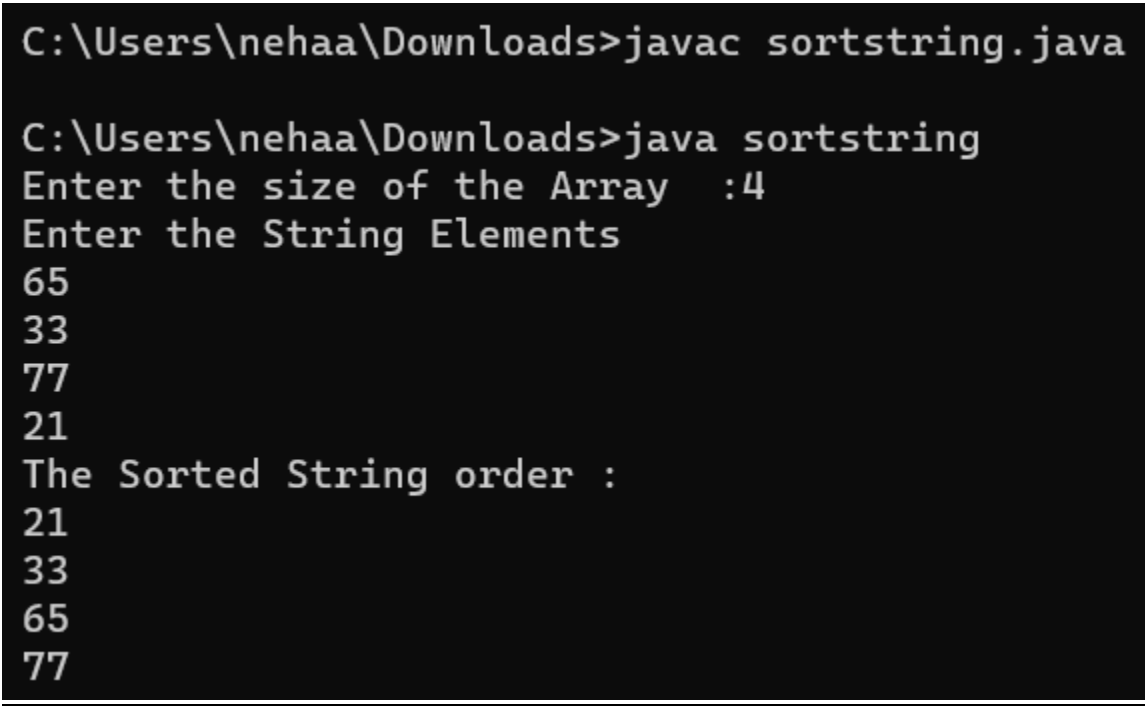
        for (i = 0; i < str.length; i++) {

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

Output



```
C:\Users\nehaa\Downloads>javac sortstring.java  
  
C:\Users\nehaa\Downloads>java sortstring  
Enter the size of the Array :4  
Enter the String Elements  
65  
33  
77  
21  
The Sorted String order :  
21  
33  
65  
77
```

Result

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

Experiment : 06**Aim :**

Search an element in an array.

CO 2:

Implement arrays and strings

Procedure

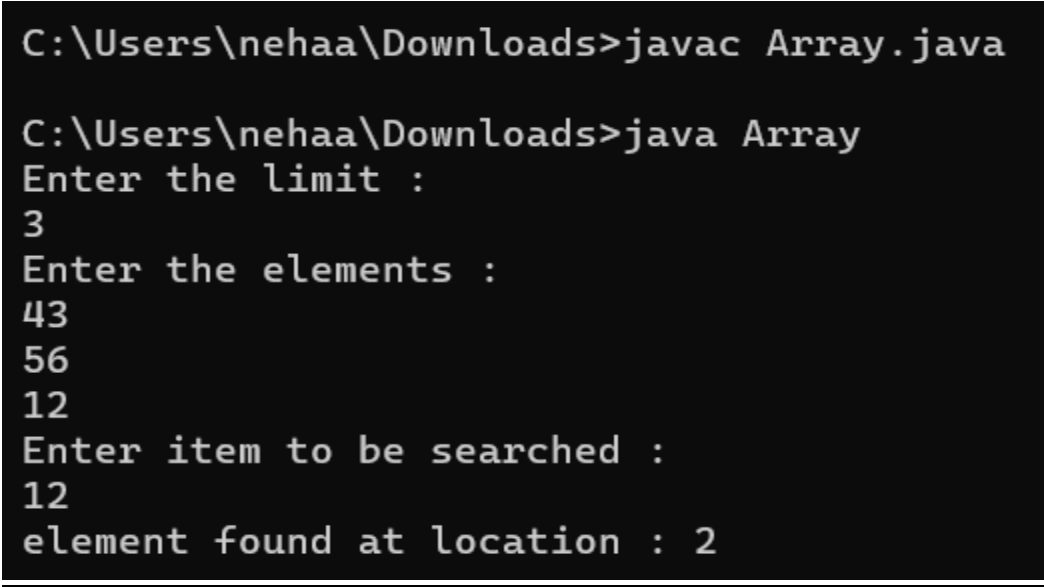
```
import java.util.Scanner;

public class Array
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the limit :");
        Integer size=sc.nextInt();
        int arr[]=new int[size];
        int i,flag=0;
        System.out.println("Enter the elements :");
        for(i=0;i<arr.length;i++)
        {
            arr[i]=sc.nextInt();
        }
        System.out.println("Enter item to be searched :");
        Integer item=sc.nextInt();
```

```
        for(i=0;i<arr.length;i++)
        {
            if(item==arr[i])
            {
                System.out.println("element found at location : "+i);

                flag=1;
            } }
        if(flag==0) {
            System.out.println("element not found :");
        }
    }
}
```

Output



```
C:\Users\nehaa\Downloads>javac Array.java

C:\Users\nehaa\Downloads>java Array
Enter the limit :
3
Enter the elements :
43
56
12
Enter item to be searched :
12
element found at location : 2
```

Result

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

Experiment : 07**Aim:**

Perform string manipulations

CO 2:

Implement arrays and strings

Procedure

```
import java.util.*;

public class StringManip{

    public static void main(String[] args){

        Scanner Sr = new Scanner(System.in);

        System.out.println("Enter the first string: ");

        String str1 = Sr.nextLine();

        System.out.println("Enter the second string: ");

        String str2 = Sr.nextLine();

        System.out.println("Concatenate: "+ str1.concat(" "+ str2));

        if(str1.equals(str2)){

            System.out.println("Both string are the same");

        }

        else{

            System.out.println("Both string are not the same");

        }

        System.out.println("Length of the first string: "+ str1.length());

        System.out.println("Length of the second string: "+ str2.length());

    }

}
```

```
        if(str1.length() == str2.length()){
            System.out.println("Both strings are of same length");
        }
        else if(str1.length() > str2.length()){
            System.out.println("The first string is longer than the second");
        }
        else{
            System.out.println("The second string is longer than the first");
        }

        String str3 = str1.concat(" " + str2);

        System.out.println("String in uppercase: " + str3.toUpperCase());
        System.out.println("String in lowercase: " + str3.toLowerCase());

        System.out.println("Enter the number of characters after which the new
substring should be formed: ");

        int ch = Sr.nextInt();

        System.out.println("Substring: " + str3.substring(ch));

        System.out.println("compareTo: " + str1.compareTo(str2));

    }}
```

Output

```
C:\MCA NOTES\S2 notes\OOP Lab\java C02>javac StringManip.java

C:\MCA NOTES\S2 notes\OOP Lab\java C02>java StringManip
Enter the first string:
neha
Enter the second string:
arya
Concatenate: neha arya
Both string are not the same
Length of the first string: 4
Length of the second string: 4
Both strings are of same length
String in uppercase: NEHA ARYA
String in lowercase: neha arya
Enter the number of characters after which the new substring should be formed:
3
Substring: a arya
compareTo: 13
```

Result

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

Experiment : 08**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.

CO 2:

Implement arrays and strings

Procedure

```
import java.util.Scanner;

public class Employee
{
    int eno;

    String ename;

    int sal;

    public void get()
    {
        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");

        sal=sc.nextInt();
    }
}
```

```
}  
  
public void print()  
  
{  
  
    System.out.println("Employee Details");  
  
    System.out.println("Employee number"+eno);  
  
    System.out.println("Employee name"+ename);  
  
    System.out.println("Employee salary"+sal);  
  
}  
  
public static void main(String args[])  
  
{  
  
    Scanner sc=new Scanner(System.in);  
  
    System.out.println("Enter the no of employees");  
  
    int n=sc.nextInt();  
  
    Employee e[]=new Employee[n];  
  
    for( int i=0;i<n;i++)  
  
    {  
  
        e[i]=new Employee();  
  
        e[i].get();  
  
    }  
  
    for( int i=0;i<n;i++)  
  
    {  
  
        e[i].print();  
  
    }  
  
}
```

```
System.out.println("Enter the eno");  
  
int val=sc.nextInt();  
  
for(int i=0;i<n;i++)  
{  
    if(e[i].eno==val)  
    {  
        e[i].print();  
        break;  
    }  
}}}
```

Output

```
C:\MCA NOTES\S2 notes\OOP Lab\java C02>javac Employee.java

C:\MCA NOTES\S2 notes\OOP Lab\java C02>java Employee
Enter the no of employees
3
enter the employee number
1
enter the employee name
neha
enter the employee salary
3000
enter the employee number
2
enter the employee name
jomol
enter the employee salary
2000
enter the employee number
3
enter the employee name
arya
enter the employee salary
4000
Employee Details
Employee number1
Employee nameneha
Employee salary3000
Employee Details
Employee number2
Employee namejomol
Employee salary2000
Employee Details
Employee number3
Employee namearya
Employee salary4000
```

Result

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

Experiment : 09**Aim :**

Area of different shapes using overloaded functions

CO 3:

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.Scanner;

class areaShapes{

void area(int a){

System.out.println("area of square is "+a*a);

}

void area(int a, int b){

System.out.println("area of rectangle "+a*b);

}

void area(int length, int breadth, int height){

System.out.println("Area of
Cuboid"+(2*(length*breadth)+2*(length*height)+2*(height*breadth)));

