20MCA132 – OBJECT ORIENTED PROGRAMMING LAB

Lab Report Submitted By

Jenny Johnson

Reg. No.: AJC22MCA-2053

In Partial Fulfilment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS (2 Year) (MCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovapally, Kanjirappally, Kottayam, Kerala – 686518]

2022-2023

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the lab report, "**20MCA132 OBJECT ORIENTED PROGRAMMING LAB"** is the bonafide work of **JENNY JOHNSON (AJC22MCA-2053)** in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year **2022-23.**

Ms. Ankitha Philip
Lab In- Charge

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

Internal Examiner

External Examiner

Course Code	Course Name	Syllabus Year	L-T-P-C
20MCA132	Object Oriented Programming Lab	2020	0-1-3-2

VISION

To promote an academic and research environment conducive for innovation centric technical education.

MISSION

MS1 - Provide foundations and advanced technical education in both theoretical and applied Computer Applications in-line with Industry demands.

MS2 - Create highly skilled computer professionals capable of designing and innovating real life solutions.

MS3 - Sustain an academic environment conducive to research and teaching focused to generate up-skilledprofessionals with ethical values.

MS4 - Promote entrepreneurial initiatives and innovations capable of bridging and contributing with sustainable, socially relevant technology solutions.

COURSE OUTCOME

CO	Outcome	Target
CO1	Understand object-oriented concepts and design classes and objects to solve problems.	60
CO2	Familiarization and understanding of arrays and strings.	60
CO3	Understand and implement object-oriented concepts like inheritance, overloading and interfaces.	60
CO4	Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.	60
CO5	Develop applications to handle events using applets	60
CO6	Develop applications using files and networking concepts.	60

COURSE END SURVEY

CO	Survey Question	Answer Format
CO1	To what extend you are able to understand object-oriented concepts and design classes and objects to solve problems?	Excellent/Very Good/Good /Fair/Poor
CO2	To what extend you are able to implement arrays and strings?	Excellent/Very Good/Good /Fair/Poor
CO3	To what extend you are able to implement object-oriented concepts like inheritance, overloading and interfaces?	Excellent/Very Good/Good /Fair/Poor
CO4	To what extend you are able to implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework?	Excellent/Very Good/Good /Fair/Poor
CO5	To what extent you are able to develop applications to handle events using applets?	Excellent/Very Good/Good /Fair/Poor
CO6	To what extend you are able to develop applications using files and networking concepts?	Excellent/Very Good/Good /Fair/Poor

CONTENT

Sl. No.	Experiment	Date	CO	Page No.
1	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.	07-03-2023	CO1	1
2	Read 2 matrices from the console and perform matrix addition.	09-03-2023	CO1	3
3	Add complex numbers	09-03-2023	CO1	5
4	Read a matrix from the console and check whether it is symmetric or not. .	14-03-2023	CO1	6
5	Program to Sort strings	16-03-2023	CO2	8
6	Search an element in an array.	16-03-2023	CO2	10
7	Perform string manipulations	16-03-2023	CO2	12
8	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.	16-03-2023	CO2	13
9	Area of different shapes using overloaded functions	21-03-2023	CO3	15
10	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.	21-03-2023	CO3	18

Sl. No.	Experiment	Date	CO	Page No.
11	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.	23-03-2023	CO3	21
12	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.	23-03-2023	CO3	24
13	Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.	28-03-2023	CO3	27
14	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.	28-03-2023	CO3	30
15	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	28-03-2023	CO3	32

Sl. No.	Experiment	Date	CO	Page No.
16	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.	04-04-2023	CO4	35
17	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.	04-04-2023	CO4	38
18	Write a user defined exception class to authenticate the user name and password.	11-04-2023	CO4	40
19	Find the average of N positive integers, raising a user defined exception for each negative input.	11-04-2023	CO4	42
20	Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class).	11-04-2023	CO4	43
21	Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface) .	13-04-2023	CO4	45
22	Program to create a generic stack and do the Push and Pop operations.	13-04-2023	CO4	47
23	Using generic method perform Bubble sort.	18-04-2023	CO4	49
24	Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.	18-04-2023	CO4	50
25	Program to remove all the elements from a linked list.	18-04-2023	CO4	52
26	Program to remove an object from the Stack when the position is passed as parameter.	27-04-2023	CO4	54
27	Program to demonstrate the creation of queue object using the PriorityQueue class.	01-06-2023	CO4	55

Sl. No.	Experiment	Date	CO	Page No.
28	Program to demonstrate the addition and deletion of elements in deque.	01-06-2023	CO4	56
29	Program to demonstrate the creation of Set object using the LinkedHashset class.	08-06-2023	CO4	57
30	Write a Java program to compare two hash set.	08-06-2023	CO4	59
31	Program to demonstrate the working of Map interfaceby adding, changing and removing elements.	13-06-2023	CO4	61
32	Program to Convert HashMap to TreeMap.	13-06-2023	CO4	62
33	Program to draw Circle, Rectangle, Line in Applet.	20-06-2023	CO5	63
34	Program to find maximum of three numbers using AWT.	20-06-2023	CO5	64
35	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.	22-06-2023	CO5	66
36	Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.	22-06-2023	CO5	69
37	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.	27-06-2023	CO5	71
38	Develop a program to handle all mouse events and window events.	04-07-2023	CO5	73
39	Write a program to write to a file, then read from the file and display the contents on the console.	20-07-2023	CO6	76
40	Write a program to copy one file to another.	20-07-2023	CO6	77
41	Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.	22-07-2023	CO6	78
42	Client Server communication using DatagramSocket - UDP	27-07-2023	CO6	80

Experiment No.: 1

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

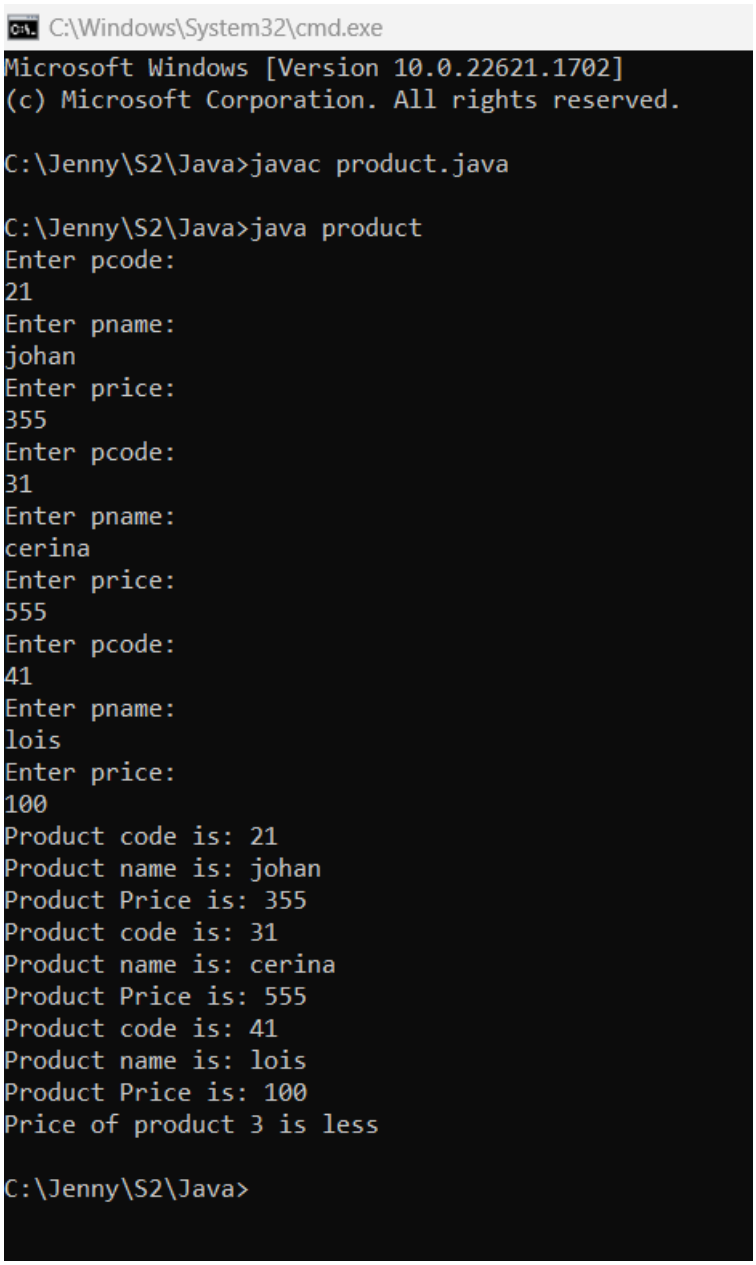
CO1: Understand object-oriented concepts and design classes and objects to solve problems.

Procedure

```
import java.util.*;
public class product
{
int pcode,price;
String pname;
public void get()
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter pcode: ");
pcode=sc.nextInt();
System.out.println("Enter pname: ");
pname=sc.next();
System.out.println("Enter price: ");
price=sc.nextInt();
}
public void display()
{
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.get();
p2.get();
p3.get();
p1.display();
p2.display();
p3.display();
if(p1.price<p2.price&&p1.price<p3.price)
{
        System.out.println("Price of first product is less");
}
else if(p2.price<p1.price&&p2.price<p3.price)
{
        System.out.println("Price of product 3 is less");
```

```
}
else
{
        System.out.println("Price of product 3 is less");
}
}
}
```

Output Screenshot



Result

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

Experiment No.: 2

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

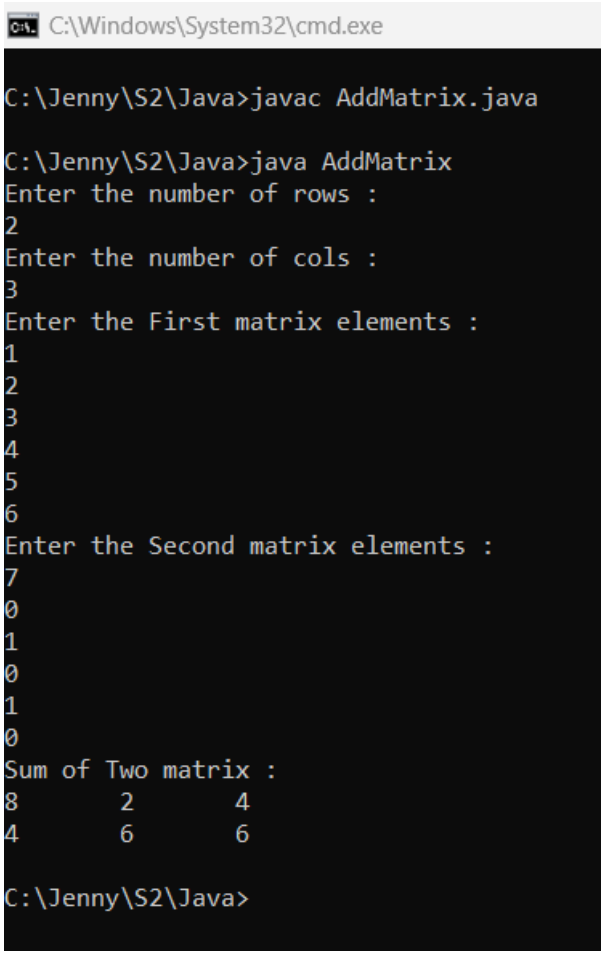
CO1: Understand object-oriented concepts and design classes and objects to solve problems.

Procedure

```
import java.util.*;
public class AddMatrix
{
public static void main(String args[])
{
int [][] a = new int [50][50];
int [][] b = new int [50][50];
int [][] c = new int [50][50];
int i,j;
Scanner sc = new Scanner(System.in);
System.out.println("Enter the number of rows : ");
int n=sc.nextInt();
System.out.println("Enter the number of cols : ");
int m=sc.nextInt();
System.out.println("Enter the First matrix elements : ");
for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
{
a[i][j]=sc.nextInt();
}}
System.out.println("Enter the Second matrix elements : ");
for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
{
b[i][j]=sc.nextInt();
}}
for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
{
c[i][j]=a[i][j]+b[i][j];
}}
System.out.println("Sum of Two matrix : ");
for (i=0;i<n;i++)
{
for (j=0;j<m;j++)
{
System.out.print(c[i][j]+"\t");
```

```
}
System.out.println();
}
}
}
```

Output Screenshot



Result

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

Experiment No.: 3

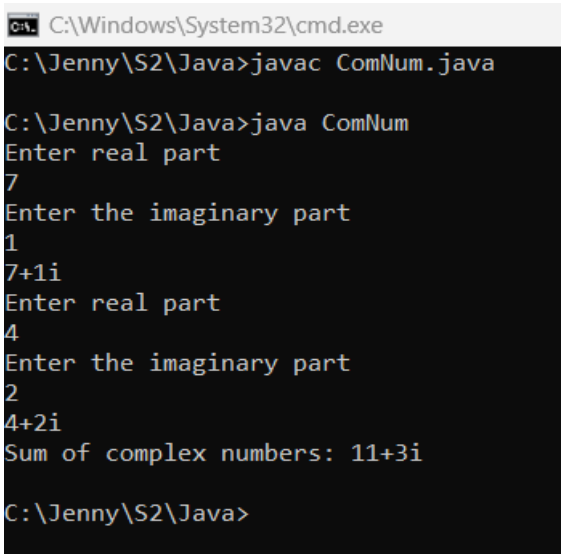
Aim: Add complex numbers

CO1: Understand object-oriented concepts and design classes and objects to solve problems.

