20MCA132 – OBJECT ORIENTED PROGRAMMING LAB

Lab Report Submitted By

VISHAL C VISWAM

Reg. No.: AJC22MCA-2099

In Partial fulfillment 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 **VISHAL C VISWAM (REG. NO. AJC22MCA-2099)** in partial fulfillment 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.**

Sr Elsin Chakkalackal S H
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	Design applications using files and network concepts.	60
CO6	Design 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.
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	25
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	29
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	32
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	35
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	38

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.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 Screenshot

```
E:\vishal\S2\JAVA\CO1>java Product
Enter product code:
101
Enter product name:
laptop
Enter product price:
45000
Enter product code:
102
Enter product name:
pc
Enter product price:
56000
Enter product code:
103
Enter product name:
table
Enter product price:
35000
Product code is: 101
Product name is: laptop
Product price is: 45000
Product code is: 102
Product name is: pc
Product price is: 56000
Product code is: 103
Product name is: table
Product price is: 35000
35000is lowest price

E:\vishal\S2\JAVA\CO1>
```

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 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 elments 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 elments 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 elments 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 elments 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 Screenshot

```
E:\vishal\S2\JAVA\CO1>javac addmatrix.java

E:\vishal\S2\JAVA\CO1>java addmatrix
enter the size of row :
2
 enter the size of the column :
2
enter the elments of first array:
12
34
66
55
the first array elments are :
12 34
66 55
enter the elments of second array :
44
87
11
13
the second array elments are :
44 87
11 13
the sum of array elments is :
56 121
77 68
```

Result

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

Experiment No: 3

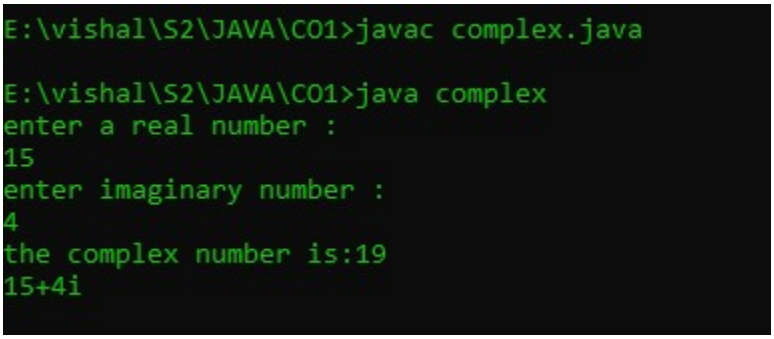
Aim
Add complex numbers.

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

Procedure

```
import java.util.*;
public class complex{
	public static void main(String[] args)
	{
		Scanner obj=new Scanner(System.in);
		System.out.println("enter a real number :");
		int real=obj.nextInt();
		System.out.println("enter imaginary number :");
		int img=obj.nextInt();
		int comp=real+img;
		System.out.println("the complex number is:"+comp);
		System.out.println(real+"+"+img+ "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.Scanner;
public class SymtrcMatrx
{
	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 Screenshot

```
E:\vishal\S2\JAVA\CO1>java SymtrcMatrx
Enter the row and col of first matrix
2
2
enter elements to first array
44
55
66
77
Printing 2d first Matrix
44 55
66 77
The matrix is a Not a Symmetric Matrix

E:\vishal\S2\JAVA\CO1>java SymtrcMatrx
Enter the row and col of first matrix
2
2
enter elements to first array
44
44
44
44
Printing 2d first Matrix
44 44
44 44
The matrix is a Symmetric Matrix
```

Result

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

Experiment No: 5

Aim

Program to Sort strings

CO2

Implement arrays and strings.

Procedure

```
import java.util.Scanner;

public class CO_2sortstrings {
	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 Screenshot

```
E:\vishal\S2\JAVA\CO2>javac CO_2sortstrings.java

E:\vishal\S2\JAVA\CO2>java CO_2sortstrings
Enter the size of the Array  :5
Enter the String Elements
kerala
day
a
beautiful
in
The Sorted String order :
a
beautiful
day
in
kerala
```

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

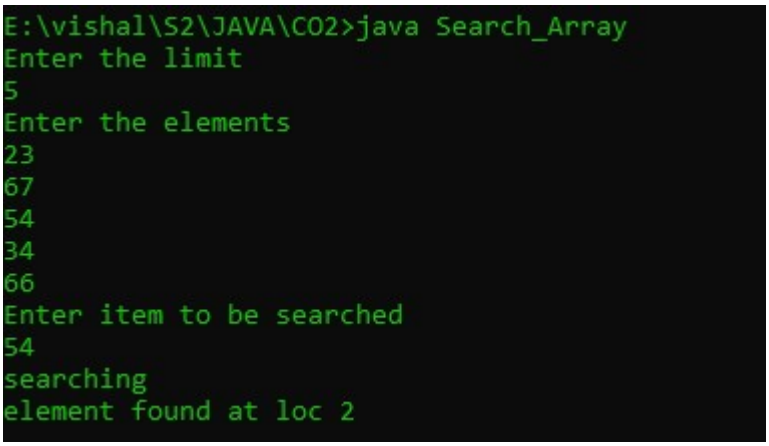
Implement arrays and strings.