}

}

public class Area {

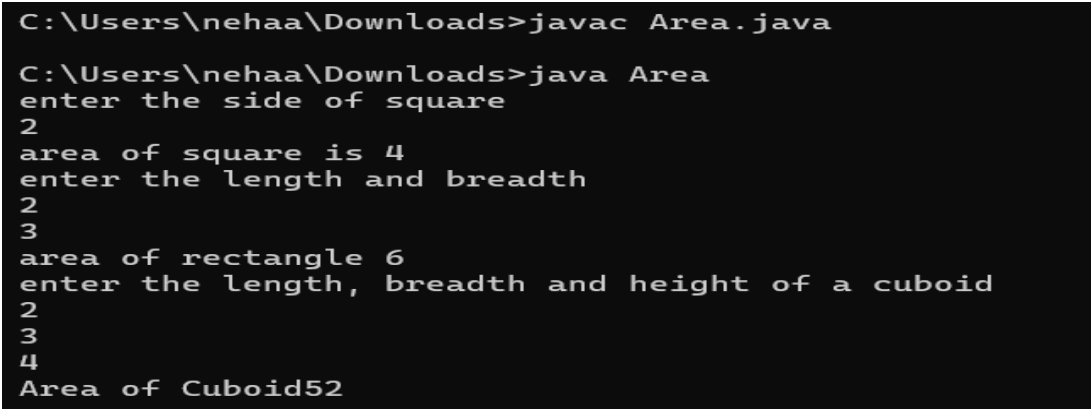
public static void main(String[] args) {

int a,b,c;

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

```
areaShapes obj=new areaShapes();  
  
System.out.println("enter the side of square");  
  
a= s.nextInt();  
  
obj.area(a);  
  
System.out.println("enter the length and breadth");  
  
a=s.nextInt();  
  
b=s.nextInt();  
  
obj.area(a,b);  
  
System.out.println("enter the length, breadth and height of a cuboid");  
  
a=s.nextInt();  
  
b=s.nextInt();  
  
c=s.nextInt();  
  
obj.area(a,b,c); } }
```

Output



```
C:\Users\nehaa\Downloads>javac Area.java  
  
C:\Users\nehaa\Downloads>java Area  
enter the side of square  
2  
area of square is 4  
enter the length and breadth  
2  
3  
area of rectangle 6  
enter the length, breadth and height of a cuboid  
2  
3  
4  
Area of Cuboid52
```

Result

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

Experiment :10**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.

CO 3:

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.*;

class Employee
{
    int empid,salary;
    String empname;
    String add;
    Employee(int id,String name,int sal,String adds)
    {
        empid = id;
        empname = name;
        add = adds;
        salary = sal;
    }
}
```

```
class Teacher extends Employee
{
    String dept;
    String sub_taught;
    Teacher(int id,String name,int sal,String adds,String dept_name,String sub)
    {
        super(id,name,sal,adds);
        dept = dept_name;
        sub_taught = sub;
    }
    public void print()
    {
        System.out.println("empid : "+empid);
        System.out.println("empname : "+empname);
        System.out.println("salary : "+salary);
        System.out.println("address : "+add);
        System.out.println("Department : "+dept);
        System.out.println("Subject Taught : "+sub_taught);
    }
}

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

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

System.out.print("Enter the limit : ");

int n = in.nextInt();

Teacher t[] = new Teacher[n];

System.out.println("Enter the employee details : ");

for(int i=0;i<n;i++)
{
    System.out.print("Enter the emp id : ");

    int id = in.nextInt();

    System.out.print("Enter the emp name : ");

    String name = in.next();

    System.out.print("Enter the salary : ");

    int sal = in.nextInt();

    System.out.print("Enter the address : ");

    String adds = in.next();

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

    String dept_name = in.next();

    System.out.print("Enter the subject taught : ");

    String sub = in.next();

    t[i] = new Teacher(id,name,sal,adds,dept_name,sub);

}

System.out.println("Employee Details : ");

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

```
{  
  
    t[i].print();  
  
}}}
```

Output

```
C:\MCA NOTES\S2 notes\OOP Lab\java C03>javac DetailsET.java  
  
C:\MCA NOTES\S2 notes\OOP Lab\java C03>java DetailsET  
Enter the limit : 2  
Enter the employee details :  
Enter the emp id : 11  
Enter the emp name : neha  
Enter the salary : 2000  
Enter the address : thiruvonam  
Enter the department : mca  
Enter the subject taught : cs  
Enter the emp id : 122  
Enter the emp name : arya  
Enter the salary : 2000  
Enter the address : nayna  
Enter the department : maths  
Enter the subject taught : maths  
Employee Details :  
empid : 11  
empname : neha  
salary : 2000  
address : thiruvonam  
Department : mca  
Subject Taught : cs  
empid : 122  
empname : arya  
salary : 2000  
address : nayna  
Department : maths  
Subject Taught : maths
```

Result

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

Experiment : 11**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.

CO 3:

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

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

```
class Person
```

```
{
```

```
    int age;
```

```
    String Pname,Paddress,Pgender;
```

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

```
    Person()
```

```
    {
```

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

```
        Pname=sc.next();
```

```
        System.out.println("Enter the gender: ");
```

```
        Pgender=sc.next();

        System.out.println("Enter the Age: ");

        age=sc.nextInt();

        System.out.println("Enter the Address: ");

        Paddress=sc.next();

    }

}

class Employee extends Person
{

    int empid,Salary;

    String Cname,Qual;

    Employee()

    {

        System.out.println("Enter the Empid: ");

        empid=sc.nextInt();

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

        Cname=sc.next();

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

        Salary=sc.nextInt();

        System.out.println("Enter the Qualification: ");
```

```
        Qual=sc.next();

    }

}

class Teacher extends Employee
{
    int tid;
    String Dep,Sub;

    Teacher()
    {
        System.out.println("Enter the Teacher ID: ");
        tid=sc.nextInt();
        System.out.println("Enter the Department: ");
        Dep=sc.next();
        System.out.println("Enter the Subject: ");
        Sub=sc.next();
    }

    void Display()
    {
        System.out.println(" Name: "+Pname);
        System.out.println(" gender: "+Pgender);
    }
}
```

```
        System.out.println(" Age: "+age);
        System.out.println(" Address: "+Paddress);
        System.out.println(" Empid: "+empid);
        System.out.println(" Company Name: "+Cname);
        System.out.println(" Salary: "+Salary);
        System.out.println(" Qualification: "+Qual);
        System.out.println(" Teacher ID: "+tid);
        System.out.println(" Department: "+Dep);
        System.out.println(" Subject: "+Sub);
    }

    public static void main(String args[])
    {
        int n,i;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the limit : ");
        n=sc.nextInt();
        Teacher t[]=new Teacher[n];
        for(i=0;i<n;i++)
        {
            System.out.println("Enter the details of Employee "+(i+1));
            t[i]=new Teacher();
        }
        System.out.println("Employee Details");
```

```
        for(i=0;i<n;i++)  
        {  
            t[i].Display();  
        }  
    }  
}
```

Output

```
C:\MCA NOTES\S2 notes\OOP Lab\java C03>javac Teacher.java  
  
C:\MCA NOTES\S2 notes\OOP Lab\java C03>java Teacher  
Enter the limit :  
2  
Enter the details of Employee 1  
Enter the Name:  
jo  
Enter the gender:  
female  
Enter the Age:  
33  
Enter the Address:  
eteu  
Enter the Empid:  
11  
Enter the Company Name:  
hss  
Enter the Salary:  
30000  
Enter the Qualification:  
mca  
Enter the Teacher ID:  
1  
Enter the Department:  
mca  
Enter the Subject:  
ipr  
Enter the details of Employee 2  
Enter the Name:  
hashi  
Enter the gender:  
male  
Enter the Age:  
32  
Enter the Address:  
errww  
Enter the Empid:  
23
```

```
Enter the Empid:
23
Enter the Company Name:
fyu
Enter the Salary:
24000
Enter the Qualification:
acca
Enter the Teacher ID:
2
Enter the Department:
acca
Enter the Subject:
commerce
Employee Details
Name: jo
gender: female
Age: 33
Address: eteu
Empid: 11
Company Name: hss
Salary: 30000
Qualification: mca
Teacher ID: 1
Department: mca
Subject: ipr
Name: hashi
gender: male
Age: 32
Address: errww
Empid: 23
Company Name: fyu
Salary: 24000
Qualification: acca
Teacher ID: 2
Department: acca
Subject: commerce
```

Result

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

Experiment : 12**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.

CO 3:

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.*;

class publisher{

    String pub_name;

    publisher()

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

        System.out.println("Publisher name :");

        pub_name=obj.next();

    }

}

class book extends publisher{

    String book_name;

    book()

    {

        Scanner obj=new Scanner(System.in);

        System.out.println("book name :");

        book_name=obj.next();

    }

}
```

```
    }}  
  
class literature extends book{  
    void display()  
    {  
        System.out.println("Publisher name is :"+pub_name);  
        System.out.println("book name is :"+book_name);  
    }}  
  
class fiction extends book{  
    void display()  
    {  
        System.out.println("Publisher name is :"+pub_name);  
        System.out.println("book name is :"+book_name);  
    }}  
  
class library  
{  
    public static void main(String[] args)  
    {  
        int i=0;  
        Scanner obj=new Scanner(System.in);  
        System.out.println("enter the total number :");  
        int size=obj.nextInt();  
        literature arr1[]= new literature[size];  
        fiction arr2[]=new fiction[size];
```

```
System.out.println("enter the details of literature books :");

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

{

arr1[i]=new literature();

}

System.out.println("enter the details of fiction books :");

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

{

    arr2[i]=new fiction();

}

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

System.out.println("details of literature books :");

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

{

    arr1[i].display();

}

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

System.out.println("details of fiction books :");


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

{

    arr1[i].display();