Procedure

```
import java.util.*;
public class ComNum{
 public static void main(String[] args)  {
   Scanner cn=new Scanner(System.in);
        System.out.println("Enter real part");
        int re=cn.nextInt();
        System.out.println("Enter the imaginary part");
        int im=cn.nextInt();
        System.out.println(re+"+"+im+"i");
        System.out.println("Enter real part");
        int rea=cn.nextInt();
        System.out.println("Enter the imaginary part");
        int ima=cn.nextInt();
        System.out.println(rea+"+"+ima+"i");
        System.out.println("Sum of complex numbers: "+(re+rea)+"+"+(im+ima)+"i");
        }}
```

Output Screenshot



Result

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

Experiment No.: 4

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

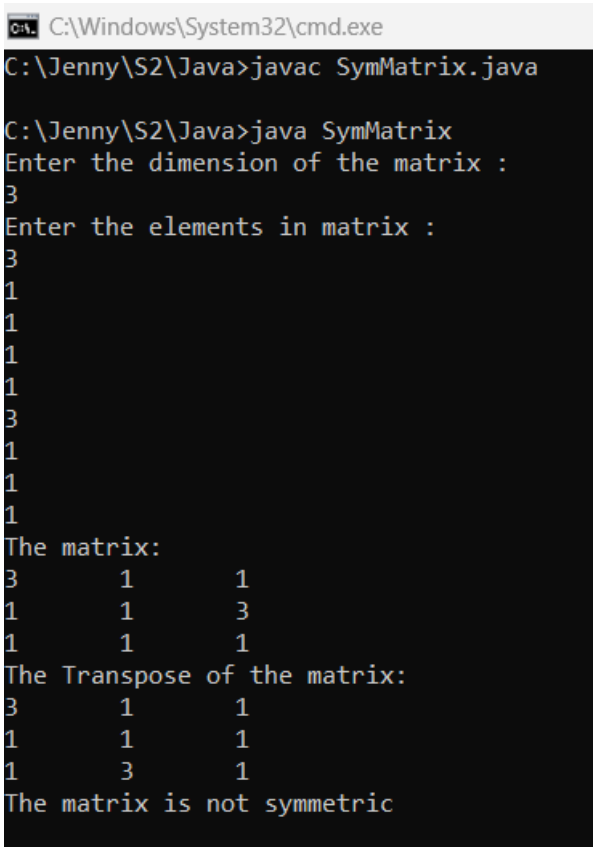
CO1: Understand object-oriented concepts and design classes and objects to solve problems.

Procedure

```
import java.util.*;
public class SymMatrix
{
	public static void main(String[] args)
	{
		Scanner Snr = new Scanner(System.in);
		System.out.println("Enter the dimension of the matrix :");
		int sz = Snr.nextInt();
		int Arr[][] = new int[sz][sz];
		int Arr1[][] = new int[sz][sz];
		System.out.println("Enter the elements in matrix :");
		for(int i=0;i<sz;i++)
		{
		for(int j=0;j<sz;j++)
			{
				Arr[i][j] = Snr.nextInt();
			}}
		System.out.println("The matrix: ");
		for(int i=0;i<sz;i++)
		{
			for(int j=0;j<sz;j++)
			{
				System.out.print(Arr[i][j] + " ");
			}
			System.out.println(" ");}
		System.out.println("The Transpose of the matrix: ");
		for(int i=0;i<sz;i++)
		{
			for(int j=0;j<sz;j++)
			{
				Arr1[i][j]=Arr[j][i];
			}
		}
				for(int i=0;i<sz;i++)
				{
			for(int j=0;j<sz;j++)
			{
				System.out.print(Arr1[i][j] + "	");
			}
			System.out.println(" ");
```

```
}
int flag=0;
for(int i=0;i<sz;i++)
{
        for(int j=0;j<sz;j++)
        {
                if(Arr[i][j] != Arr1[i][j])
                {
                        flag=1;
                        break;}}}
if(flag==1)
{
        System.out.println("The matrix is not symmetric");
}
else
{
        System.out.println("The matrix is symmetric");
}       }}
```

Output Screenshot



Result

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

Experiment No.: 5

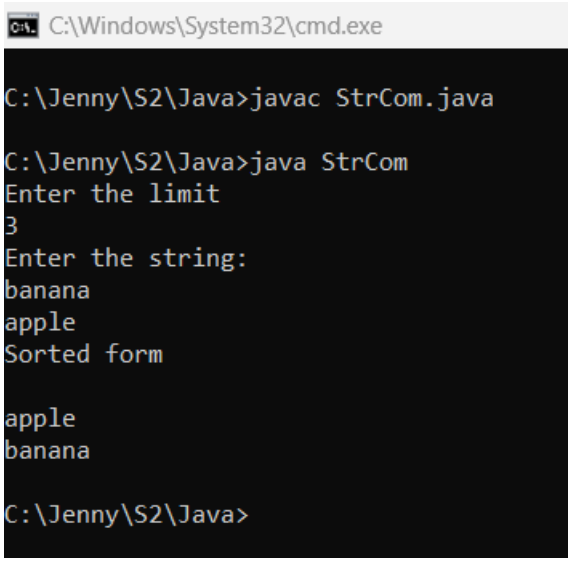
Aim: Program to Sort strings

CO2: Familiarization and understanding of arrays and strings

Procedure

```
import java.util.*;
public class StrCom
{
public static void main(String[] args)
{
        String temp;

Scanner sm=new Scanner(System.in);
System.out.println("Enter the limit");
int l=sm.nextInt();
System.out.println("Enter the string: ");
String a[]=new String[l];
for(int i=0;i<l;i++)
{
        a[i]=sm.nextLine();
}
for(int i=0;i<l;i++)
{
for(int j=i+1;j<l;j++)
{
        if(a[i].compareTo(a[j])>0)
        {
           temp=a[i];
                a[i]=a[j];
                a[j]=temp;
        }
}
}
System.out.println("Sorted form");
for(int i=0;i<l;i++)
{
        System.out.println(a[i]);
}
}
}
```

Output Screenshot



Result

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

Experiment No.: 6

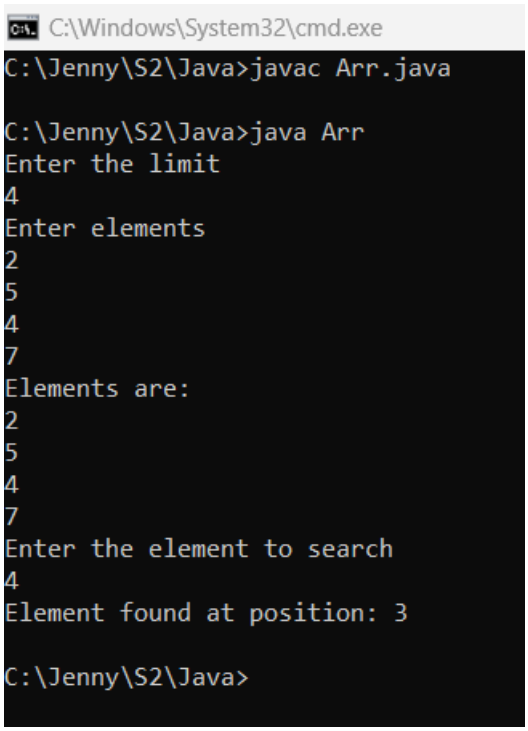
Aim: Search an element in an array.

CO2: Familiarization and understanding of arrays and strings

Procedure

```
import java.util.*;
public class Arr
{
public static void main(String[] args)
{
int i,temp;
Scanner aro=new Scanner(System.in);
System.out.println("Enter the limit");
int n=aro.nextInt();
int arr[]=new int[5];

System.out.println("Enter elements");

for(i=0;i<n;i++)
{
        arr[i]=aro.nextInt();
}
System.out.println("Elements are: ");
for(i=0;i<n;i++)
{
        System.out.println(arr[i]);
}
System.out.println("Enter the element to search");
int el=aro.nextInt();
for(i=0;i<n;i++)
{
        if(el==arr[i])
        {
                System.out.println("Element found at position: "+(i+1));
                break;
        }
}

}
}
```

Output Screenshot



Result

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

Experiment No.: 7

Aim: Perform string manipulations

CO2: Familiarization and understanding of arrays and strings

Procedure

```
import java.util.*;
public class StringM{
public static void main(String[] args){
Scanner sm = new Scanner(System.in);
System.out.println("Enter the string: ");
String a=sm.nextLine();
System.out.println("Enter the second string: ");
String b=sm.nextLine();
System.out.println("Concatination of strings: "+a.concat(b));
System.out.println("lenth:"+a.length());
System.out.println("Length:"+b.length());
System.out.println("Uppercase: "+a.toUpperCase());
System.out.println("Lowercase: "+a.toLowerCase());
System.out.println("Uppercase: "+b.toUpperCase());
System.out.println("Lowercase: "+b.toLowerCase());
System.out.println("Strings equal? "+a.equals(b));
System.out.println("Substring from first string: "+a.substring(3));
System.out.println("Substring from first string: "+b.substring(3));
}}
```

Output Screenshot

```
C:\Jenny\S2\Java>javac StringM.java

C:\Jenny\S2\Java>java StringM
Enter the string:
jenny
Enter the second string:
jenny
Concatination of strings: jennyjenny
lenth:5
Length:5
Uppercase: JENNY
Lowercase: jenny
Uppercase: JENNY
Lowercase: jenny
Strings equal? true
Substring from first string: ny
Substring from first string: ny
```

Result

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

Experiment No.: 8

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.

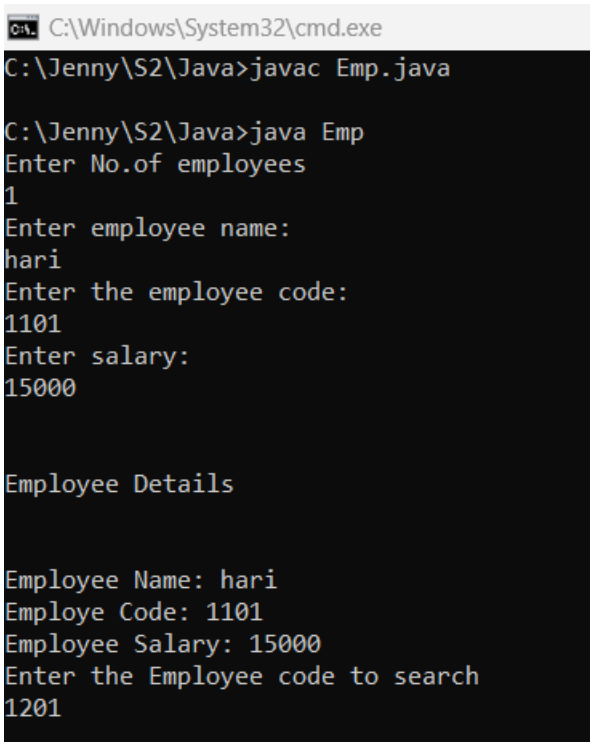
CO2: Familiarization and understanding of arrays and strings

Procedure

```
import java.util.*;
public class Emp
{
int eno,salary;
String ename;
public void get()
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter employee name: ");
ename=sc.next();
System.out.println("Enter the employee code: ");
eno=sc.nextInt();
System.out.println("Enter salary: ");
salary=sc.nextInt();
}
public void display()
{
            System.out.println("\n");
System.out.println("Employee Details");
  System.out.println("\n");
System.out.println("Employee Name: "+ename);
System.out.println("Employe Code: "+eno);
System.out.println("Employee Salary: "+salary);
}
 public static void main(String[] args)
{
  Scanner sc =new Scanner(System.in);
  System.out.println("Enter No.of employees");
  int n=sc.nextInt();
  Emp e[]=new Emp[n];
  for(int i=0;i<n;i++)
  {
  e[i]=new Emp();
  e[i].get();
  }
  for(int i=0;i<n;i++)
  {
  e[i].display();
  }
```

```
 System.out.println("Enter the Employee code to search");
 int v=sc.nextInt();
 for(int i=0;i<n;i++)
 {
 if(e[i].eno==v)
 {
          System.out.println("Employee found");
          e[i].display();
 }
 }
}
}
```

Output Screenshot



Result

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

Experiment No.: 9

Aim: Area of different shapes using overloaded functions

CO3: Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

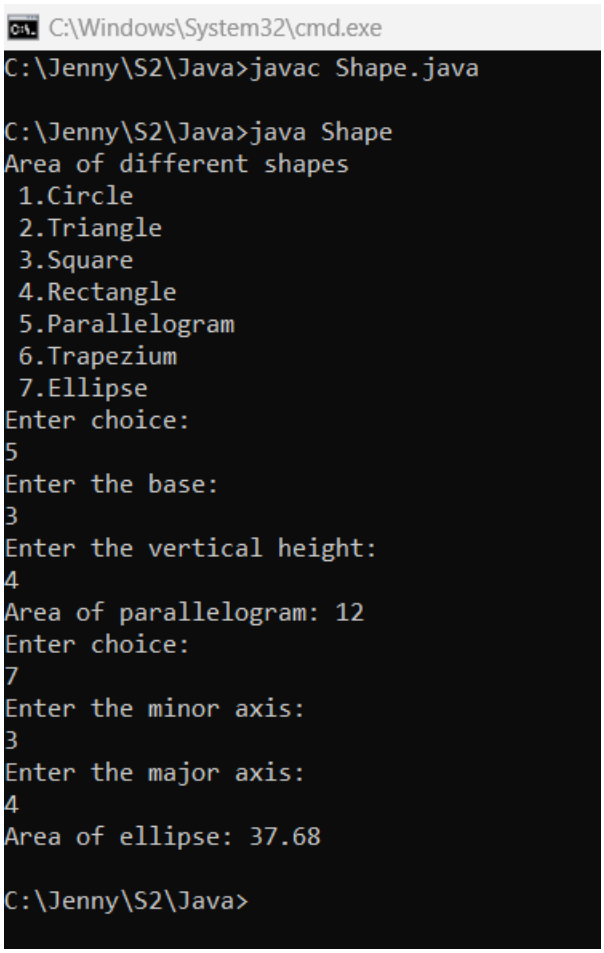
Procedure

```
import java.util.*;
class Shape{
        int r,h,a,p1,p2,t2,t3;
        double b,l,w,t1;
public void area(double r){
double c=3.14*(r*r);
System.out.println("Area of circle: "+c);
}
public void area(double b,int h){
        double t=(b*h)/2;
        System.out.println("Area of triangle: "+t);
}
public void area(int a){
int s=a*a;
System.out.println("Area of square: "+s);
}
public void area(double l,double w){
        double r=l*w;
        System.out.println("Area of rectangle: "+r);
}
public void area(int p1,int p2){
        int p=p1*p2;
        System.out.println("Area of parallelogram: "+p);
}
public void area(double t1,int t2,int t3){
        double tr=0.5*(t2+t3)*t1;
        System.out.println("Area of trapezium: "+tr);
}
public void area(float e1,float e2){
double c=3.14*(e1*e2);
System.out.println("Area of ellipse: "+c);
}
public static void main(String[] args){
int ch;
   Scanner s=new Scanner(System.in);
        Shape sh =new Shape();
        System.out.println("Area of different shapes"+"\n"+" 1.Circle"+"\n"+"
2.Triangle"+"\n"+" 3.Square "+"\n"+" 4.Rectangle "+"\n"+" 5.Parallelogram "+"\n"+"
6.Trapezium"+"\n"+" 7.Ellipse");
        do{
```

```
System.out.println("Enter choice: ");
ch=s.nextInt();
switch(ch){
        case 1:{
System.out.println("Enter the radius: ");
double r=s.nextDouble();
sh.area(r);
        }
        break;
        case 2:{
System.out.println("Enter the breadth: ");
double b=s.nextDouble();
System.out.println("Enter the height: ");
int h=s.nextInt();
sh.area(b,h);
        }
        break;
        case 3:{
System.out.println("Enter the length: ");
int a=s.nextInt();
sh.area(a);
        }
        break;
        case 4:{
System.out.println("Enter the length: ");
double l=s.nextDouble();
System.out.println("Enter the breadth: ");
double w=s.nextDouble();
sh.area(l,w);
        }
        break;
        case 5:{
System.out.println("Enter the base: ");
int p1=s.nextInt();
System.out.println("Enter the vertical height: ");
int p2=s.nextInt();
sh.area(p1,p2);
        }
        break;
        case 6:{
System.out.println("Enter the height of trapezium: ");
double t1=s.nextDouble();
System.out.println("Enter the length of parallel side: ");
int t2=s.nextInt();
System.out.println("Enter the length of parallel side: ");
int t3=s.nextInt();
sh.area(t1,t2,t3);
        }
        break;
        case 7:{
```

```
        System.out.println("Enter the minor axis: ");
        float e1=s.nextFloat();
        System.out.println("Enter the major axis: ");
        float e2=s.nextFloat();
        sh.area(e1,e2);
                }
                break;
        }
        }
        while(ch!=7);
}
}
```