}}}
```

Output



```
C:\Users\nehaa\Downloads>javac library.java

C:\Users\nehaa\Downloads>java library
enter the total number :
1
enter the details of literature books :
Publisher name :
harry
book name :
prince
enter the details of fiction books :
Publisher name :
stephen
book name :
world

details of literature books :
Publisher name is :harry
book name is :prince

details of fiction books :
Publisher name is :harry
book name is :prince
```

Result

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

Experiment : 13**Aim :**

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

CO3:

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.*;

interface student{

    public void getS();

    public void dispS();

}

interface sports

{

    public void getSp();

    public void dispSp();

}

class result implements student,sports

{

    Scanner obj=new Scanner(System.in);

    String name,spitem;

    int m1,m2,roll,rank;
```

```
double total,percentage;

public void getS()
{
    System.out.println("enter the name :");
    name=obj.next();
    System.out.println("enter the roll number :");
    roll=obj.nextInt();
    System.out.println("enter the mark 1 :");
    m1=obj.nextInt();
    System.out.println("enter the mark2 :");
    m2=obj.nextInt();
    total=m1+m2;
    percentage=(total*200)/100;
}

public void getSp()
{
    System.out.println("enter the sports item which the student participated
in :");

    spitem=obj.next();
    System.out.println("enter the rank of the student :");
    rank=obj.nextInt();
}

public void dispS(){
```

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

        System.out.println("Roll Number of the Student :"+roll);

        System.out.println("Mark 1 the Student :"+m1);

        System.out.println("Mark 2 of the Student :"+m2);

        System.out.println("Total Mark of the Student :"+total);

        System.out.println("Percentage of the Student :"+percentage);

    }

    public void dispSp()

    {

        System.out.println("Sports item :"+spitem);

        System.out.println("The Rank :"+rank);

    }}

    public class multi{

        public static void main(String[] args)

        {

            result obj=new result();

            obj.getS();

            obj.getSp();

            obj.dispS();

            obj.dispSp();

        }}
```

Output

```
C:\Users\nehaa\Downloads>javac multi.java

C:\Users\nehaa\Downloads>java multi
enter the name :
neg
enter the roll number :
1
enter the mark 1 :
12
enter the mark2 :
22
enter the sports item which the student participated in :
2
enter the rank of the student :
23
Name of the Student :neg
Roll Number of the Student :1
Mark 1 the Student :12
Mark 2 of the Student :22
Total Mark of the Student :34.0
Percentage of the Student :68.0
Sports item :2
The Rank :23
```

Result

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

Experiment : 14**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.

CO 3:

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.*;

interface Prototype
{
    void area();
    void peri();
}

class Rectangle implements Prototype
{
    int l,b,a,p;

    public void area()
    {
        Scanner in = new Scanner(System.in);

        System.out.println("Enter the length and breadth of rectangle : ");

        l = in.nextInt();
        b = in.nextInt();

        a=l*b;
```

```
        System.out.println("Area of rectangle : "+a);
    }

    public void peri()
    {
        p=2*(l+b);
        System.out.println("Perimeter of rectangle : "+p);
    }
}

class Circle implements Prototype
{
    float r,a,p;

    public void area()
    {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the radius of circle : ");
        r = in.nextFloat();
        a=3.14f*(r*r);
        System.out.println("Area of circle : "+a);
    }

    public void peri()
    {
        p=(2*3.14f)*r;
        System.out.println("Perimeter of circle : "+p);
    }
}
```

```
    }    }

public class ShapeDime
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        int op=1;
        do
        {
            System.out.println(" Select any of the following shapes to
calculate area and perimeter : ");

            System.out.println(" 1. Circle ");
            System.out.println(" 2. Rectangle ");
            System.out.println(" 3. Exit ");
            int ch=in.nextInt();
            switch(ch)
            {
                case 1 :Circle c = new Circle();
                    c.area();
                    c.peri();
                    break;

                case 2 :Rectangle r = new Rectangle();
                    r.area();
```

```
        r.peri();  
        break;  
    case 3 :System.exit(0);  
        break;  
    default : System.out.println("Invalid choice ");  
    }  
    while(op!=0);} }
```

Output

```
C:\MCA NOTES\S2 notes\OOP Lab\java C03>javac ShapeDime.java  
C:\MCA NOTES\S2 notes\OOP Lab\java C03>java ShapeDime  
Select any of the following shapes to calculate area and perimeter :  
1. Circle  
2. Rectangle  
3. Exit  
1  
Enter the radius of circle :  
4  
Area of circle : 50.24  
Perimeter of circle : 25.12  
Select any of the following shapes to calculate area and perimeter :  
1. Circle  
2. Rectangle  
3. Exit  
2  
Enter the length and breadth of rectangle :  
2  
3  
Area of rectangle : 6  
Perimeter of rectangle : 10  
Select any of the following shapes to calculate area and perimeter :  
1. Circle  
2. Rectangle  
3. Exit  
3
```

Result

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

Experiment : 15**Aim :**

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

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

CO 3:

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.Scanner;

interface calc
{
    void calculate();
}

class bill implements calc
{
    String date,name,p_id;
    int quantity;
    double unit_price,total,namount=0;

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

```
public void getdata()
{
    System.out.println("\nEnter product id:");
    p_id = sc.nextLine();
    System.out.println("Enter product name:");
    name = sc.nextLine();
    System.out.println("Enter the Quantity:");
    quantity = sc.nextInt();
    System.out.println("Enter the unit price:");
    unit_price = sc.nextDouble();
}

public void calculate()
{
    total = quantity * unit_price;
}

public void display()
{
    System.out.println(p_id+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit_price+"\t"+total);
}

}

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

```
{  
int n,i;  
  
double namount=0,t;  
  
int ran;  
  
String date;  
  
t = Math.random() *1000000;  
  
ran = (int) t;  
  
Scanner sc = new Scanner(System.in);  
  
System.out.println("Order no. #" +ran);  
  
System.out.println("Enter the date:");  
  
date = sc.nextLine();  
  
System.out.println("Enter how many products are there:");  
  
n = sc.nextInt();  
  
bill ob[] = new bill[n];  
  
for(i=0;i<n;i++)  
ob[i] = new bill();  
  
for(i=0;i<n;i++){  
ob[i].getdata();  
ob[i].calculate();  
}  
  
System.out.println("Date:" +date);  
  
System.out.println("Product Id Name Quantity unit price Total ");
```

```

for(i=0;i<n;i++){
    ob[i].display();
    namount += ob[i].total;}

System.out.println("\t\t\tNet.Amount\t\t"+namount);

}}

```

Output

```

C:\MCA NOTES\S2 notes\OOP Lab\java CO3>java bills
Order no. #946088
Enter the date:
22/2/2000
Enter how many products are there:
3

Enter product id:
11
Enter product name:
pencil
Enter the Quantity:
2
Enter the unit price:
2

Enter product id:
122
Enter product name:
pen
Enter the Quantity:
2
Enter the unit price:
2

Enter product id:
22
Enter product name:
scale
Enter the Quantity:
2
Enter the unit price:
10
Date:22/2/2000

```

Product Id	Name	Quantity	unit price	Total
11	pencil	2	2.0	4.0
122	pen	2	2.0	4.0
22	scale	2	10.0	20.0
Net.Amount				28.0

Result

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

Experiment :16**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.

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
package graphics;
import java.util.*;
interface shapes{
    public double RecArea();
    public double CircArea();
    public double SquareArea();
    public double TriangArea();
}
public class Graphics implements shapes {
    Scanner obj = new Scanner(System.in);
    int r,l,b,s;
    double pi = 3.14, area;
    public double RecArea(){
        System.out.print("Enter the Length of Rectangle: ");
        l=obj.nextInt();
        System.out.print("Enter the Breadth of Rectangle: ");
        b=obj.nextInt();
        area=l*b;
        return area;
    }
    public double CircArea(){
        System.out.print("Enter the Radius of Circle: ");
        r =obj.nextInt();
        area = pi * r * r;
        return area;
    }
}
```

```
public double SquareArea(){
    System.out.print("Enter the Side of the Square: ");
    s = obj.nextInt();
    area = s * s;
    return area;
}
public double TriangArea(){
    System.out.print("Enter the Width of the Triangle: ");
    double base = obj.nextDouble();
    System.out.print("Enter the Height of the Triangle: ");
    double height = obj.nextDouble();
    double area = (base* height)/2;
    return area;
}
}

import graphics.Graphics;
import java.util.*;
public class Areas{
    public static void main(String []args){
        Scanner sc = new Scanner(System.in);
        Graphics Obj = new Graphics();
        int choice = 0;
        while(choice != 5){
            System.out.println("-----AREAS OF SHAPES-----\n1. Rectangle\n2.
Circle\n3. Square\n4. Triangle\n5. Exit");

            System.out.print("Enter your choice: ");
            choice = sc.nextInt();
            switch(choice){
                case 1:
                    System.out.println("Area of Rectangle: " + Obj.RecArea());
                    break;
                case 2:
                    System.out.println("Area of Circle: " + Obj.CircArea());
```

```
        break;
    case 3:
        System.out.println("Area of Square: " + Obj.SquareArea());
        break;
    case 4:
        System.out.println("Area of Triangle: " + Obj.TriangArea());
        break;
    case 5:
        System.exit(0);
        break;
    default:
        System.out.println("Select a valid option!");
    }
}
```

Output

```
C:\MCA NOTES_Neha\S2 notes\OOP Lab\java C04>javac -d . Area.java
C:\MCA NOTES_Neha\S2 notes\OOP Lab\java C04>javac -d . shapes.java
C:\MCA NOTES_Neha\S2 notes\OOP Lab\java C04>java Graphics.shapes
Enter the length of the rectangle
2
Enter the breadth of the rectangle
3
Area of the rectangle is 6.0
Enter the height of the triangle
2
Enter the base of the triangle
2
Area of the triangle is 2.0
Enter the side of the square
2
Area of the square:4.0
Enter the radius of the circle
2
Area of the Square is:12.56
```

Result

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

Experiment : 17**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

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

package Arithmetic;

interface operations

{

 public void input();

 public void addition();

 public void subtract();

 public void multiply();

 public void division();

}

package Arithmetic;

import java.util.*;

public class basic implements operations

{

 double a,b,add,diff,mul,div;

 public void input()

 {

 Scanner sc=new Scanner(System.in);

 System.out.println("Enter two numbers");

 a=sc.nextInt();

 b=sc.nextInt();

 }

 public void addition()

 {

 add=a+b;

```
        System.out.println("Sum is:"+add);

    }
    public void subtract()
    {
        diff=a-b;
        System.out.println("Difference is:"+diff);
    }
    public void multiply()
    {
        mul=a*b;
        System.out.println("Multiplication is:"+mul);
    }
    public void division()
    {
        div=a/b;
        System.out.println("Division is:"+div);
    }
    public static void main(String args[])
    {
        basic b=new basic();
        b.input();
        b.addition();
        b.subtract();
        b.multiply();
        b.division();
    }
}
```

Output

```
C:\MCA NOTES_Neha\S2 notes\OOP Lab\java C04>javac basic.java
C:\MCA NOTES_Neha\S2 notes\OOP Lab\java C04>javac operations.java
C:\MCA NOTES_Neha\S2 notes\OOP Lab\java C04>java Arithmetic.basic
Enter two numbers
4
2
Sum is:6.0
Difference is:2.0
Multiplication is:8.0
Division is:2.0
```

Result

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

Experiment : 18**Aim :**

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

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

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

```
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();
```

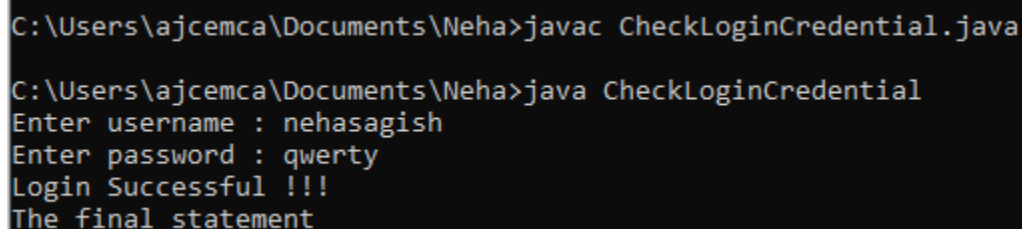
```
        System.out.print("Enter password : ");
```

```
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("qwerty"))
        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 final statement");
}}}
```

Output



```
C:\Users\ajcemca\Documents\Neha>javac CheckLoginCredential.java
C:\Users\ajcemca\Documents\Neha>java CheckLoginCredential
Enter username : nehasagish
Enter password : qwerty
Login Successful !!!
The final statement
```

Result

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

Experiment : 19**Aim :**

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

CO 4:

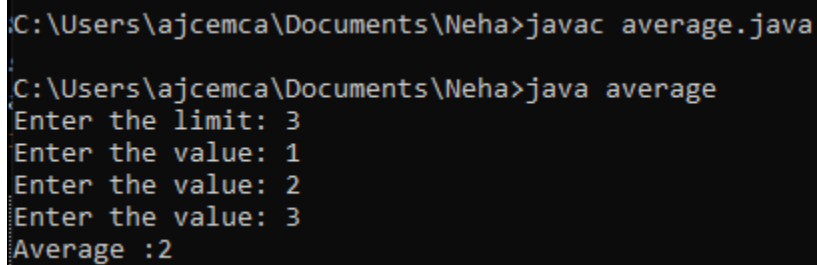
Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.Scanner;
class nIntExcep extends Exception{
    public nIntExcep(String str){
        super(str);
    }}public class average{
    public static void main(String[] args){
        Scanner Snr=new Scanner(System.in);
        int arr[];
        int sz, total=0, avg, count=0;
        System.out.print("Enter the limit: ");
        sz = Snr.nextInt();
        arr = new int[sz];
        for(int i=0;i<sz;i++)
        {
            System.out.print("Enter the value: ");
            int val = Snr.nextInt();
            arr[i] = val;
        }
        try {
            for(int i=0;i<sz;i++) {
                if(arr[i]<0){
                    throw new nIntExcep("Numbers must be positive");
                }
                else{
                    total += arr[i];
                }
            }
        }
    }
}
```

```
        count++;  
    } }  
    avg=total/count;  
    System.out.println("Average :"+avg);  
}  
catch(nIntExcep e){  
    System.out.println(":: An Exception Occurred :: "+ e);  
} } }
```

Output



```
C:\Users\ajcemca\Documents\Neha>javac average.java  
C:\Users\ajcemca\Documents\Neha>java average  
Enter the limit: 3  
Enter the value: 1  
Enter the value: 2  
Enter the value: 3  
Average :2
```

Result

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

Experiment : 20**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. Use java.util package and Collection framework

Procedure

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

```
class MulTable extends Thread
```

```
{
    public void run()
    {
        int num=5;
        System.out.println("THE MULTIPLICATION TABLE::");
        for(int i=0;i<=10;i++)
        {
            System.out.printf("%d * %d=%d\n",num,i,num*i);
        }
    }
}
```

```
class PrimeNo extends Thread
```

```
{
    public void run()
    {
        int i,j,flag;
        Scanner sc=new Scanner(System.in);
        System.out.println("first N prime numbers");
        System.out.println("enter the limit :");
        int N=sc.nextInt();
        System.out.println("the prime numbers between 1 and "+N+"are");
        for(i=1;i<=N;i++)
        {
            if(i==1 || i==0)
                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 PrimeThread
{
    public static void main(String[] args) throws InterruptedException
    {
        MulTable a=new MulTable();
        a.start();
        a.sleep(2000);
        PrimeNo b=new PrimeNo();
        b.start();
        b.sleep(200);
    }
}
```

Output

```
C:\Users\ajcemca\Documents\Neha>javac PrimeThread.java

C:\Users\ajcemca\Documents\Neha>java PrimeThread
THE MULTIPLICATION TABLE::
5 * 0=0
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
first N prime numbers
enter the limit :
2
the prime numbers between 1 and 2 are
2
```

Result

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

Experiment : 21**Aim :**

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

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;
class fibonacci implements Runnable
{
    public void run()
    {
        int first=0,second=1,next;
        Scanner sc=new Scanner(System.in);
        System.out.println("THE FIBONACCI SERIES");
        System.out.println("enter the total number of terms");
        int n=sc.nextInt();
        System.out.println("THE GENERATED SERIES ::");
        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()
    {
        Scanner sc=new Scanner(System.in);
        int lower,upper;
```

```
        System.out.println("THE EVEN NUMBERS");
        System.out.println("enter the lower limit");
        lower=sc.nextInt();
        System.out.println("enter the upper limit");
        upper=sc.nextInt();
        System.out.println("the even numbers from"+lower+"and"+upper+"are");
        for(int i=lower;i<=upper;i++)
        {
            if(i%2!=0)
            {
                continue;
            }
            else
            {
                System.out.println(i+" ");
            }
        }
    }
```

```
public class threadprgrm
{
    public static void main(String[] args)throws InterruptedException
    {
        fibonacci obj1=new fibonacci();
        Thread a=new Thread(obj1);
        a.start();
        a.sleep(2000);
        evenNo obj2=new evenNo();
        Thread b=new Thread(obj2);
        b.start();
        b.sleep(1000);
    }
}
```

Output

```
C:\Users\ajcemca\Documents\Neha>javac threadprgrm.java

C:\Users\ajcemca\Documents\Neha>java threadprgrm
THE FIBONACCI SERIES
enter the total number of terms
3
THE GENERATED SERIES ::
0
1
1
2
THE EVEN NUMBERS
enter the lower limit
2
enter the upper limit
3
the even numbers from2and3are
2
```

Result

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

Experiment : 22**Aim :**

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

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;

class arrayStack
{
    public int arr[];
    public int top, size, len;

    public arrayStack(int n)
    {
        size = n;
        len = 0;
        arr = new int[size];
        top = -1;
    }
    public boolean isEmpty()
    {
        return top == -1;
    }
    public boolean isFull()
    {
        return top == size - 1 ;
    }
    public int peek()
    {
        return arr[top];
    }
}
```

```
public void push(int k)
{
    if(top + 1 >= size)
        System.out.println(" overflow ");
    if(top + 1 < size )
        arr[++top] = k;
}
public int pop()
{
    if( isEmpty() )
        System.out.println(" underflow ");
    return arr[top--];
}
public void display()
{
    System.out.print("\nStack = ");
    for (int i = top; i >= 0; i--)
        System.out.print(arr[i]+" ");
    System.out.println();
}
}
public class Genericstack
{
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter Size of the Stack ");
        int n = obj.nextInt();

        arrayStack stk = new arrayStack(n);

        int ch = 7;
        do{
            System.out.println("\nStack Operations");
            System.out.println("1. push");
```

```
System.out.println("2. pop");
System.out.println("3. peek");
System.out.println("4. check empty");
System.out.println("5. check full");
int choice = obj.nextInt();
switch (choice)
{
case 1 :
    System.out.println("Enter integer element to push");
    stk.push( obj.nextInt() );

    break;
case 2 :
    System.out.println("Popped Element = " + stk.pop());

    break;
case 3 :
    System.out.println("Peek Element = " + stk.peek());

    break;
case 4 :
    System.out.println("Empty status = " + stk.isEmpty());
    break;
case 5 :
    System.out.println("Full status = " + stk.isFull());
    break;
default :
    System.out.println("enter valid option \n ");
    break;
}
} while (ch != 0);
}
```

Output

```
C:\Users\ajcemca\Documents\Neha>javac GenericStack.java

C:\Users\ajcemca\Documents\Neha>java GenericStack
Enter Size of the Stack
2

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
1
Enter integer element to push
1

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
1
Enter integer element to push
2

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
1
Enter integer element to push
3
overflow
```