Output Screenshot



Result

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

Experiment No.: 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.

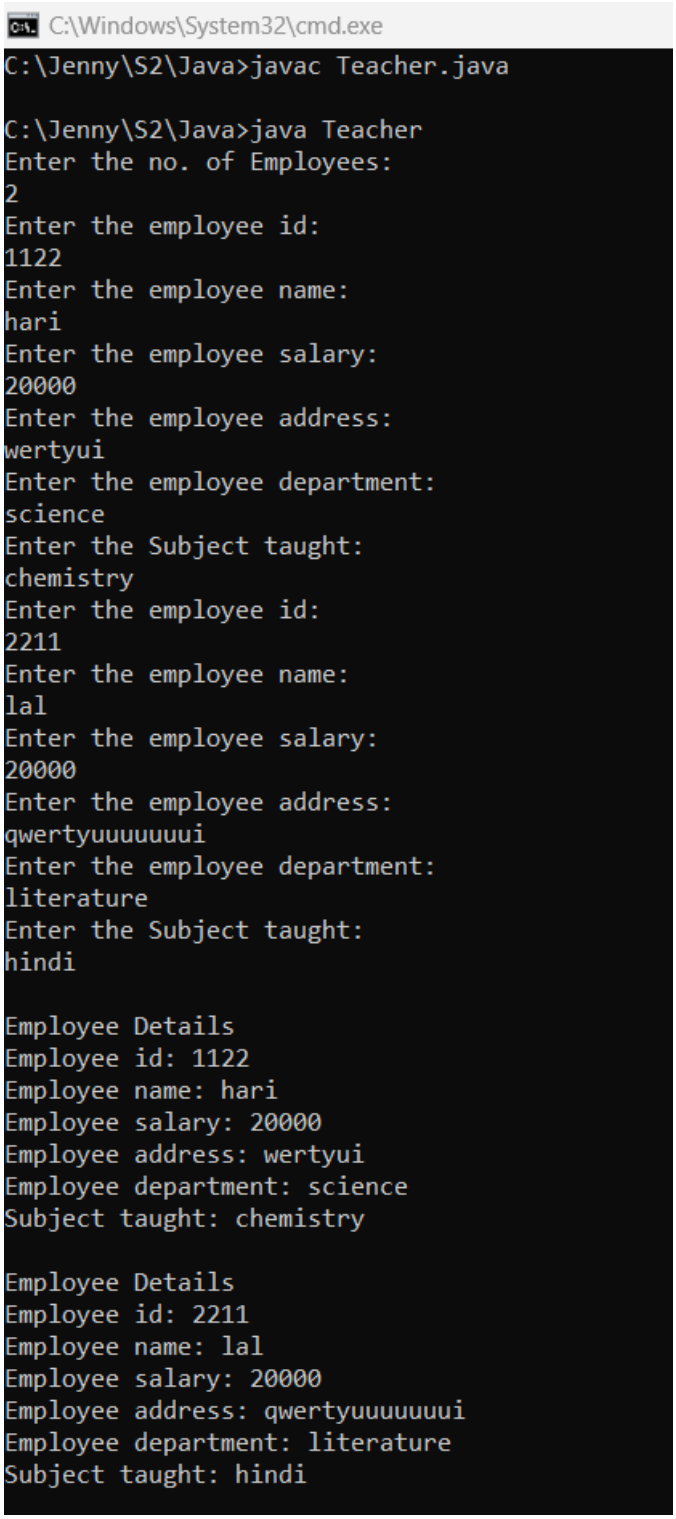
CO3: Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
class Employee
{
	int eid;
	String ename;
	int esal;
	String eadd;
	Employee(int id, String name, int salary , String add)
	{
		eid=id;
		ename=name;
		esal=salary;
		eadd=add;
	}
}
class Teacher extends Employee
{
	String dep;
	String sub;
	Teacher(int id, String name, int salary , String add,String dept, String subt)
	{
		super(id,name,salary,add);
		dep=dept;
		sub=subt;
	}

	void show()
	{
		System.out.println();
		System.out.println("Employee Details");
		System.out.println("Employee id: "+eid);
		System.out.println("Employee name: "+ename);
		System.out.println("Employee salary: "+esal);
		System.out.println("Employee address: "+eadd);
		System.out.println("Employee department: "+dep);
```

```
		System.out.println("Subject taught: "+sub);
	}


	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 id=sc.nextInt();
			System.out.println("Enter the employee name:");
			String name= sc.next();
			System.out.println("Enter the employee salary:");
			int salary=sc.nextInt();
			System.out.println("Enter the employee address:");
			String add=sc.next();
			System.out.println("Enter the employee department:");
			String dept=sc.next();
			System.out.println("Enter the Subject taught:");
			String subt=sc.next();
			t[i]= new Teacher(id,name,salary,add,dept,subt);

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

Output Screenshot



Result

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

Experiment No.: 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.

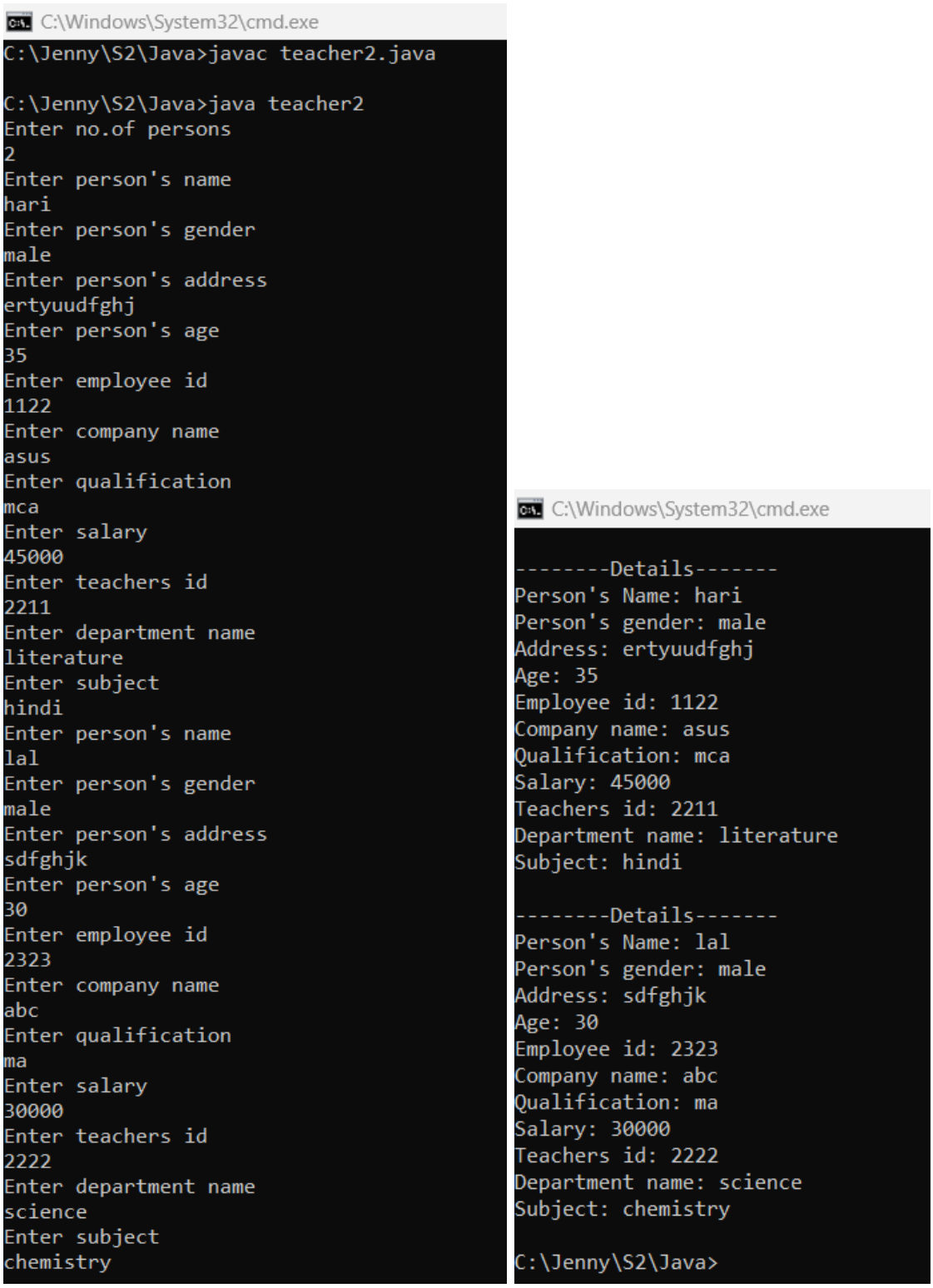
CO3: Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
class person
{
        int age;
        String name,gender,add;
         person()
        {
                Scanner sc =new Scanner(System.in);
                System.out.println("Enter person's name");
                name=sc.next();
                System.out.println("Enter person's gender");
                gender=sc.next();
                System.out.println("Enter person's address");
                add=sc.next();
                System.out.println("Enter person's age");
      age=sc.nextInt();
        }
}
class employee extends person
{
        int id,sal;
        String company,quali;
 employee()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter employee id");
      id=sc.nextInt();
                System.out.println("Enter company name");
                company=sc.next();
                System.out.println("Enter qualification");
                quali=sc.next();
                System.out.println("Enter salary");
      sal=sc.nextInt();
```

```
        }
}
class teacher2 extends employee
{
        int tid;
        String dep,sub;
        teacher2()
        {
    Scanner sc=new Scanner(System.in);
                System.out.println("Enter teachers id");
     tid=sc.nextInt();
                System.out.println("Enter department name");
                dep=sc.next();
                System.out.println("Enter subject");
                sub=sc.next();
        }
        void display()
        {
            System.out.println();
                 System.out.println("--------Details-------");
                System.out.println("Person's Name: "+name);
                System.out.println("Person's gender: "+gender);
                System.out.println("Address: "+add);
                System.out.println("Age: "+age);
                System.out.println("Employee id: "+id);
                System.out.println("Company name: "+company);
                System.out.println("Qualification: "+quali);
                System.out.println("Salary: "+sal);
                System.out.println("Teachers id: "+tid);
                System.out.println("Department name: "+dep);
                System.out.println("Subject: "+sub);
        }
        public static void main(String[] args)
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter no.of persons");
                int n=sc.nextInt();
     teacher2 t[]=new teacher2[n];
                for(int i=0;i<n;i++)
                {
                        t[i]=new teacher2();
                }
                for(int i=0;i<n;i++)
                {
      t[i].display();
                }
        }
}
```

Output Screenshot



Result

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

Experiment No.: 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.

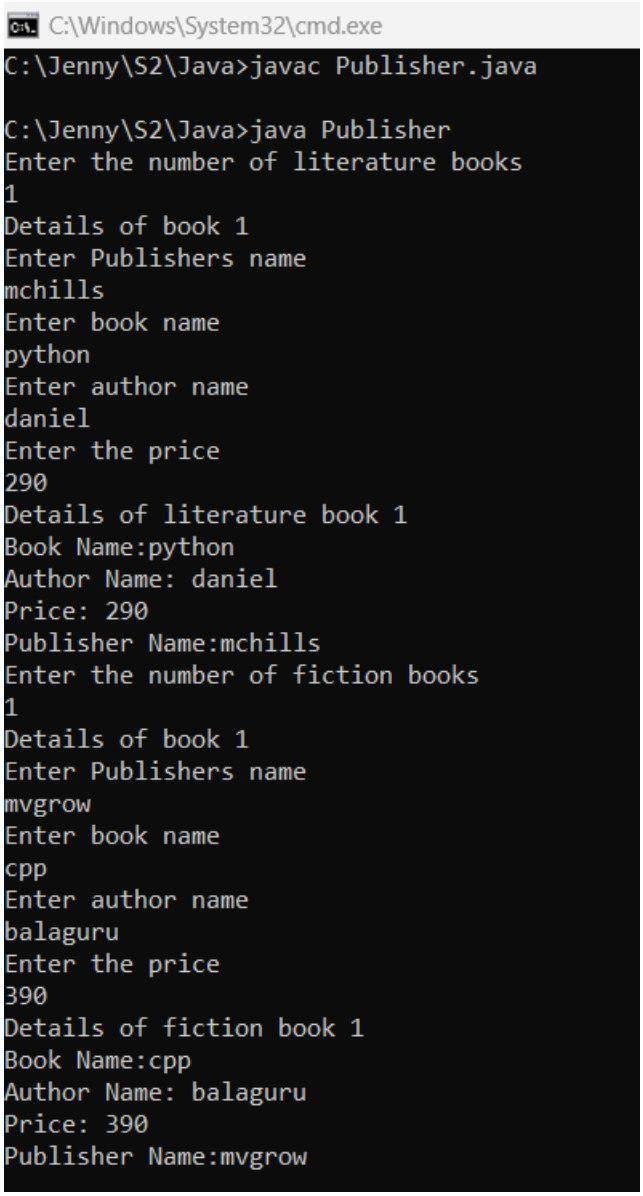
CO3: Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
class Publisher
{
	String pname;
	Publisher()
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter Publishers name");
		pname=sc.next();
	}
	public static void main(String[] args)
	{
		int i,n,n1;
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter the number of literature books");
		n=sc.nextInt();
		literature[] l=new literature[n];
		for(i=0;i<n;i++)
		{
			System.out.println("Details of book "+(i+1));
		l[i]=new literature();
		}
		for(i=0;i<n;i++)
		{
			System.out.println("Details of literature book "+(i+1));
		l[i].ldisp();
		}
		System.out.println("Enter the number of fiction books");
		n1=sc.nextInt();
		fiction[] f=new fiction[n1];
		for(i=0;i<n1;i++)
		{
			System.out.println("Details of book "+(i+1));
		f[i]=new fiction();
		}
		for(i=0;i<n1;i++)
		{
			System.out.println("Details of fiction book "+(i+1));
		f[i].fdisp();
```

```
                }

        }
}
class book extends Publisher
{
        String author,name;
        int price;
        book()
        {
           Scanner sc=new Scanner(System.in);
                System.out.println("Enter book name");
                name=sc.next();
                System.out.println("Enter author name");
                author=sc.next();
                System.out.println("Enter the price");
                price=sc.nextInt();
        }
}
class literature extends book
{
         void ldisp()
         {
                System.out.println("Book Name:"+name);
                System.out.println("Author Name: "+author);
                System.out.println("Price: "+price);
                System.out.println("Publisher Name:"+pname);
         }
}
class fiction extends book
{

         void fdisp()
         {
                System.out.println("Book Name:"+name);
      System.out.println("Author Name: "+author);
                System.out.println("Price: "+price);
      System.out.println("Publisher Name:"+pname);
         }
}
```

Output Screenshot



Result

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

Experiment No.: 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.

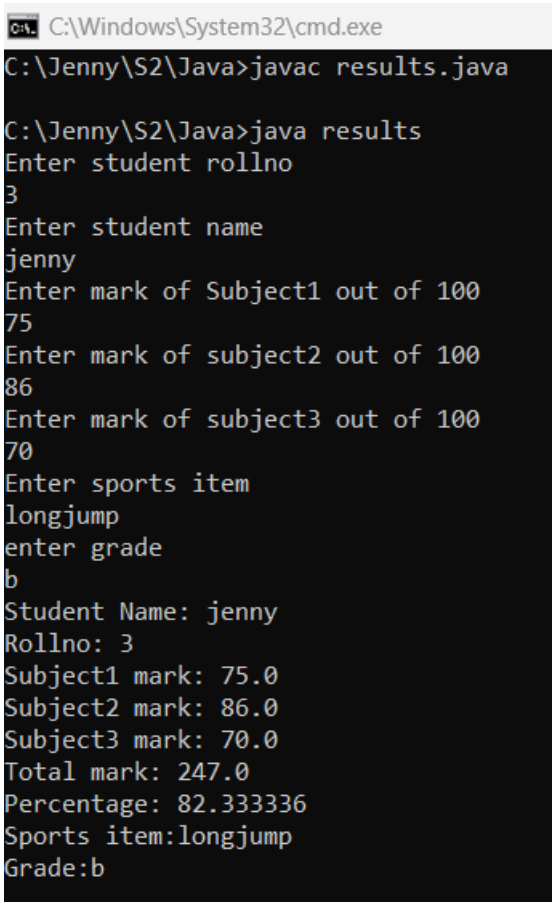
CO3: Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
interface student
{
        public void get();
        public void disp();
}
interface sports
{
        public void get1();
        public void disp1();
}
class results implements student,sports
{
        float t,p;
        float m1,m2,m3;
        int rno;
        String name,sport,grade;
        public void get()
        {
                Scanner sc =new Scanner(System.in);
                System.out.println("Enter student rollno");
                rno=sc.nextInt();
                System.out.println("Enter student name");
      name=sc.next();
                System.out.println("Enter mark of Subject1 out of 100");
                m1=sc.nextFloat();
                System.out.println("Enter mark of subject2 out of 100");
                m2=sc.nextFloat();
                System.out.println("Enter mark of subject3 out of 100");
                m3=sc.nextFloat();
        }
        public void get1()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter sports item");
                sport=sc.next();
                System.out.println("enter grade");
                grade=sc.next();
        }
```

```
        void cal()
        {
        t=m1+m2+m2;
        p=(t/300)*100;
        }
        public void disp()
        {

                System.out.println("Student Name: "+name);
                System.out.println("Rollno: "+rno);
                System.out.println("Subject1 mark: "+m1);
                System.out.println("Subject2 mark: "+m2);
                System.out.println("Subject3 mark: "+m3);
                System.out.println("Total mark: "+t);
                System.out.println("Percentage: "+p);
        }
        public void disp1()
        {
        System.out.println("Sports item:"+sport);
        System.out.println("Grade:"+grade);
        }


public static void main(String[] args)
        {
      results s=new results();
      s.get();
      s.get1();
      s.cal();
      s.disp();
      s.disp1();
        }
}
```

Output Screenshot



Result

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

Experiment No.: 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.

CO3: Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
interface proto{
        public void area();
        public void perimeter();
        public void get();
}
class rect implements proto{
        int l,b;
        double r,peri;
        public void get(){
         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(){
         r=l*b;
         System.out.println("Area of rectangle: "+r);
        }
        public void perimeter(){
                peri=(l+b)*2;
                System.out.println("Perimeter of rectangle: "+peri);
        }}
 class circle implements proto{
         double c,pe,r;
         public void get(){
         Scanner sc=new Scanner(System.in);
         System.out.println("Enter the radius");
         r=sc.nextInt();
         }
         public void area(){
         c=3.14*(r*r);
          System.out.println("Area of circle: "+c);
         }
         public void perimeter(){
                 pe=2*(3.14*(r*r));
                 System.out.println("Perimeter of circle: "+pe);
```

```
        }
public static void main(String[] args){
        int c;
        Scanner sc=new Scanner(System.in);
         rect re=new rect();
    circle ci=new circle();
        do{
                System.out.println("1.Rectangle"+"\n"+"2.Circle"+"\n"+"3.Exit");
                System.out.println("Enter tour choice");
                c=sc.nextInt();
                switch(c)
                {
                        case 1:
                         re.get();
                         re.area();
                         re.perimeter();
                         break;
                         case 2:
                         ci.get();
                         ci.area();
                         ci.perimeter();
                         break;
                }
        }
        while(c!=2);
}
}
```

Output Screenshot

```
C:\Jenny\S2\Java>javac circle.java

C:\Jenny\S2\Java>java circle
1.Rectangle
2.Circle
3.Exit
Enter your choice
2
Enter the radius
3
Area of circle: 28.26
Perimeter of circle: 56.52
```

Result

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

Experiment No.: 15

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 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(".....");
System.out.println("ID    NAME    QUANTITY    PRICE");
for(i=0;i<n;i++)
        {
                b[i].disp();
                net=net+b[i].total;
        }
        System.out.println("          Net Amount: "+net);
}
}
```

Output Screenshot

```
C:\Jenny\S2\Java>javac bill.java

C:\Jenny\S2\Java>java bill
Enter order number
2201
Enter order date
1/3/2023
Enter no.of products
2
Enter product id
101
Enter product name
A
Enter the quantity of the product
2
Enter product unit price
25
Enter product id
102
Enter product name
B
Enter the quantity of the product
1
Enter product unit price
100
.............BILL.............
Order no.:2201
Date:1/3/2023
.........................
ID      NAME      QUANTITY      PRICE
101      A      2      25.0      50.0
102      B      1      100.0      100.0
                  Net Amount: 150.0
```

Result

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

Experiment No.: 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.

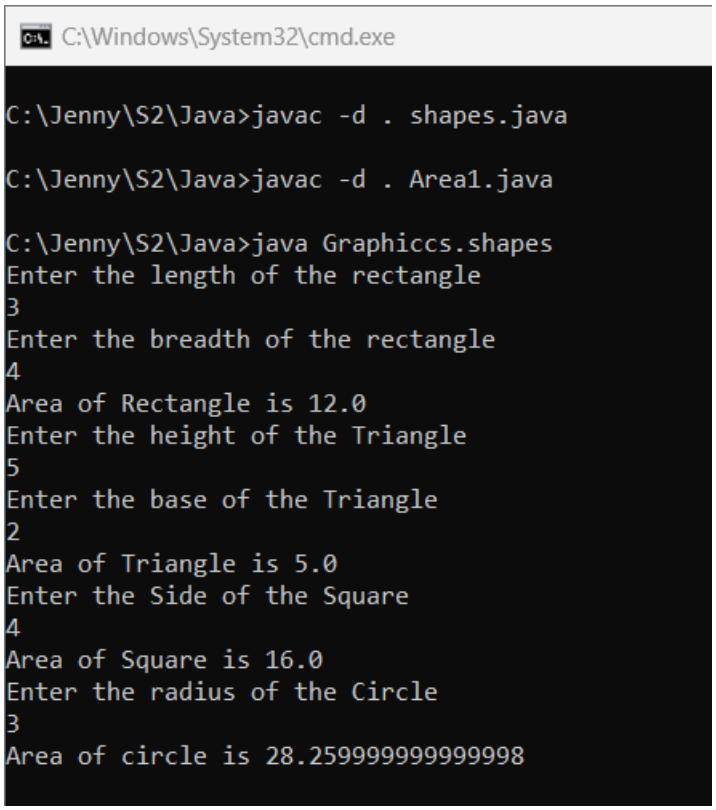
CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
//shapes.java

package Graphiccs;
import java.util.*;

public class shapes implements Area
{
double lr,lb,ra,th,tb,ta,saa,sa,cr,cc;
public void getRect()
{
Scanner ab= new Scanner(System.in);
System.out.println("Enter the length of the rectangle");
lr=ab.nextInt();
System.out.println("Enter the breadth of the rectangle");
lb=ab.nextInt();

}
public void Rectangle()
{
ra=lr*lb;
System.out.println("Area of Rectangle is "+ra);
}
public void getTri()
{
Scanner cb= new Scanner(System.in);
System.out.println("Enter the height of the Triangle");
th=cb.nextInt();
System.out.println("Enter the base of the Triangle");
tb=cb.nextInt();
}
public void Triangle()
{
ta=0.5*th*tb;
System.out.println("Area of Triangle angle is "+ta);
}
public void getSqr()
{
Scanner sq= new Scanner(System.in);
System.out.println("Enter the Side of the Square");
```

```
sa=sq.nextInt();
}
public void Square()
{
saa=sa*sa;
System.out.println("Area of Square is "+saa);
}

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 Graphiccs;
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 getCrl();
}
```

Output Screenshot



Result

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

Experiment No.: 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.

CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

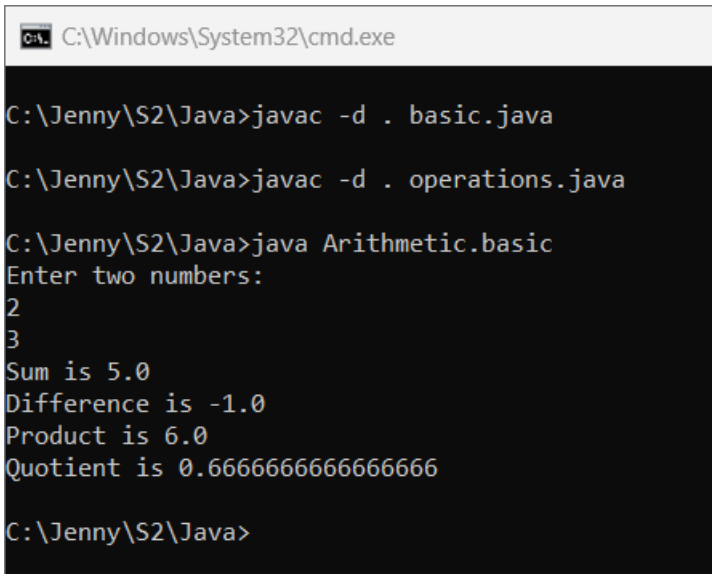
Procedure

```
//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 substract()
{

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.substract();
o.multiply();
o.division();
}
}
```

```
// operations.java

package Arithmetic;
interface operations
{
public void input();
public void add();
public void substract();
public void multiply();
public void division();

}
```