```
Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
2
Popped Element = 2

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
3
Peek Element = 1

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
4
Empty status = false

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
5
Full status = false

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
```

Result

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

Experiment : 23**Aim :**

Using generic method perform Bubble sort.

CO 4:

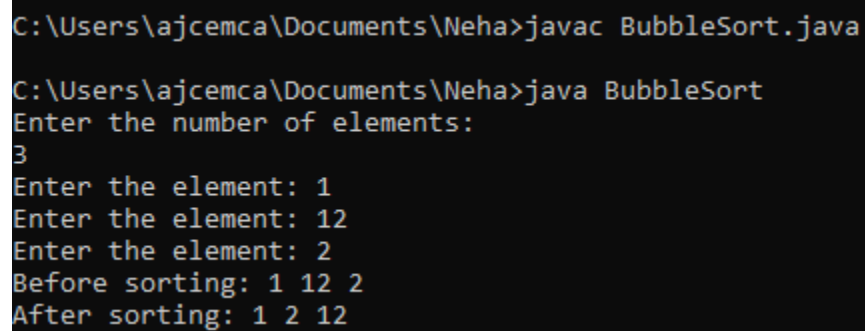
Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;
public class BubbleSort{
    int sz;
    int[] Arr;
    public BubbleSort(int n){
        sz = n;
        Arr = new int[sz];
    }
    public void insert(int i, int f){
        Arr[i] = f;
    }
    public void display(int i){
        System.out.print(Arr[i]+ " ");
    }
    public void Sort(int n){
        int temp;
        for(int i=0; i<n; i++){
            for(int j=i+1; j<n; j++){
                if(Arr[i] > Arr[j]){
                    temp = Arr[i];
                    Arr[i] = Arr[j];
                    Arr[j] = temp;
                }
            }
        }
    }
}
```

```
public static void main(String[] args){
    Scanner Snr= new Scanner(System.in);
    System.out.println("Enter the number of elements: ");
    int size = Snr.nextInt();
    BubbleSort arr = new BubbleSort(size);
    for(int i=0; i<size; i++){
        System.out.print("Enter the element: ");
        int val = Snr.nextInt();
        arr.insert(i, val);
    }
    System.out.print("Before sorting: ");
    for(int i=0; i<size; i++){
        arr.display(i);
    }
    System.out.print("\nAfter sorting: ");
    for(int i=0; i<size; i++){
        arr.Sort(size);
        arr.display(i);
    }
}
```

Output



```
C:\Users\ajcemca\Documents\Neha>javac BubbleSort.java
C:\Users\ajcemca\Documents\Neha>java BubbleSort
Enter the number of elements:
3
Enter the element: 1
Enter the element: 12
Enter the element: 2
Before sorting: 1 12 2
After sorting: 1 2 12
```

Result

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

Experiment : 24**Aim :**

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

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

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

```
public class arraylist
```

```
{
```

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

```
    {
```

```
        ArrayList<String> newlist=new ArrayList<String>();
```

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

```
        System.out.println("enter the size of the array :");
```

```
        int sz= obj.nextInt();
```

```
        String item;
```

```
        for(int i=0;i<sz;i++)
```

```
        {
```

```
            System.out.println("enter the array elements in string :");
```

```
            item=obj.next();
```

```
            newlist.add(item);
```

```
        }
```

```
        System.out.println("the array elements are :"+newlist);
```

```
    }
```

```
}
```

Output

```
C:\Users\ajcemca\Documents\Neha>javac arraylist.java  
  
C:\Users\ajcemca\Documents\Neha>java arraylist  
enter the size of the array :  
3  
enter the array elements in string :  
1  
enter the array elements in string :  
2  
enter the array elements in string :  
43  
the array elemnts are :[1, 2, 43]
```

Result

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

Experiment : 25**Aim :**

Program to remove all the elements from a linked list

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
public class CO4Q11
{
    public static void main(String ar[])
    {
        LinkedList<String> L=new LinkedList<>();
        L.add("gold");
        L.add("silver");
        L.add("bronze");
        System.out.println(L);
        L.remove("bronze");
        System.out.println(L);
        L.add("blue");
        L.add("green");
        L.add("white");
        L.add("black");
        System.out.println(L);
        L.remove(2);
        System.out.println(L);
    }
}
```

```
L.removeLast();  
  
System.out.println(L);  
  
L.removeFirst();  
  
System.out.println(L);  
  
}  
  
}
```

Output

```
A:\S2 Lab Programs\Oops>javac C04Q11.java  
  
A:\S2 Lab Programs\Oops>java C04Q11  
[gold, silver, bronze]  
[gold, silver]  
[gold, silver, blue, green, white, black]  
[gold, silver, green, white, black]  
[gold, silver, green, white]  
[silver, green, white]
```

Result

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

Experiment : 26**Aim :**

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

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

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

```
public class removeStack{
    public static void main(String args[])
    {
        Stack<String> StackDemo=new Stack<String>();

        Scanner obj=new Scanner(System.in);
        String n;
        System.out.println("enter the size");
        int sz=obj.nextInt();
        for(int i=0;i<sz;i++)
        {
            System.out.println("enter the value");
            n=obj.next();
            StackDemo.add(n);
        }

        System.out.println("Stack : "+ StackDemo);
        String rem_ele = StackDemo.remove(2);
        System.out.println("removed element : "+rem_ele);
        System.out.println("stack after remove operation : "+StackDemo);
    }
}
```

```
    }  
}
```

Output

```
C:\Users\ajcemca\Documents\Neha>javac removeStack.java  
  
C:\Users\ajcemca\Documents\Neha>java removeStack  
enter the size  
3  
enter the value  
1  
enter the value  
21  
enter the value  
20  
Stack : [1, 21, 20]  
removed element : 20  
stack after remove operation : [1, 21]
```

Result

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

Experiment : 27**Aim :**

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

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;
class priorityQ{
    public static void main(String args[])
    {
        PriorityQueue<Integer> pQ = new PriorityQueue<Integer>();

        Scanner obj=new Scanner(System.in);
        int n;
        System.out.println("enter the size");
        int sz=obj.nextInt();
        for(int i=0;i<sz;i++)
        {
            System.out.println("enter the value");
            n=obj.nextInt();
            pQ.add(n);
        }

        System.out.println(pQ.peek());
        System.out.println(pQ);
        System.out.println(pQ.poll());
        System.out.println(pQ.peek());
        System.out.println(pQ);
    }
}
```

Output

```
C:\Users\ajcemca\Documents\Neha>javac priorityQ.java

C:\Users\ajcemca\Documents\Neha>java priorityQ
enter the size
3
enter the value
21
enter the value
11
enter the value
32
11
[11, 21, 32]
11
21
[21, 32]
```

Result

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

Experiment : 28**Aim :**

Program to demonstrate the addition and deletion of elements in deque

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;
public class Dqueue{
    public static void main(String args[])
    {
        Deque<Integer> dq = new ArrayDeque<>();

        Scanner obj=new Scanner(System.in);
        int n;
        System.out.println("enter the size");
        int sz=obj.nextInt();
        for(int i=0;i<sz;i++)
        {
            System.out.println("enter the value through front end");
            n=obj.nextInt();
            dq.addFirst(n);
        }
        for(int i=0;i<sz;i++)
        {
            System.out.println("enter the value through last end");
            n=obj.nextInt();
            dq.addLast(n);
        }

        int first = dq.removeFirst();
        int last = dq.removeLast();
        System.out.println("first :"+ first +" last :"+ last);
    }
}
```

```
    }  
}
```

Output

```
C:\Users\ajcemca\Documents\Neha>javac Dqueue.java  
  
C:\Users\ajcemca\Documents\Neha>java Dqueue  
enter the size  
3  
enter the value through front end  
21  
enter the value through front end  
11  
enter the value through front end  
34  
enter the value through last end  
11  
enter the value through last end  
22  
enter the value through last end  
23  
first :34 last :23
```

Result

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

Experiment : 29**Aim :**

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

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;

public class CO4Q15{

    public static void main(String ar[])

    {

        HashSet <Integer> hs=new HashSet<>();

        hs.add(1);

        hs.add(2);

        hs.add(3);

        hs.add(4);

        hs.add(5);

        hs.add(6);

        System.out.println("set= "+hs);

        hs.remove(5);

        System.out.println("set= "+hs);

        int x=hs.size();

        System.out.println("size of set object= "+x);

        hs.remove(2);
```

```
        System.out.println("set= "+hs);

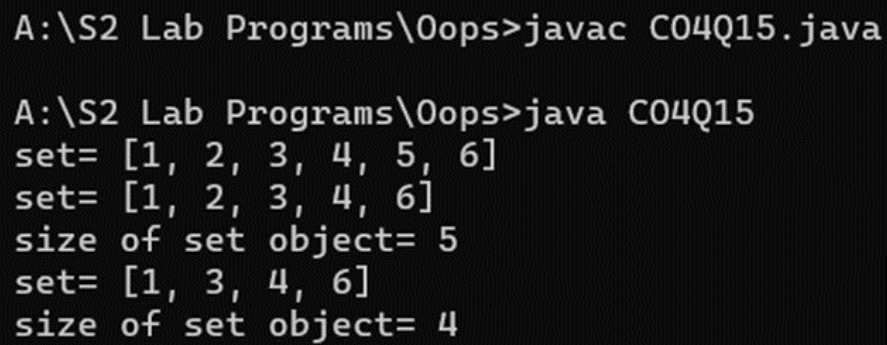
        int y=hs.size();

        System.out.println("size of set object= "+y);

    }

}
```

Output

A screenshot of a Windows command prompt window with a black background and white text. The prompt shows the execution of a Java program. The first command is 'javac C04Q15.java' and the second is 'java C04Q15'. The output of the program is displayed below the second command, showing the state of a set and its size at different points in the execution.

```
A:\S2 Lab Programs\Ooops>javac C04Q15.java

A:\S2 Lab Programs\Ooops>java C04Q15
set= [1, 2, 3, 4, 5, 6]
set= [1, 2, 3, 4, 6]
size of set object= 5
set= [1, 3, 4, 6]
size of set object= 4
```

Result

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

Experiment : 30**Aim :**

Write a Java program to compare two hash set.

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
public class CO4Q16{

    public static void main(String[] args) {

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

        h_set.add("Red");

        h_set.add("Green");

        h_set.add("Black");

        h_set.add("Orange");

        h_set.add("Pink");

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

        h_set2.add("Red");

        h_set2.add("Pink");

        h_set2.add("Black");

        h_set2.add("Orange");

        for (String element : h_set){

            System.out.println(h_set2.contains(element) ? "Yes" : "No");

        }

    }

}
```

Output

```
A:\S2 Lab Programs\Oops>javac C04Q16.java
```

```
A:\S2 Lab Programs\Oops>java C04Q16
```

```
Yes
```

```
Yes
```

```
Yes
```

```
Yes
```

```
No
```

Result

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

Experiment : 31**Aim :**

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

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;

public class CO4Q17 {

    public static void main(String args[])

    {

        Map<Integer, String> hm2= new HashMap<Integer, String>();

        hm2.put(new Integer(1), "All");

        hm2.put(new Integer(2), "are");

        hm2.put(new Integer(3), "Equal");

        System.out.println(hm2);

        hm2.put(new Integer(2), "Core");

        System.out.println(hm2);

        hm2.remove(new Integer(1));

        System.out.println(hm2);

    }}

```

Output

```
A:\S2 Lab Programs\Ooops>java C04Q17
{1=All, 2=are, 3=Equal}
{1=All, 2=Core, 3=Equal}
{2=Core, 3=Equal}
```

Result

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

Experiment : 32**Aim :**

Program to Convert HashMap to TreeMap .

CO 4:

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;

public class CO4Q18{

    public static void main(String ar[]){

        Map<String,String> map=new HashMap<>();

        System.out.println("enter the limit");

        Scanner sc=new Scanner(System.in);

        int n= sc.nextInt();

        System.out.println("enter the roll number and name");

        while(n!=0)
        {
            String e=sc.next();
            String s=sc.next();
            map.put(e,s);
            n--;
        }

        System.out.println("hashmap"+map);

        Map<String,String> treeMap=new TreeMap<>();

        treeMap.putAll(map);

        System.out.println("Treemap"+treeMap);
```

```
}}
```

Output

```
A:\S2 Lab Programs\Oops>java C04Q18
enter the limit
2
enter the roll number and name
12
anna
17
anina
hashmap{12=anna, 17=anina}
Treemap{12=anna, 17=anina}
```

Result

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

Experiment : 33**Aim :**

Program to draw Circle, Rectangle, Line in Applet.

CO 5:

Develop applications to handle events using applets

Procedure

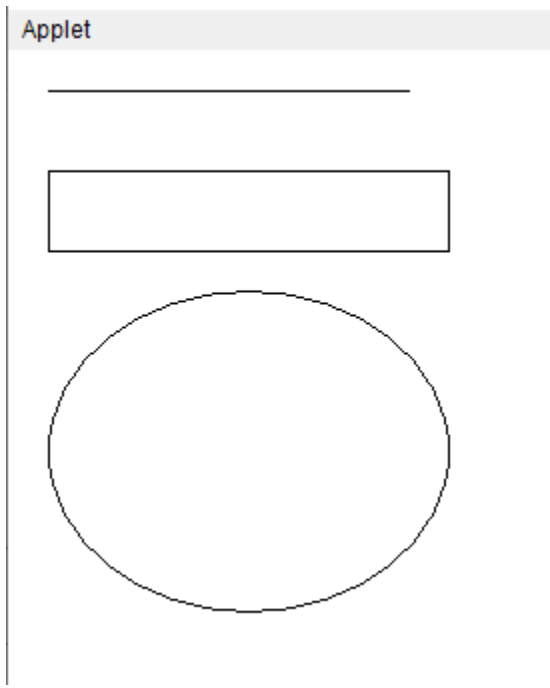
```
import java.awt.*;
import java.applet.*;
public class shape extends Applet
{

    public void paint(Graphics g)
    {
        g.drawLine(20,20,200,20);
        g.drawRect(20,60,200,40);
        g.drawOval(20,120,200,160);
    }
}
```

```
<html>
<head>
<title>
</title>
</head>
<body>
<div align="center">
<applet code="shape.class" width="800" height="500">
</applet>
</div>
</body>
</html>
```

Output

```
C:\java\bin>javac shape.java  
C:\java\bin>appletviewer first.html
```



Result

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

Experiment : 34**Aim :**

Program to find maximum of three numbers using AWT

CO 5:

Develop applications to handle events using applets

Procedure

```
import java.awt.*;

import java.applet.*;

public class MaxOfThree extends Applet
{
    TextField T1,T2,T3;

    public void init(){
        T1 = new TextField(10);
        T2 = new TextField(10);
        T3 = new TextField(10);

        add(T1);

        add(T2);

        add(T3);

        T1.setText("0");
        T2.setText("0");
        T3.setText("0");
    }

    public void paint(Graphics g){
```

```
int a, b, c,result;

String str;

g.drawString("Enter value to Check the Maximum of 3 ",10,50);

str=T1.getText();
a=Integer.parseInt(str);
str=T2.getText();
b=Integer.parseInt(str);
str=T3.getText();
c=Integer.parseInt(str);

g.setColor(Color.blue);
if (a>b) {
    if (a>c)
        result=a;
    else
        result=c;
}
else{
    if (b>c)
        result=b;
    else
```



```
        result=c;

    }

    g.drawString("Maximum of 3 No is "+result,10,70);

    showStatus("MAXIMUM OF 3 NUMBERS");

}

public boolean action(Event e, Object o){

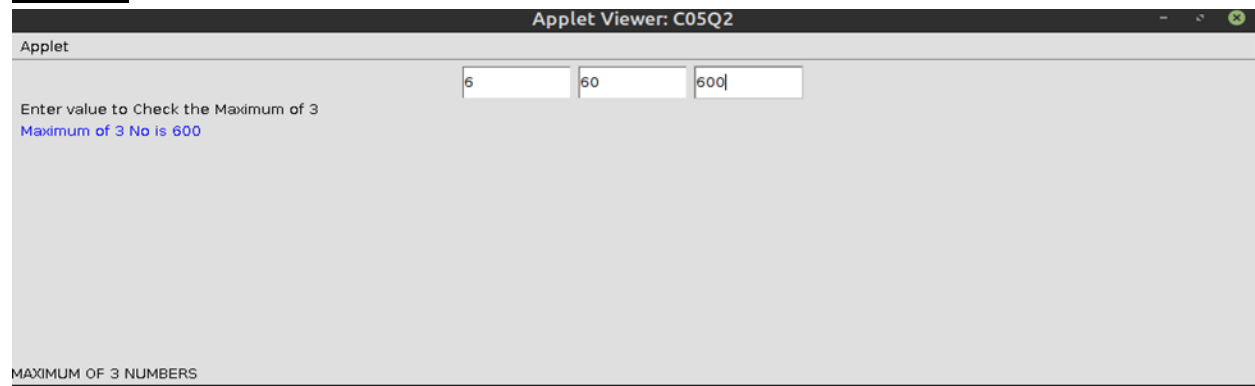
    repaint();

    return true;

}

}
```

Output



Result

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

Experiment : 35**Aim :**

Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.

CO 5:

Develop applications to handle events using applets

Procedure

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

```
public class happysad extends Applet implements ActionListener {
```

```
    TextField t1, t2, t3, t4;
```

```
    Button b;
```

```
    Label l1, l2, l3, l4;
```

```
    public void init() {
```

```
        l1 = new Label("mark1");
```

```
        t1 = new TextField(5);
```

```
        l2 = new Label("mark2");
```

```
        t2 = new TextField(5);
```

```
        l3 = new Label("mark3");
```

```
        t3 = new TextField(5);
```

```
        l4 = new Label("result");
```

```
        t4 = new TextField(5);
```

```
        t1.setBounds(210, 40, 100, 20);
```

```
        t2.setBounds(210, 80, 100, 20);
```

```
        t3.setBounds(210, 120, 100, 20);
```

```
        t4.setBounds(210, 140, 100, 20);
```

```
        l1.setBounds(100, 40, 100, 20);
```

```
12.setBounds(100, 80, 100, 20);
13.setBounds(100, 120, 100, 20);
14.setBounds(100, 140, 100, 20);

b = new Button("find");
b.setBounds(230, 150, 60, 50);

add(l1);
add(l2);
add(l3);
add(l4);
add(t1);
add(t2);
add(t3);
add(t4);
add(b);
b.addActionListener(this);
}

public void actionPerformed(ActionEvent e) {
    int x = 0;
    int y = 0;
    int z = 0;
    int total = 0;

    try {
        x = Integer.parseInt(t1.getText());
        y = Integer.parseInt(t2.getText());
        z = Integer.parseInt(t3.getText());

        total = (x + y + z) / 3;
        t4.setText(String.valueOf(total));
    } catch (NumberFormatException ex) {
        t4.setText("Invalid input");
    }
}
```

```
}

@Override
public void paint(Graphics g) {
    int x = 0;
    int y = 0;
    int z = 0;
    int total = 0;

    x = Integer.parseInt(t1.getText());
    y = Integer.parseInt(t2.getText());
    z = Integer.parseInt(t3.getText());

    total = (x + y + z) / 3;

    if (total > 50) {
        g.setColor(Color.YELLOW);
        g.fillOval(80, 70, 150, 150);

        g.setColor(Color.BLACK);
        g.fillOval(120, 120, 15, 15);
        g.fillOval(170, 120, 15, 15);