Output Screenshot



Result

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

Experiment No.: 18

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

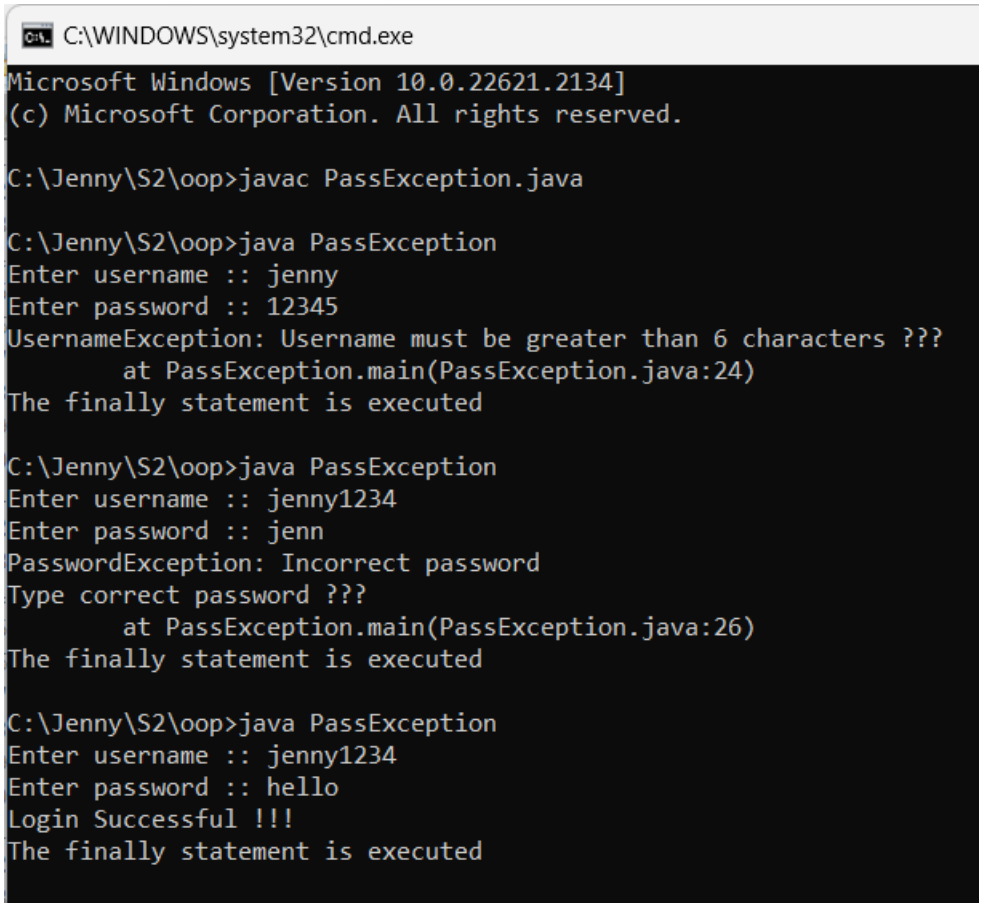
CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

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 PassException {
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



Result

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

Experiment No.: 19

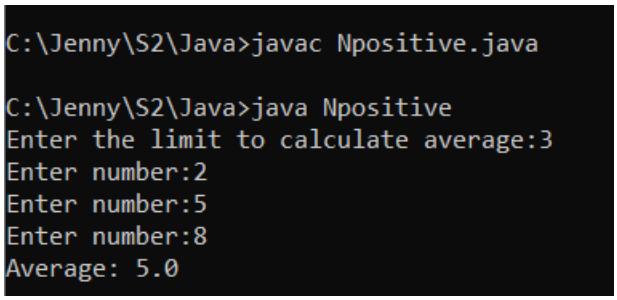
Aim: Find the average of N positive integers, raising a user defined exception for each negative input.

CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.Scanner;
import java.util.InputMismatchException;
public class Npositive{
public static void main(String args[]){
double total = 0, N, userInput;
Scanner input = new Scanner(System.in);
while (true){
System.out.print("Enter the limit to calculate average:");
userInput = input.nextDouble();
if (userInput > 0){
N = userInput;
break;}
else
System.out.println("N must be positive.");}
for (int i = 0; i < N; i++){
while (true){
System.out.print("Enter number:");
try{
userInput = input.nextDouble();
total += userInput;
break;}
catch (InputMismatchException e){
input.nextLine();
System.out.println("Input must bea number. Try again");}}}
System.out.println("Average: "+ total / N);}}
```

Output Screenshot



Result

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

Experiment No.: 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).

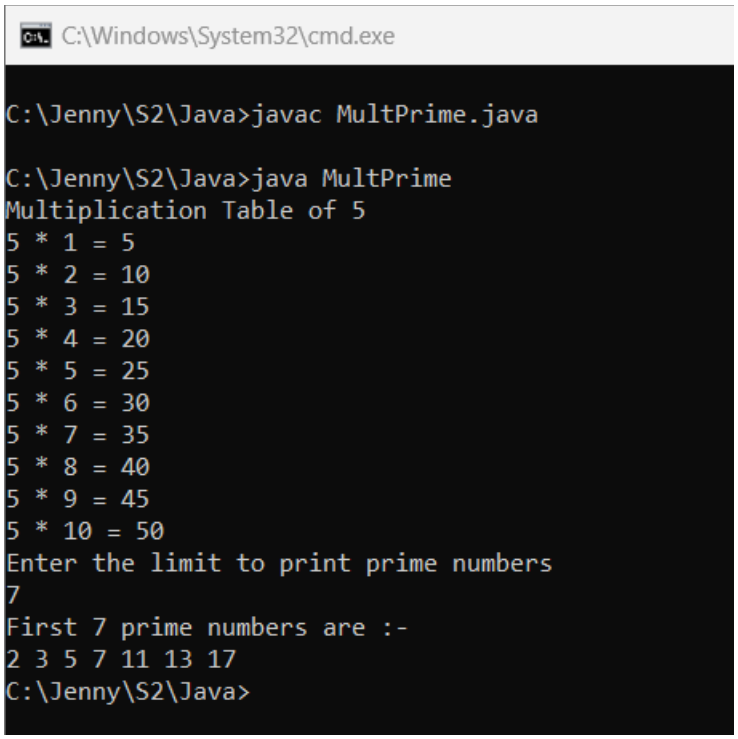
CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.*;
class ThreadA extends Thread
{
public void run( )
{
int n = 5;
for (int i = 1; i <= 10; ++i)
System.out.println(n + " * " + i +
" = " + n * i);
}
}

class ThreadB extends Thread
{
public void run( )
{
Scanner sc = new Scanner(System.in);
int i,n,p,count,flag;

System.out.println("Enter the limit to print prime numbers");
n=sc.nextInt();
System.out.println("First "+n+" prime numbers are :-");

p=2;
i=1;
while(i<=n)
{
flag=1;

for(count=2;count<=p-1;count++)
{
if(p%count==0)
{
flag=0;
break;
}
}
if(flag==1)
{
```

```
System.out.print(p+" ") ;
i++;
}
p++;
}
}
}

public class MultPrime
{
public static void main(String args[]) {
ThreadA a = new ThreadA();
ThreadB b = new ThreadB();
a.start();
b.start();
System.out.println("Multiplication Table of 5");
}
}
```

Output Screenshot



Result

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

Experiment No.: 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).

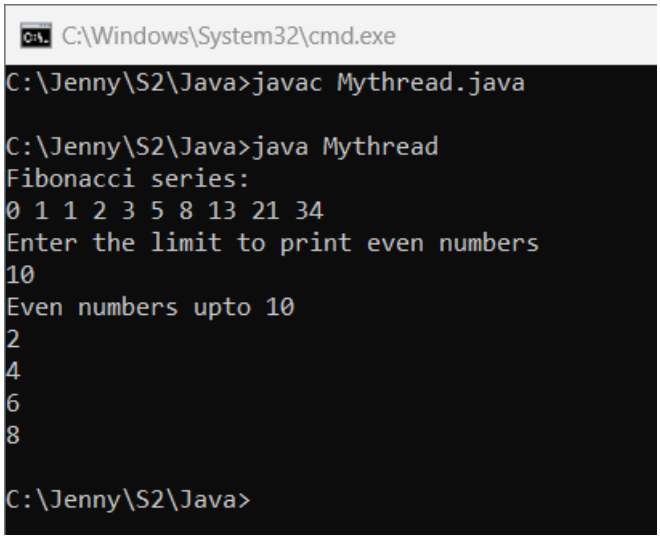
CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.*;
public class Mythread {

public static void main(String[] args) {
Runnable r = new Runnable1();
Thread t = new Thread(r);
t.start();
Runnable r2 = new Runnable2();
Thread t2 = new Thread(r2);
t2.start();
}
}

class Runnable2 implements Runnable{
public void run()
{
        Scanner sc=new Scanner(System.in);
        System.out.println("\n"+"Enter the limit to print even numbers");
        int n=sc.nextInt();
        System.out.println("Even numbers upto "+ n);
for(int i=1;i<n;i++)
{
if(i%2 == 0)
System.out.println(i);
}
}
}

class Runnable1 implements Runnable{
public void run()
{
int n1=0,n2=1,n3,i,count=10;
System.out.println("Fibonacci series: ");
System.out.print(n1+" "+n2);//printing 0 and 1

for(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are already printed
{
n3=n1+n2;
System.out.print(" "+n3);
```

```
n1=n2;
n2=n3;
}
}
}
```

Output Screenshot



Result

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

Experiment No.: 22

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 stack{
        public static void main(String args[]){
                Scanner obj =new Scanner(System.in);
                int ch;
                int top=-1;
                int n;
                System.out.println("Enter the size");
                n=obj.nextInt();
                int s[]=new int[n];
                do{
                System.out.println("1.PUSH ");
                System.out.println("2.POP ");
                System.out.println("3.PEEK ");
                System.out.println("4.DISPLAY ");
                System.out.println("5.IS EMPTY ");
                System.out.println("Select your Option  ");
                ch=obj.nextInt();
                switch(ch){
                        case 1:
                        if(top!=n){
                        System.out.println(" Enter the Element to be    inserted :- ");
                        s[++top]=obj.nextInt();
                        System.out.println("Inserted to the Stack :- ");          }
                        break;
                        case 2:
                        if(top==-1){
                        System.out.println("Stack Under Flow");}
                        else{
                        top=top-1;
                        System.out.println("One Element is deleted");}
                        break;
                        case 3:
                        System.out.println("TOP Of Stack is :" + s[top] );
                        break;
                        case 5:
                        if(top==-1){
                        System.out.println("Stack Is Empty!!");}
                        else{
                        System.out.println("Stack is not Empty !!");}
```

```
                    break;
                    case 4 :
                    if(top==-1){
                    System.out.println("Stack Is Empty!!");}
                    else{
                    System.out.println("Stack Elements are!!");
                    for (int i=0;i<=top;i++){
                    System.out.println(s[i]);}}
                    break;}
                System.out.println("1.Exit ");
                System.out.println("0.Continue");
                ch=obj.nextInt();
                } while(ch==0);
}}
```

Output Screenshot

```
C:\Jenny\S2\Java>javac stack.java

C:\Jenny\S2\Java>java stack
Enter the size
3
1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.IS EMPTY
Select your Option
1
 Enter the Element to be inserted :-
3
Inserted to the Stack :-
1.Exit
0.Continue
0
1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.IS EMPTY
Select your Option
4
Stack Elements are!!
3
```

Result

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

Experiment No.: 23

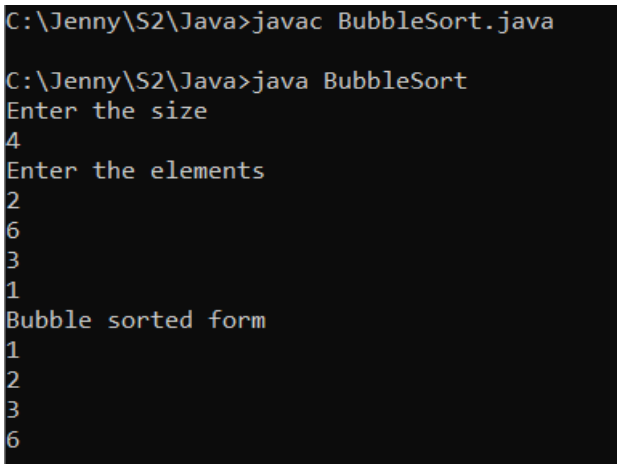
Aim: Using generic method perform Bubble sort.

CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.*;
public class BubbleSort{
	public static void main(String[] args){
		int n,temp;
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter the size");
		n=sc.nextInt();
		int bub[]=new int[n];
		System.out.println("Enter the elements");
		for(int i=0;i<n;i++){
			bub[i]=sc.nextInt();}
			for(int i=0;i<n;i++){
				for(int j=i+1;j<n;j++){
					if(bub[i]>bub[j]){
						temp=bub[i];
						bub[i]=bub[j];
						bub[j]=temp;}}}
			System.out.println("Bubble sorted form");
			for(int i=0;i<n;i++){
					System.out.println(bub[i]);    }}}
```

Output Screenshot



Result

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

Experiment No.: 24

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.*;

public class Array_list
{
	public static void main(String[] args)
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the limit: ");
		int l = sc.nextInt();
		ArrayList<String> color = new ArrayList<String>();

		for(int i=0;i<l;i++)
		{
			System.out.println("Enter color: ");
			color.add(sc.next());
		}
		System.out.println(color);
		System.out.println("First color: "+color.get(0));
		System.out.println("Second color: "+color.get(1));
		System.out.println("After changing second color as blue: ");
		color.set(1,"Blue");
		System.out.println(color);
		//System.out.println(color.get(1));
		System.out.println("After removing 1st element: ");
		color.remove(0);
		System.out.println(color);
		System.out.println("To get the size of the array list: "+color.size());
		/*for(int i=0;i<color.size();i++)
		{
			System.out.println(color.get(i));
		}*/
		System.out.println("Sorting the list in Alphabetical order: ");
		Collections.sort(color);
		System.out.println(color);
		System.out.println("Last element after sorting is  "+Collections.max(color));


	}
}
```

Output Screenshot

```
C:\Jenny>javac Array_list.java

C:\Jenny>java Array_list
Enter the limit:
4
Enter color:
red
Enter color:
black
Enter color:
green
Enter color:
pink
[red, black, green, pink]
First color: red
Second color: black
After changing second color as blue:
[red, Blue, green, pink]
After removing 1st element:
[Blue, green, pink]
To get the size of the array list: 3
Sorting the list in Alphabetical order:
[Blue, green, pink]
Last element after sorting is pink
```