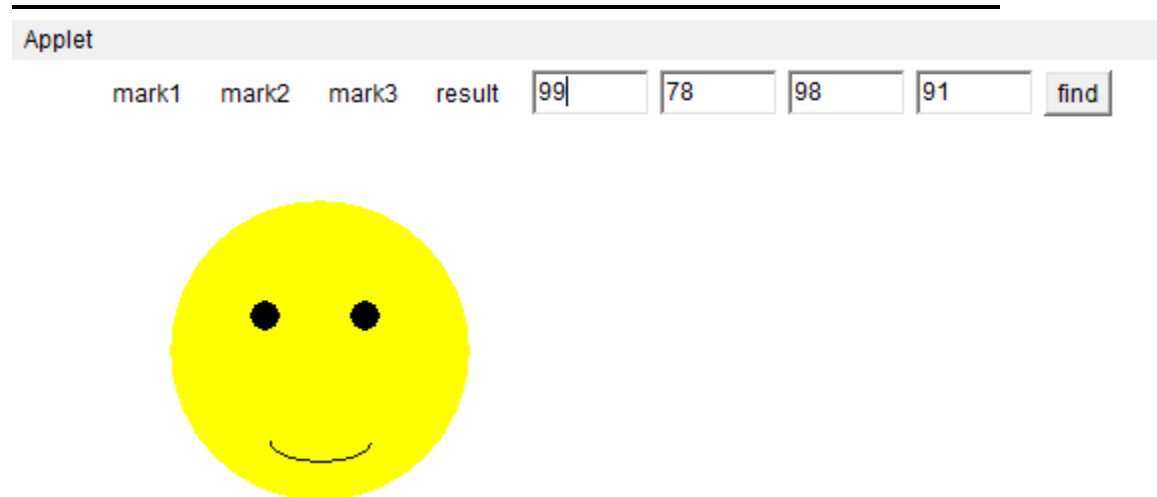
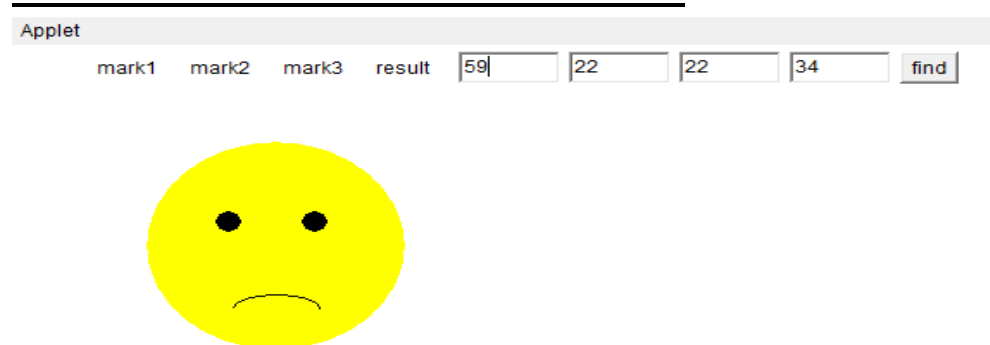
        g.drawArc(130, 180, 50, 20, 180, 180);
    } else {
        g.setColor(Color.YELLOW);
        g.fillOval(80, 70, 150, 150);

        g.setColor(Color.BLACK);
        g.fillOval(120, 120, 15, 15);
        g.fillOval(170, 120, 15, 15);
        g.drawArc(130, 180, 50, 20, 180, -180);
    }
}
}
```

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

Output

```
C:\java\bin>javac happysad.java  
C:\java\bin>appletviewer happysad.java
```



Result

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

Experiment : 36**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.

CO 5:

Develop applications to handle events using applets

Procedure

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class house 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.GRAY);
        g.fillPolygon(x,y,3);
        g.drawRect(150,150,150,200);
        g.setColor(Color.CYAN);
```

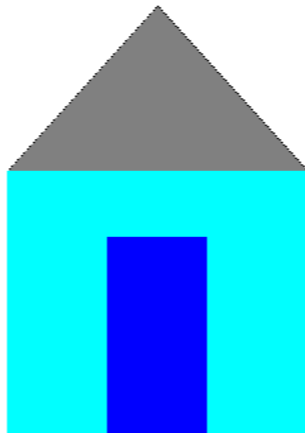
```
g.fillRect(150,150,150,200);
g.drawRect(200,200,50,150);
g.setColor(Color.blue);
g.fillRect(200,200,50,150);
if(a>200 && a<300 && b>200 && b<300)
{
g.setColor(Color.red);
g.fillRect(200, 200, 50, 150);
}
}
public void mouseClicked(MouseEvent e)
{
}
public void mouseClicked(MouseEvent e)
{
}
public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

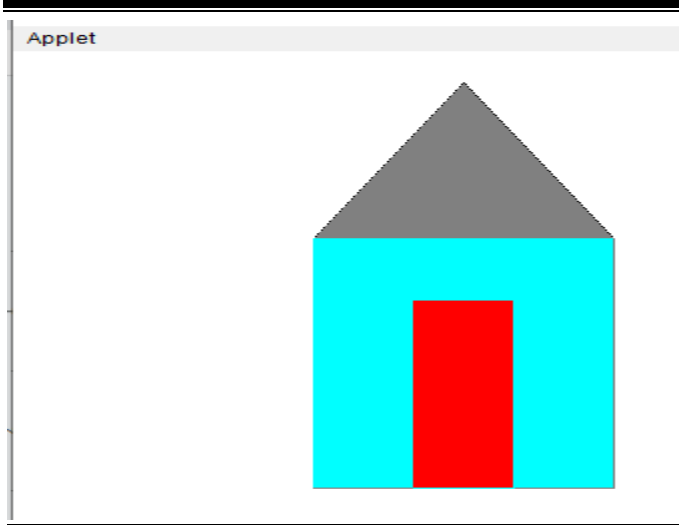
```
b=e.getY();  
repaint();  
}  
public void mouseReleased(MouseEvent e)  
{  
}  
}  
  
<html>  
<body>  
<applet code="house.class" width="700" height="700">  
</applet>  
</body>  
</html>
```

Output

```
C:\java\bin>javac houseNeha.java  
C:\java\bin>appletviewer nhouse.html
```

Applet



**Result**

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

Experiment : 37**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.

CO 5:

Develop applications to handle events using applets

Procedure

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.awt.Graphics;

public class fourshapes extends Applet implements ItemListener

{

    Choice c;

    int n;


    public void init()

    {

        Label l;

        l=new Label("Select an option");

        l.setBounds(80,100,100,50);

        add(l);
```

```
        c=new Choice();
        c.addItem("Choose shape:");
        c.addItem("Rectangle");
        c.addItem("Triangle");
        c.addItem("Square");
        c.addItem("Circle");
        c.addItemListener(this);
        add(c);
    }
    public void paint(Graphics g)
    {
        if(n==0)
        {
        }
        if(n==1)
        {
            g.drawRect(20,60,200,40);
            g.setColor(Color.yellow);
            g.fillRect(20,60,200,40);
            g.setColor(Color.pink);
        }
        if(n==2)
        {
```

```
        int[] x = new int[]{50, 50,200};

        int[] y = new int[]{500, 400, 500};

        g.drawPolygon(x,y,3);

        g.setColor(Color.blue);

        g.fillPolygon(x,y,3);

    }

    if(n==3)

    {

        g.drawRect(100, 100, 100, 100);

        g.setColor(Color.pink);

        g.fillRect(100,100,100,100);

        g.setColor(Color.gray);

    }

    if(n==4)

    {

        g.setColor(Color.red);

        g.drawOval(20, 120, 200, 160);

        g.fillOval(20,120,200,160);

        g.setColor(Color.green);

    }

}
```

```
        public void itemStateChanged (ItemEvent e)
        {
            n = c.getSelectedIndex();
            c.repaint();
        }
    }
/*
<html>
<head>
<title>
</title>
</head>
<body>
<div align="center">
<applet code="fourshapes.class" width="800" height="500">
</applet>
</div>
</body>
</html>
*/
```

Output

```
C:\java\bin>javac fourshapes.java  
C:\java\bin>appletviewer fourshapes.java
```

Applet

Select an option

Rectangle ▾

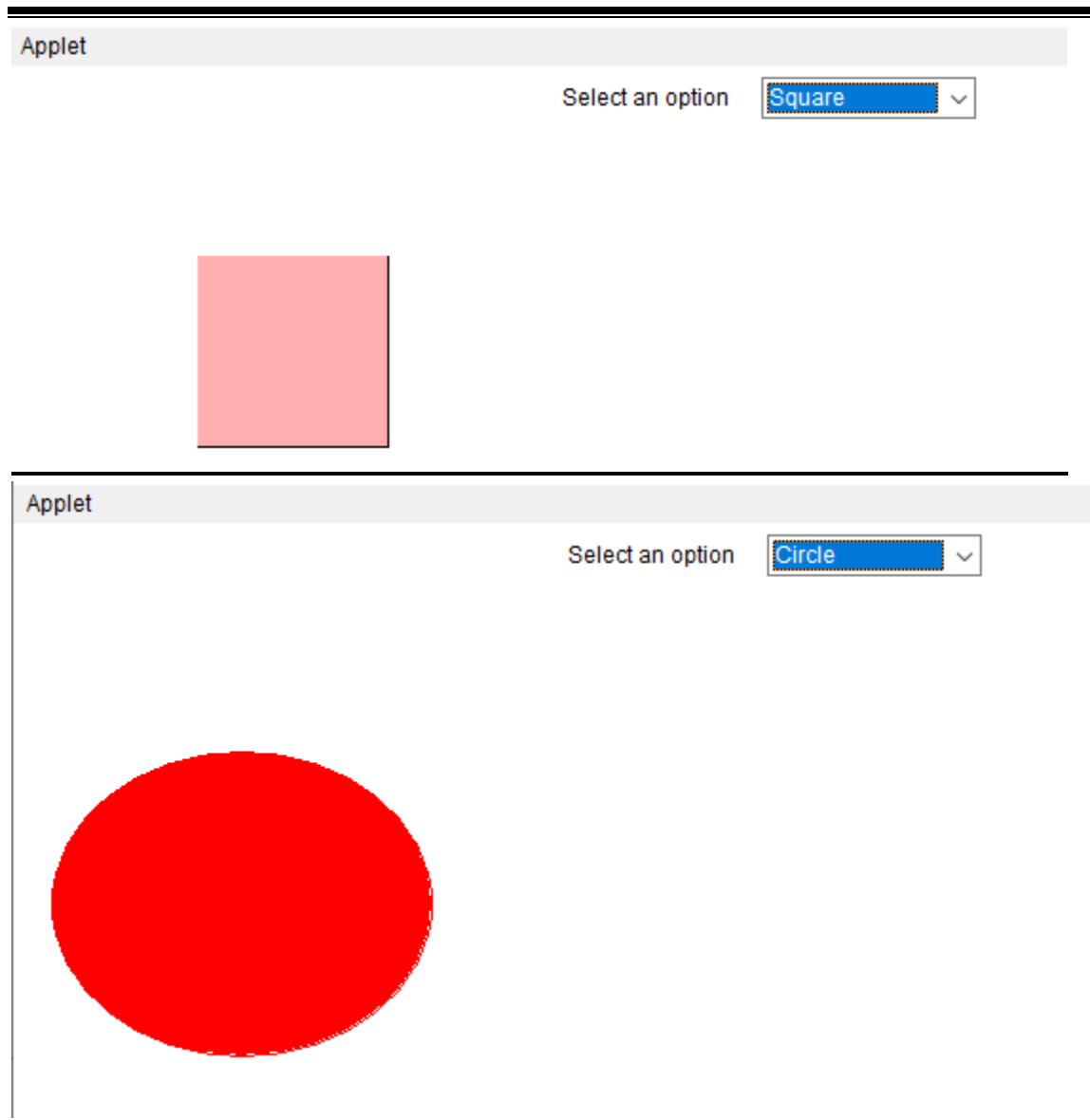


Applet

Select an option

Triangle ▾



**Result**

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

Experiment : 38**Aim :**

Develop a program to handle all mouse events and window events

CO6:

Develop applications using files and networking concepts

Procedure

```
import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

class Mouse extends Frame implements MouseMotionListener, MouseListener {

    static JLabel label1, label2, label3, label4, label5;

    Mouse()

    {

    }

    public static void main(String[] args)

    {

        JFrame f = new JFrame("all mouse events and window events");

        f.setSize(900, 300);

        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel p = new JPanel();

        JPanel p1 = new JPanel();

        f.setLayout(new FlowLayout());

        JLabel l1, l2;

        l1 = new JLabel("MouseMotionListener events :");
```

```
l2 = new JLabel("MouseListener events :");

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

Mouse m = new Mouse();

f.addMouseMotionListener(m);
f.addMouseListener(m);

p.add(l1);
p.add(label1);
p.add(label2);

p1.add(l2);
p1.add(label3);
p1.add(label4);
p1.add(label5);

f.add(p);
f.add(p1);

f.show();

}

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());

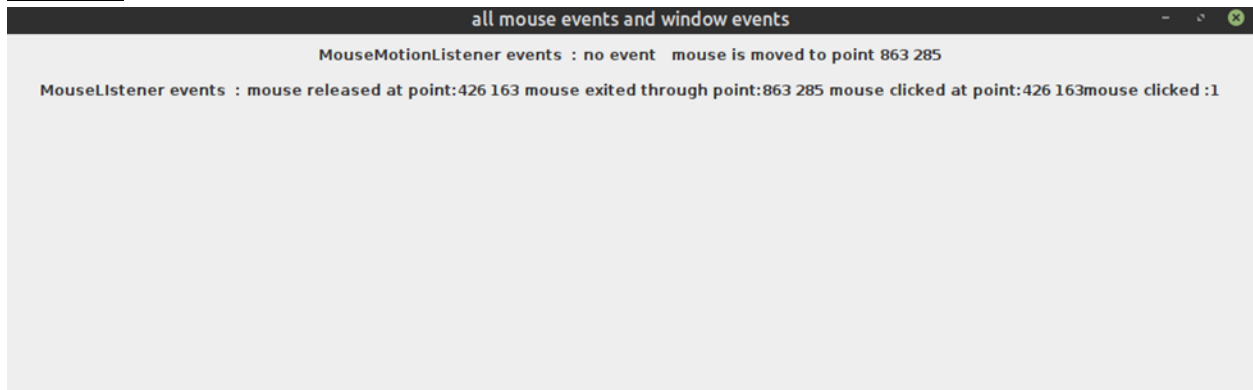
}

public void mouseClicked(MouseEvent e)

{
```

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

Output



Result

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

Experiment : 39**Aim :**

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

CO 6:

Develop applications using files and networking concepts

Procedure

```
import java.io.BufferedReader;

import java.io.FileWriter;

import java.io.FileReader;

import java.io.IOException;

public class filerw

{

    public static void main(String args[])

    {

        try

        {

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

            Writer.write("Welcome to java");

            Writer.close();

            FileReader Reader=new FileReader("Files.txt");

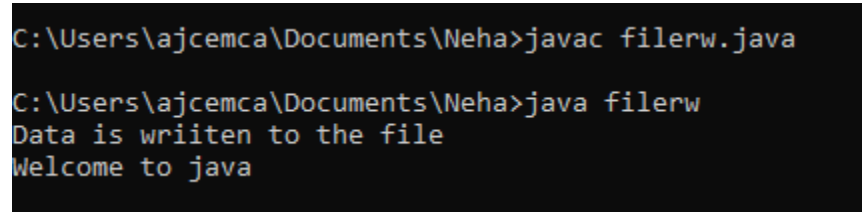
            BufferedReader br=new BufferedReader(Reader);

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

            String line;
```

```
        while((line=br.readLine())!=null)
        {
            System.out.println(line);
        }
        Reader.close();
    }
    catch(IOException e)
    {
        System.out.println("Error");
    }
}
}
```

Output



```
C:\Users\ajcemca\Documents\Neha>javac filerw.java
C:\Users\ajcemca\Documents\Neha>java filerw
Data is writen to the file
Welcome to java
```

Result

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

Experiment : 40**Aim :**

Write a program to copy one file to another.

CO 6:

Develop applications using files and networking concepts

Procedure

```
import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

public class CO6Q3 {

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

        FileInputStream fileinput = new FileInputStream("co6q2.txt");

        FileOutputStream fileoutput = new FileOutputStream("co6q3.txt");

        int i;

        while ((i = fileinput.read()) != -1) {

            fileoutput.write(i);

        }

        System.out.println("Successfully copied one file to another");

        fileinput.close();

        fileoutput.close();

    }

}
```

Output

```
A:\S2 Lab Programs\Ops>javac C06Q3.java  
  
A:\S2 Lab Programs\Ops>java C06Q3  
Successfully copied one file to another
```



co6q3 - Notepad

File Edit View

new file is created

Result

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

Experiment : 41**Aim :**

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

CO 6:

Develop applications using files and networking concepts

Procedure

```
import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

public class CO6Q4

{

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

        FileInputStream source = new FileInputStream ("source.txt");

        FileOutputStream destination_odd = new FileOutputStream ("odd.txt");

        FileOutputStream destination_even = new FileOutputStream ("even.txt");

        int i;

        while((i = source.read()) != -1){

            if(i%2==0) {

                destination_even.write(i);

            }

            else {

                destination_odd.write(i);

            }

        }

    }

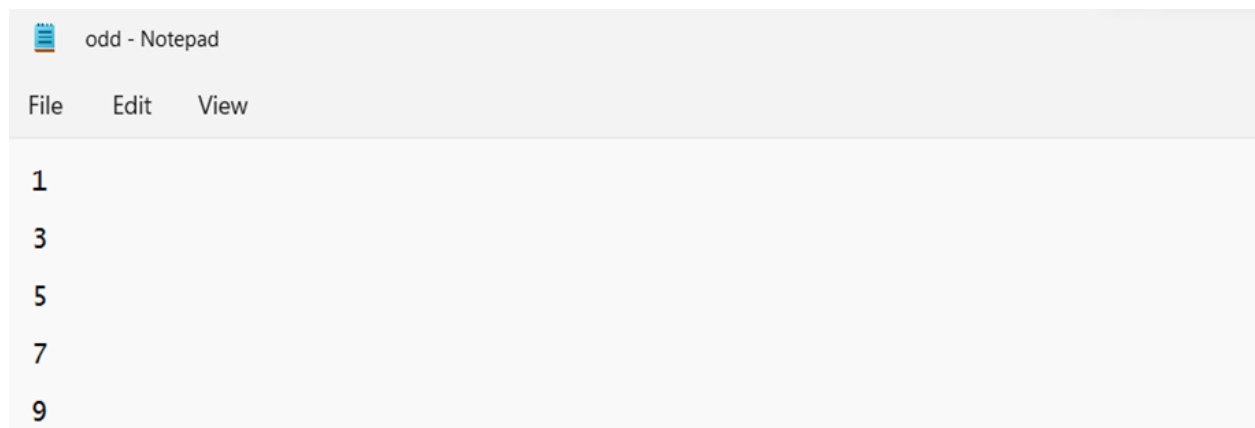
}
```


```
    }  
  
    }  
  
    System.out.println("copied");  
  
    source.close();  
  
    destination_even.close();  
  
    destination_odd.close();  
  
    }  
  
}
```

Output

```
A:\S2 Lab Programs\Oops>javac C06Q4.java
```

```
A:\S2 Lab Programs\Oops>java C06Q4  
copied
```



 even - Notepad
File Edit View

2
4
6
8

Result

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

Experiment : 42**Aim :**

Client Server communication using DatagramSocket - UDP

CO 6:

Develop applications using files and networking concepts

Procedure

```
import java.io.*;

import java.net.*;

public class Myser {

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

        DatagramSocket server=new DatagramSocket(9000);

        byte[] buf=new byte[256];

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

        server.receive(packet);

        String response =new String(packet.getData());

        System.out.println(" Server : "+response);

        server.close();

    }

}

import java.io.*;

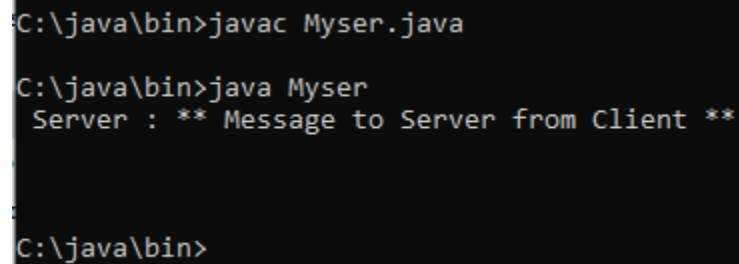
import java.net.*;

public class Mycl {

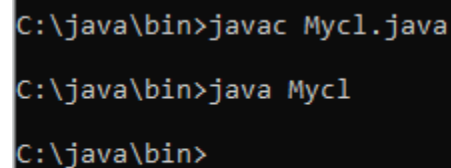
    public static void main(String[] args) throws IOException {
```

```
DatagramSocket client= new DatagramSocket();  
  
InetAddress add=InetAddress.getByName("localhost");  
  
String str = "** Message to Server from Client **";  
  
byte[] bufBytes = str.getBytes();  
  
DatagramPacket  
DatagramPacket(bufBytes,bufBytes.length,add,9000);  
  
client.send(datagramPacket);  
  
client.close();  
  
}  
  
}
```

Output



```
C:\java\bin>javac Myser.java  
  
C:\java\bin>java Myser  
Server : ** Message to Server from Client **  
  
C:\java\bin>
```



```
C:\java\bin>javac Mycl.java  
  
C:\java\bin>java Mycl  
  
C:\java\bin>
```

Result

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