Result

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

Experiment No.: 25

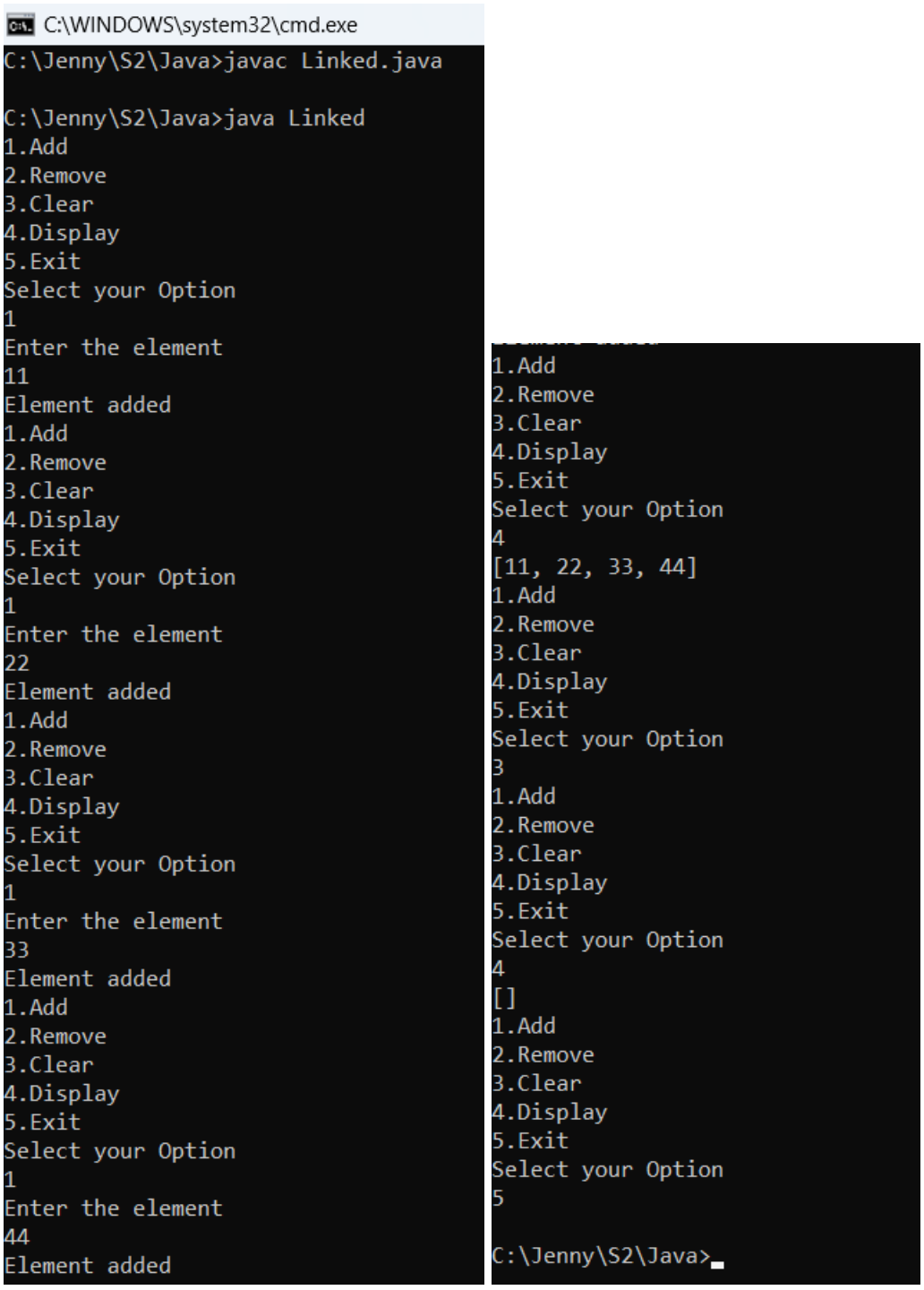
Aim: Program to remove all the elements from a linked list.

CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.*;
        public class Linked
        {
                public static void main (String [] args)
                {
                        LinkedList<String> list = new LinkedList<String>();
                        Scanner obj =new Scanner(System.in);
                        int ch;
                do
                {
                        System.out.println("1.Add”+”\n”+”2.Remove”+”\n”+”
3.Clear”+”\n”+”4.Display”+”\n”+”5.Exit”+”\n”+” Select your Option ");
                                ch=obj.nextInt();
                                switch(ch)
                                {
                                        case 1:
                                                String val;
                                                System.out.println("Enter the element");
                                                val=obj.next();
                                                list.add(val);
                                                System.out.println("Element added");
                                                break;
                                        case 2:
                                           String v;
                                                 System.out.println("element to be removed");
                                                 v=obj.next();
                                                 list.remove(v);
                                                 System.out.println("Element removed");
                                                break;
                                        case 3:
                                                list.clear();
                                                break;
                                        case 4 :
                                                System.out.println(list);
                                                break;
                                }
                }while(ch !=5);n

                }
        }
```

Output Screenshot



Result

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

Experiment No.: 26

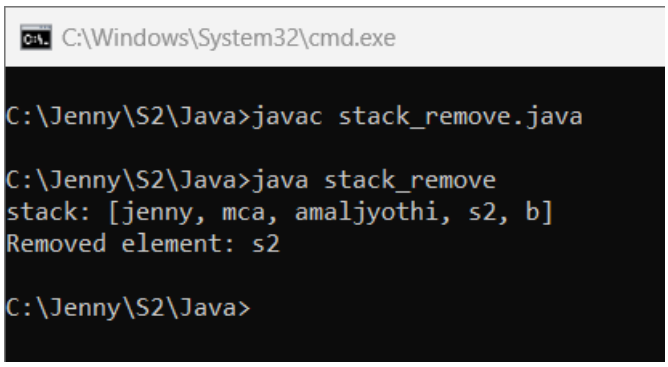
Aim: Program to remove an object from the Stack when the position is passed as parameter.

CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.*;
public class stack_remove
{
	public static void main(String[] args)
	{
		Stack<String> strm = new Stack<String>();
		strm.add("jenny");
		strm.add("mca");
		strm.add("amaljyothi");
		strm.add("s2");
		strm.add("b");
		System.out.println("stack: "+ strm);
		String rm=strm.remove(3);
		System.out.println("Removed element: "+rm);
	}
}
```

Output Screenshot



Result

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

Experiment No.: 27

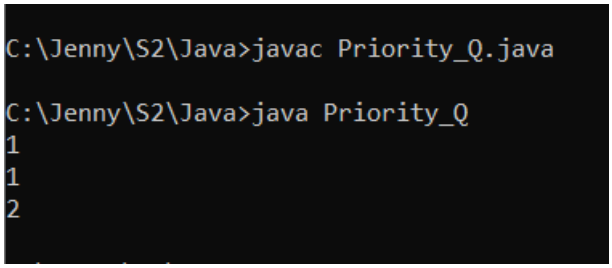
Aim: Program to demonstrate the creation of queue object using the PriorityQueue class.

CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.*;
public class Priority_Q
{
	public static void main(String[] args)
	{
		PriorityQueue<Integer> pq=new PriorityQueue<Integer>();
		pq.add(1);
		pq.add(2);
		pq.add(3);
		System.out.println(pq.peek());
		System.out.println(pq.poll());
		System.out.println(pq.peek());
	}
}
```

Output Screenshot



Result

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

Experiment No.: 28

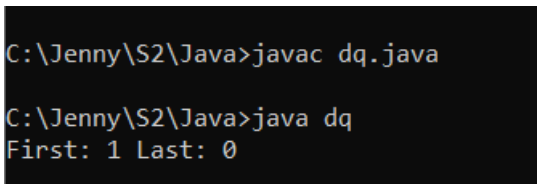
Aim: Program to demonstrate the addition and deletion of elements in deque.

CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.*;
public class dq
{
	public static void main(String[] args)
	{
		Deque<Integer> deq =new ArrayDeque<>();
		deq.addFirst(1);
		deq.addLast(0);
		int first=deq.removeFirst();
		int last=deq.removeLast();
		System.out.println("First: "+first+" Last: "+last);
	}
}
```

Output Screenshot



Result

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

Experiment No.: 29

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.LinkedHashSet;
public class Hash
{

	public static void main(String[] args)
	{

		LinkedHashSet<String> linkedset
			= new LinkedHashSet<String>();
		linkedset.add("A");
		linkedset.add("B");
		linkedset.add("C");
		linkedset.add("D");
		linkedset.add("A");
		linkedset.add("E");

		System.out.println("Size of LinkedHashSet = "+ linkedset.size());

		System.out.println("Original LinkedHashSet:"+ linkedset);

		System.out.println("Removing D from LinkedHashSet: "+
linkedset.remove("D"));

		System.out.println("Trying to Remove Z which is not "+ "present: " +
linkedset.remove("Z"));

		System.out.println("Checking if A is present="+ linkedset.contains("A"));

		System.out.println("Updated LinkedHashSet: "+ linkedset);
	}
}
```

Output Screenshot

```
C:\Jenny\S2\Java>javac Hash.java

C:\Jenny\S2\Java>java Hash
Size of LinkedHashSet = 5
Original LinkedHashSet:[A, B, C, D, E]
Removing D from LinkedHashSet: true
Trying to Remove Z which is not present: false
Checking if A is present=true
Updated LinkedHashSet: [A, B, C, E]
```

Result

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

Experiment No.: 30

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 CompareSet
{
   public static void main(String[] argv)
   {
      Set<String> set1 = new HashSet<String>();
      set1.add("Paul");
      set1.add("Donal");
      set1.add("William");
      set1.add("Johnson");
      set1.add("Emma");
      System.out.println("The values in the 1st set are: "+ set1);
      System.out.println();
      System.out.println("The size of the 1st set is: "+ set1.size());
      Set<String> set2 = new HashSet<String>();
      set2.add("Paul");
      set2.add("Johnson");
      set2.add("Donal");
      set2.add("Emma");
      set2.add("William");
      System.out.println("The values in the 2nd set are: "+ set2);
      System.out.println();
      System.out.println("The size of the 2nd set is: "+ set2.size());
      boolean result = set1.equals(set2);
      if(result) {
         System.out.println("Set1 and Set2 are equal.");
      }else {
         System.out.println("Set1 and Set2 are not equal.");
      }
   }
}
```

Output Screenshot

```
C:\Jenny\S2\Java>javac CompareSet.java

C:\Jenny\S2\Java>java CompareSet
The values in the 1st set are: [Johnson, Donal, William, Paul, Emma]

The size of the 1st set is: 5
The values in the 2nd set are: [Johnson, Donal, William, Paul, Emma]

The size of the 2nd set is: 5
Set1 and Set2 are equal.
```

Result

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

Experiment No.: 31

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.*;
class map {
	public static void main(String args[])
	{
		Map<String, Integer> hm= new HashMap<String, Integer>();
		hm.put("a",10);
		hm.put("b",20);
		hm.put("c", 30);
		hm.put("d", 40);
			System.out.println("map content:"+hm);
			hm.put("a",50);
		   System.out.println("map content after changing value:"+hm);
			hm.remove("c",30);
			System.out.println("map content after removing value:"+hm);
	}
}
```

Output Screenshot

```
C:\Jenny\S2\Java>javac map.java

C:\Jenny\S2\Java>java map
map content:{a=10, b=20, c=30, d=40}
map content after changing value:{a=50, b=20, c=30, d=40}
map content after removing value:{a=50, b=20, d=40}
```

Result

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

Experiment No.: 32

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.stream.*;
public class HashToTree
{
  public static void main(String args[])
  {
    Map<String, String> map = new HashMap<>();
    map.put("1", "One");
    map.put("2", "Two");
    map.put("3", "Three");
    map.put("4", "Four");
    map.put("5", "Five");
    map.put("6", "Six");
    map.put("7", "Seven");
    map.put("8", "Eight");
    map.put("9", "Nine");
    System.out.println("HashMap = " + map);
    Map<String, String> treeMap = new TreeMap<>();
    treeMap.putAll(map);
    System.out.println("TreeMap = " + treeMap);
  }
}
```

Output Screenshot

```
C:\Jenny\S2\Java>javac HashToTree.java

C:\Jenny\S2\Java>java HashToTree
HashMap = {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}
TreeMap = {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}
```

Result

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

Experiment No.: 33

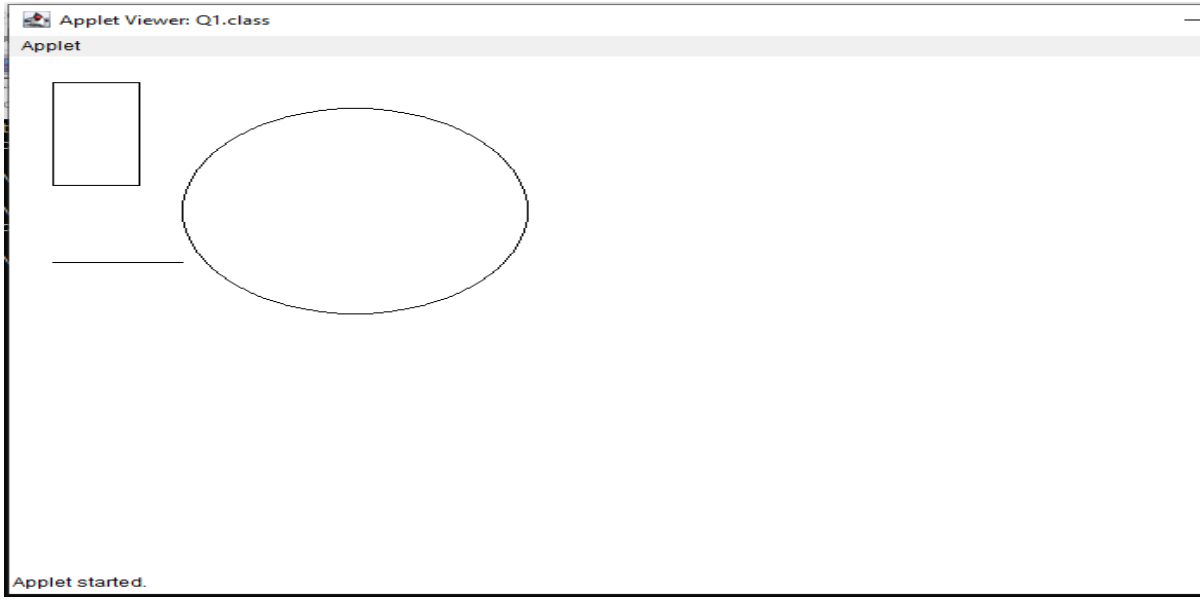
Aim: Program to draw Circle, Rectangle, Line in Applet.

CO5: Develop applications to handle events using applets

Procedure

```
import java.awt.*;
import java.applet.*;
public class Q1 extends Applet{
        public void paint(Graphics g){

                g.drawOval(100,50,200,200);
                g.drawRect(25,25,50,100);
                g.drawLine(25,200,100,200);
        }}
/*<html>
<head></head>
<body>
<APPLET CODE="Q1.class" WIDTH="800" HEIGHT="500"></APPLET>
</body>
</html>*/
```

Output Screenshot



Result

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

Experiment No.: 34

Aim: Program to find maximum of three numbers using AWT.

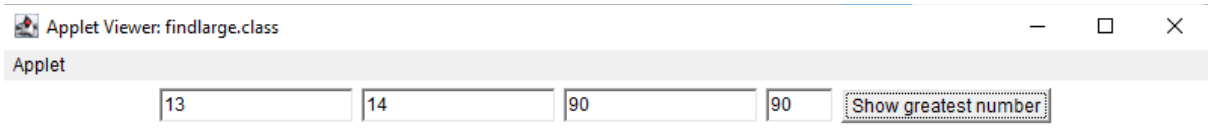
CO5: Develop applications to handle events using applets

Procedure

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class findlarge extends Applet implements ActionListener
{
TextField t1,t2,t3,t4;
Button b1;
public void init()
{
t1=new TextField(15);
t1.setBounds(100,25,50,20);
t2=new TextField(15);
t2.setBounds(100,25,50,20);
t3=new TextField(15);
t3.setBounds(100,25,50,20);
t4=new TextField("Ans");
t4.setBounds(175,50,50,20);
b1= new Button("Show greatest number");
b1.setBounds(175,65,50,40);
add(t1);
add(t2);
add(t3);
add(t4);
add(b1);
b1.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
int i,j,k;
i=Integer.parseInt(t1.getText());
j=Integer.parseInt(t2.getText());
k=Integer.parseInt(t3.getText());
if(i<j)
{
if(j<k)
t4.setText(""+k);
else
t4.setText(""+j);
}
else
t4.setText(""+i);
```

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

Output Screenshot



Result

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

Experiment No.: 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.

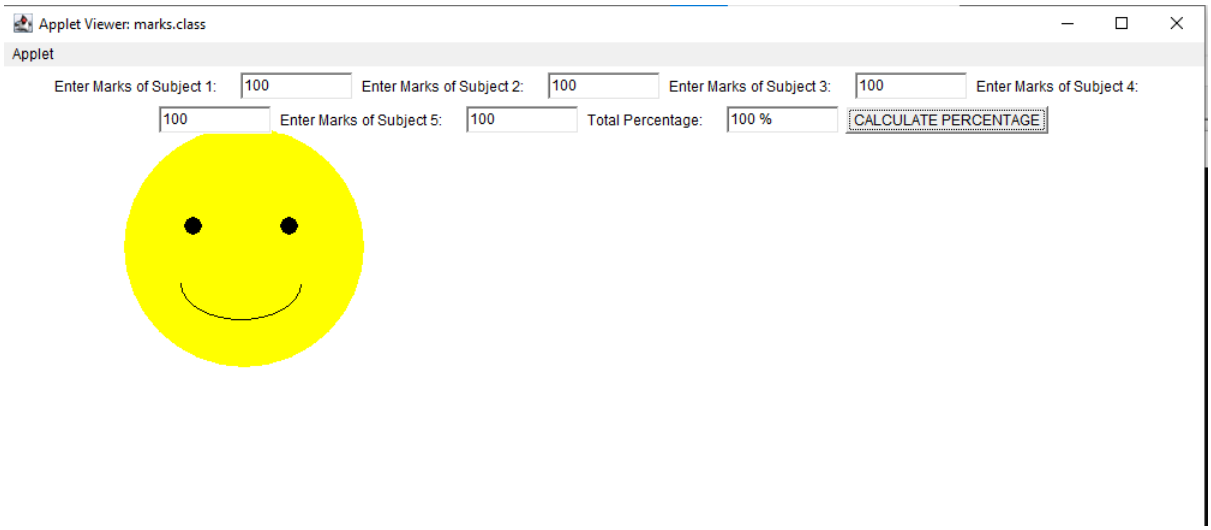
CO5: Develop applications to handle events using applets

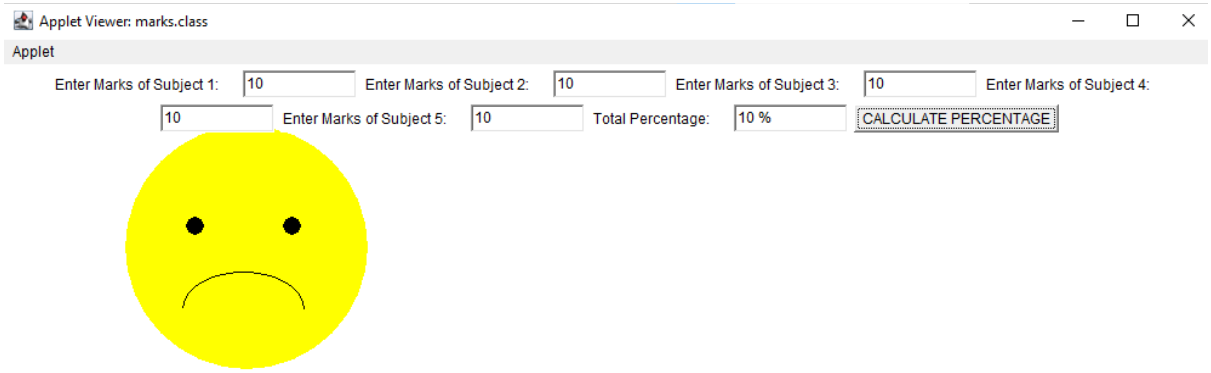
Procedure

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class marks extends Applet implements ActionListener
{
public int per =0;
Label l1 = new Label("Enter Marks of Subject 1: ");
Label l2 = new Label("Enter Marks of Subject 2: ");
Label l3 = new Label("Enter Marks of Subject 3: ");
Label l4 = new Label("Enter Marks of Subject 4: ");
Label l5 = new Label("Enter Marks of Subject 5: ");
Label l6 = new Label("Total Percentage: ");

TextField t1 = new TextField(10);
TextField t2 = new TextField(10);
TextField t3 = new TextField(10);
TextField t4 = new TextField(10);
TextField t5 = new TextField(10);
TextField t6 = new TextField(10);
Button b1 = new Button("CALCULATE PERCENTAGE");
public marks()
{
add(l1);
add(t1);
add(l2);
add(t2);
add(l3);
add(t3);
add(l4);
add(t4);
add(l5);
add(t5);
add(l6);
add(t6);
add(b1);
b1.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
int m1 = Integer.parseInt(t1.getText());
```

```
int m2= Integer.parseInt(t2.getText());
int m3= Integer.parseInt(t3.getText());
int m4= Integer.parseInt(t4.getText());
int m5= Integer.parseInt(t5.getText());

if(e.getSource()==b1)
{
int add=m1+m2+m3+m4+m5;
per=add/5;
t6.setText(String.valueOf(per)+" %");

repaint();
}

}
public void paint(Graphics g)
{
if(per>=50)
{
        g.setColor(Color.yellow);
        g.fillOval(100,50,200,200);
        g.setColor(Color.black);
        g.fillOval(150,125,15,15);
        g.fillOval(230,125,15,15);
        g.drawArc(147,150,100,60,0,-180);
}
else if(per>0 && per<50)
{
   g.setColor(Color.yellow);
        g.fillOval(100,50,200,200);
        g.setColor(Color.black);
        g.fillOval(150,125,15,15);
        g.fillOval(230,125,15,15);
        g.drawArc(147,170,100,60,0,180);
}
}
public static void main(String args[])
{
new marks();
}
}

/*<html><head>
</head>
<body><div align="center">
<applet code="marks.class"width="1000"height="1000">
</applet></div>
</body></html>*/
```

Output Screenshot





Result

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

Experiment No.: 36

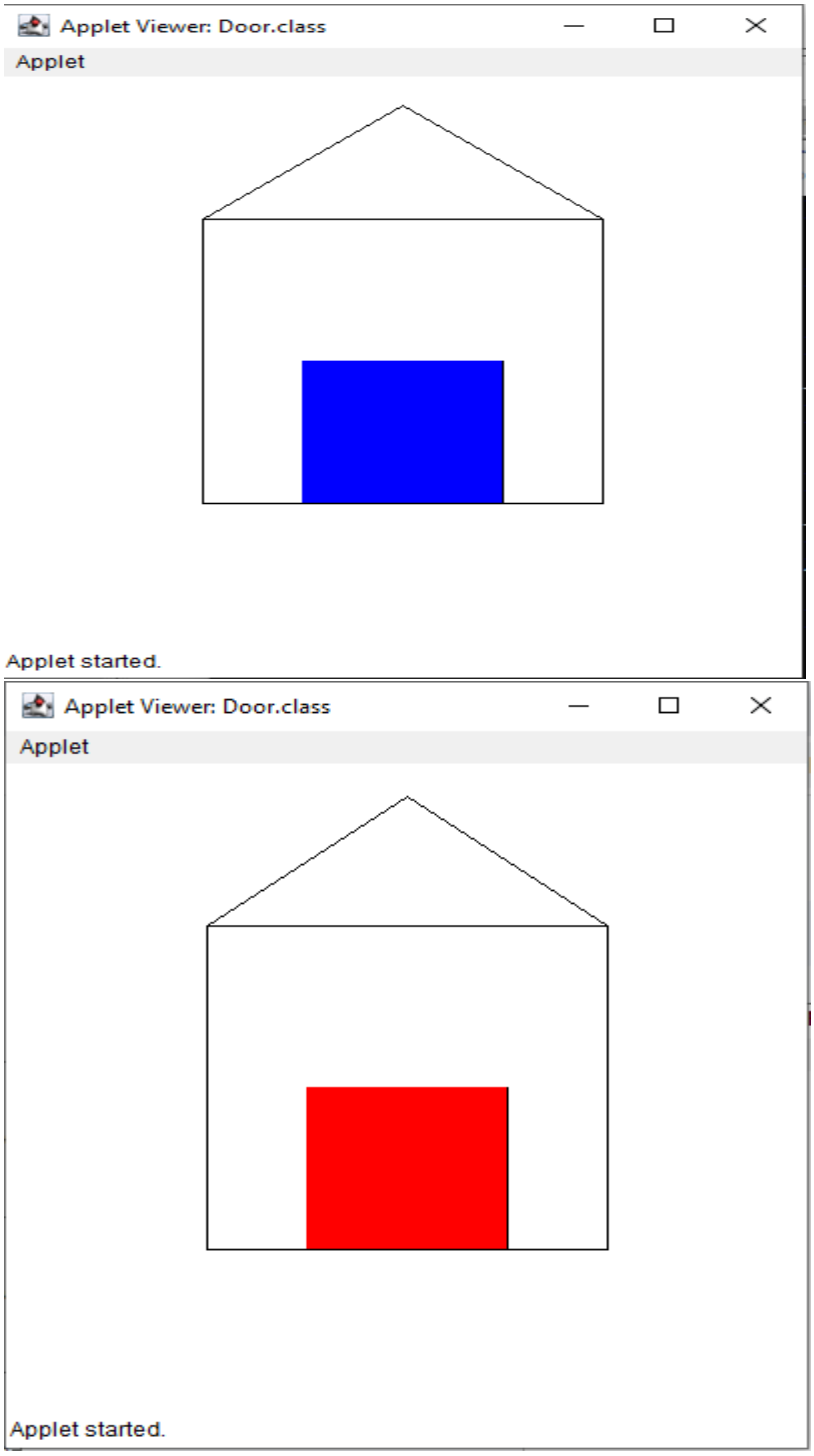
Aim: Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

CO5: Develop applications to handle events using applets

Procedure

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.*;
public class Door extends Applet{
private boolean isDoorBlue = true;
public void init() {
setSize(400, 400);
addMouseListener(new MouseAdapter() {
public void mouseClicked(MouseEvent e) {
isDoorBlue = !isDoorBlue;
repaint();
}});
}
public void paint(Graphics g) {
g.setColor(Color.BLACK);
g.drawRect(100, 100, 200, 200);
g.drawLine(100, 100, 200, 20);
g.drawLine(300, 100, 200, 20);
g.drawRect(150, 200, 100, 100);
if (isDoorBlue) {
g.setColor(Color.BLUE);
}
else {
g.setColor(Color.RED);
}
g.fillRect(150, 200, 100, 100);
}
}
/*
<html>
<head>
<title>Applet</title>
</head>
<body>
<applet code="Door.class" width="500" height="500"></applet>
</body>
</html>
*/
```

Output Screenshot



Result

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

Experiment No.: 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.

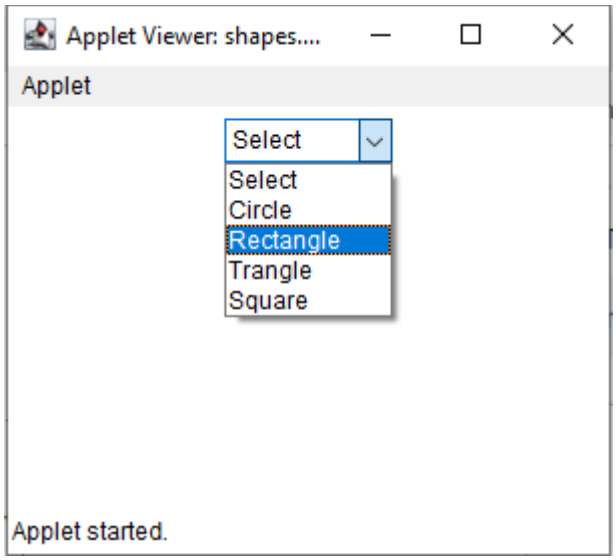
CO5: Develop applications to handle events using applets

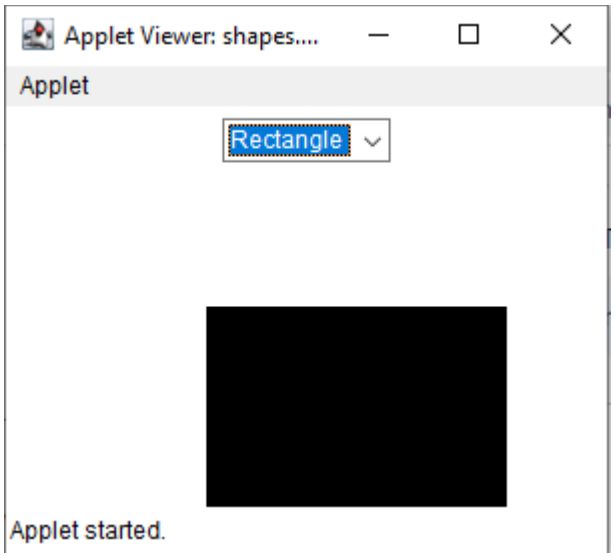
Procedure

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class shapes extends Applet implements ItemListener
{
        Choice ch;
        int s;
        public void init()
        {
                ch=new Choice();
                ch.addItem("Select");
                ch.addItem("Circle");
                ch.addItem("Rectangle");
                ch.addItem("Trangle");
                ch.addItem("Square");
                add(ch);
                ch.addItemListener(this);
        }
        public void itemStateChanged(ItemEvent e)
        {
                s=ch.getSelectedIndex();
                repaint();
        }
        public void paint(Graphics g)
        {
                if(s==1)
                        g.fillOval(100,100,150,150);
                if(s==2)
                        g.fillRect(100,100,150,100);
                if(s==3)
                {
                        g.drawLine(100,50,100,100);
                   g.drawLine(100,50,50,100);
                        g.drawLine(100,100,50,100);
                }
                if(s==4)
                        g.fillRect(100,100,100,100);
        }
}
```

```
/*<html><head>
<title>choice</title>
</head>
<body>
<applet code="shapes.class"width="300"height="200">
</applet>
</body>
</html>*/
```

Output Screenshot





Result

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

Experiment No.: 38

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

CO5: Develop applications to handle events using applets

Procedure

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

public class MouseEventAndWindowEvent extends Frame implements MouseListener,
WindowListener
{

   public MouseEventAndWindowEvent()
         {
      setTitle("Mouse and Window Event Demo");
      setSize(400, 300);
      // Add event listeners to the frame
      addMouseListener(this);
      addWindowListener(this);
   }

   // MouseListener methods
   public void mouseClicked(MouseEvent e)
         {
      System.out.println("Mouse Clicked ");
   }

   public void mousePressed(MouseEvent e)
         {
      System.out.println("Mouse Pressed ");
   }

   public void mouseReleased(MouseEvent e)
         {
      System.out.println("Mouse Released ");
   }

   public void mouseEntered(MouseEvent e)
         {
      System.out.println("Mouse Entered");
   }

   public void mouseExited(MouseEvent e)
         {
      System.out.println("Mouse Exited");
   }
```

```
    // WindowListener methods
    public void windowOpened(WindowEvent e)
            {
        System.out.println("Window Opened");
    }

    public void windowClosing(WindowEvent e)
            {
        System.out.println("Window Closing");
        System.exit(0); // Terminate the program when the window is closed
    }

    public void windowClosed(WindowEvent e)
            {
        System.out.println("Window Closed");
    }

    public void windowIconified(WindowEvent e)
            {
        System.out.println("Window Iconified (Minimized)");
    }


    public void windowDeiconified(WindowEvent e)
            {
        System.out.println("Window Deiconified (Restored)");
    }

    public void windowActivated(WindowEvent e)
            {
        System.out.println("Window Activated");
    }


    public void windowDeactivated(WindowEvent e)
            {
        System.out.println("Window Deactivated");
    }

    public static void main(String[] args)
            {
        MouseEventAndWindowEvent demo = new MouseEventAndWindowEvent();
        demo.setVisible(true);
    }
}
```

Output Screenshot

```
C:\Jenny\S2\Java>javac MouseEventAndWindowEvent.java

C:\Jenny\S2\Java>java MouseEventAndWindowEvent
Window Activated
Window Opened
Mouse Entered
Mouse Pressed
Mouse Released
Mouse Clicked
Mouse Exited
Window Iconified (Minimized)
Window Deactivated
Window Deiconified (Restored)
Window Activated
Mouse Entered
Mouse Exited
Window Closing
```

Result

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

Experiment No.: 39

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.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class FileWriteRead {
public static void main(String[] args) {
      try {
         FileWriter writer = new FileWriter("file.txt",true);
         writer.write("File is created with name file.txt ");
         writer.close();
         FileReader reader = new FileReader("file.txt");
         BufferedReader br= new BufferedReader(reader);
         String line;
         System.out.println("Data read from the file");
         while ((line = br.readLine()) != null) {
            System.out.println(line);
         }
         reader.close();
} catch (IOException e) {
         System.out.println("-----Error-----");
      }}}
```

Output Screenshot

```
C:\Jenny\S2\oop\java\bin>javac FileWriteRead.java

C:\Jenny\S2\oop\java\bin>java FileWriteRead
Data read from the file
File is created with name file.txt
```

Result

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

Experiment No.: 40

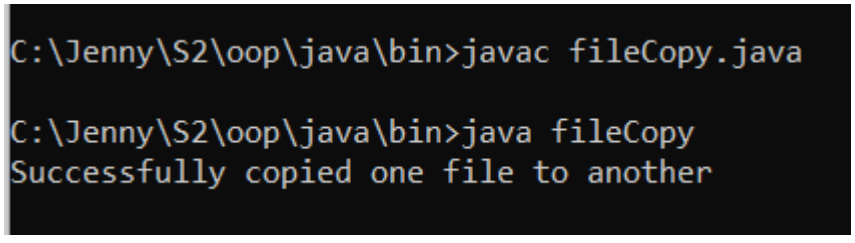
Aim: Write a program to copy one file to another.

CO6: Design applications using files and networking concepts.

Procedure

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
public class fileCopy
{
   public static void main(String[] args) throws IOException
        {
      FileInputStream fileinput = new FileInputStream("file.txt");

      FileOutputStream fileoutput = new FileOutputStream("file2.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 Screenshot



Result

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

Experiment No.: 41

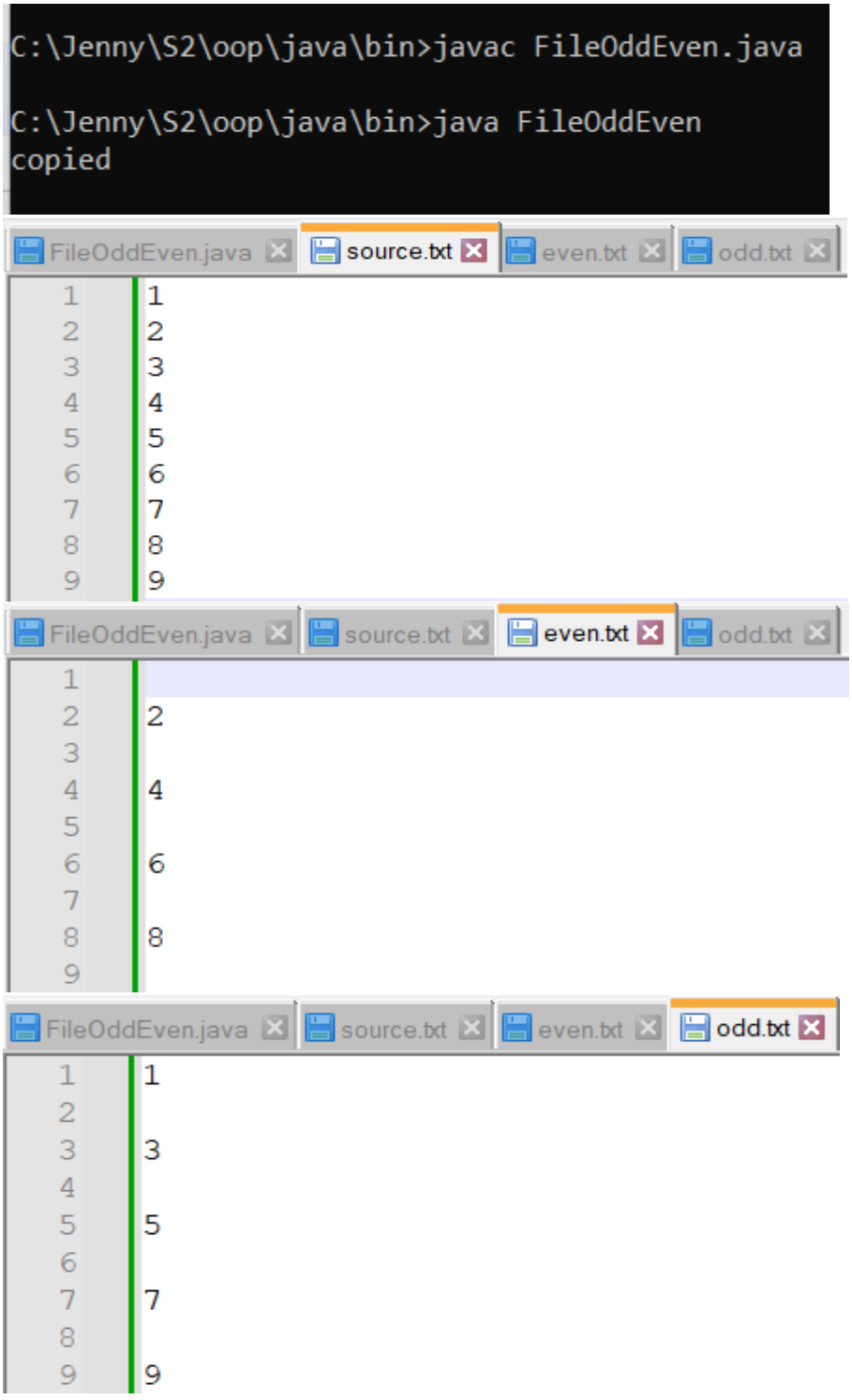
Aim: Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

CO6: Design applications using files and networking concepts.

Procedure

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class FileOddEven
{

   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);
                                    //System.out.println("Even :"+i);
         }
         else {
            destination_odd.write(i);
                                    //System.out.println("Odd :"+i);
         }
      }
      System.out.println("copied");
      source.close();
      destination_even.close();
      destination_odd.close();



   }

}
```

Output Screenshot



Result

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

Experiment No.: 42

Aim: Client Server communication using DatagramSocket - UDP

CO6: Design applications using files and networking concepts.

Procedure

Server.java

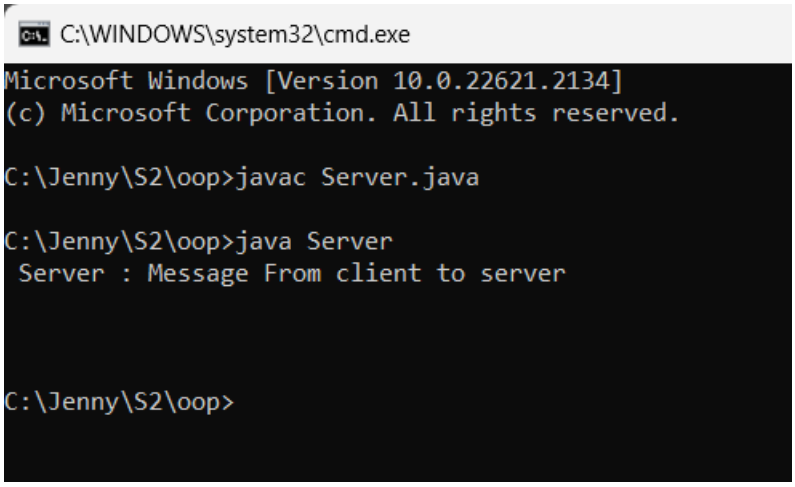
```
import java.io.*;
import java.net.*;
public class Server
{
public static void main(String[] args) throws IOException
{
        DatagramSocket server=new DatagramSocket(9000);
        byte[] buf=new byte[256];
        DatagramPacket packet=new DatagramPacket(buf,buf.length);
        server.receive(packet);
        String response =new String(packet.getData());
        System.out.println(" Server : "+response);
        server.close();
}
}
```

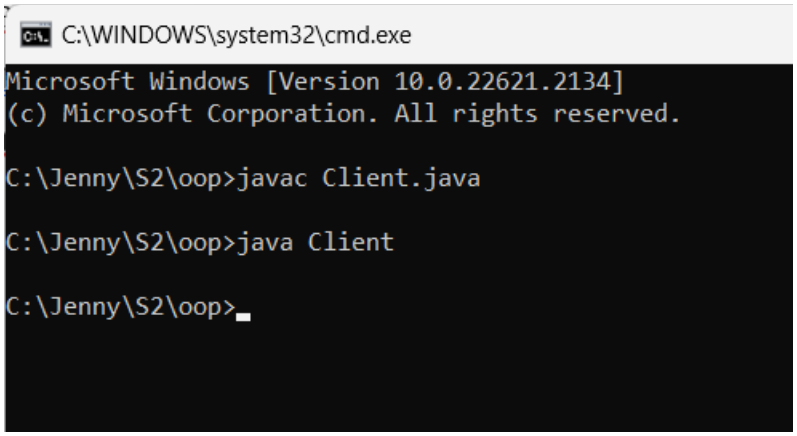
Client.java

```
import java.io.*;
import java.net.*;
public class Client
{
public static void main(String[] args) throws IOException
{
        DatagramSocket client= new DatagramSocket();
        InetAddress add=InetAddress.getByName("localhost");
```

```
        String str ="Message From client to server";
        byte[] bufBytes = str.getBytes();
        DatagramPacket datagramPacket=new
        DatagramPacket(bufBytes,bufBytes.length,add,9000);
        client.send(datagramPacket);
        client.close();
}
}
```

Output Screenshot





Result

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