Procedure

```
import java.util.Scanner;
public class  Search_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 loc "+i);
			flag=1;

		 }

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

	  }
```

}

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

Implement arrays and strings.

Procedure

```
import java.util.*;
public class string_manip{
	public static void main(String[] args){
		Scanner obj=new Scanner(System.in);
		System.out.println("enter first string :");
		String first=obj.nextLine();
		System.out.println("enter second string :");
		String second=obj.nextLine();
		System.out.println(first.concat(second));
		if(first == second)
		{
			System.out.println("two strings are same");
		}
		else
		{
			System.out.println("strings are not same");
		}
		System.out.println("the length of the first string is :"+first.length());
		System.out.println("the length of the first string is :"+second.length());
		System.out.println(first.equals(second));
		int len1=first.length();
		int len2=second.length();
		if(len1==len2)
		{
			System.out.println("the length is same");
		}
		else
		{
		System.out.println("lengths are not same");
		}
		System.out.println(first.toUpperCase());
		System.out.println(second.toUpperCase());
		System.out.println(first.toLowerCase());
```

```
            System.out.println(second.toLowerCase());

            System.out.println("enter string to locate :");
            String loc=obj.nextLine();
            System.out.println(first.indexOf(loc));
            System.out.println(first.substring(1,5));

        }
}
```

Output Screenshot

```
E:\vishal\S2\JAVA\CO2>javac string_manip.java

E:\vishal\S2\JAVA\CO2>java string_manip
enter first string :
vishal
enter second string :
viswam
vishalviswam
strings are not same
the length of the first string is :6
the length of the first string is :6
false
the length is same
VISHAL
VISWAM
vishal
viswam
enter string to locate :
2
-1
isha
```

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, eNameeSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

CO2

Implement arrays and strings.

Procedure

```
import java.util.*;

public class employee_search{
        int eno;
        String ename;
        int salary;

        public void get()
        {
                Scanner obj=new Scanner(System.in);
                System.out.println("enter employee number :");
                eno =obj.nextInt();
                System.out.println("enter employee name :");
                ename =obj.next();
                System.out.println("enter employee salary :");
                salary =obj.nextInt();

        }
        public void display()
        {
                System.out.println("employee number :"+eno);
                System.out.println("employee name :"+ename);
                System.out.println("employee salary :"+salary);
        }

        public static void main(String[] args)
        {
                int flag=0;
                Scanner obj=new Scanner(System.in);
                System.out.println("enter the size :");
                int n=obj.nextInt();
```

```
                employee_search e1[]= new employee_search[n];
                for(int i=0;i<n;i++)
                {
                        e1[i]=new employee_search();
                        e1[i].get();
                }
                for(int i=0;i<2;i++)
                {
                        e1[i].display();
                }
                System.out.println("enter employee number to search for the employee details :");
                int item=obj.nextInt();

                for(int i=0;i<n;i++)
                {
                        if(e1[i].eno ==item)
                        {
                                e1[i].display();
                                flag++;
                                break;
                        }
                }
                if(flag==0)
                        {
                                System.out.println("not found");
                        }
        }
}
```

Output Screenshot

```
E:\vishal\S2\JAVA\CO2>javac employee_search.java

E:\vishal\S2\JAVA\CO2>java employee_search
enter the size :
2
enter employee number :
101
enter employee name :
vishal
enter employee salary :
45000
enter employee number :
102
enter employee name :
amal
enter employee salary :
56000
employee number :101
employee name :vishal
employee salary :45000
employee number :102
employee name :amal
employee salary :56000
enter employee number to search for the employee details :
101
employee number :101
employee name :vishal
employee salary :45000
```

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

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.*;
class CO3_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(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);
        CO3_Shape sh =new CO3_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.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 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

```
E:\vishal\S2\JAVA>javac CO3_Shape.java

E:\vishal\S2\JAVA>java CO3_Shape
Area of different shapes
 1.Circle
 2.Triangle
 3.Square
 4.Rectangle
 5.Parallelogram
 6.Ellipse
Enter choice:
1
Enter the radius:
23
Area of circle: 1661.0600000000002
Enter choice:
2
Enter the breadth:
43
Enter the height:
5
Area of triangle: 107.5
Enter choice:
3
Enter the length:
23
Area of square: 529
Enter choice:
4
Enter the length:
66
Enter the breadth:
13
Area of rectangle: 858.0
Enter choice:
6
Enter the minor axis:
34
Enter the major axis:
222
Area of ellipse: 23700.72
Enter choice:
```

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

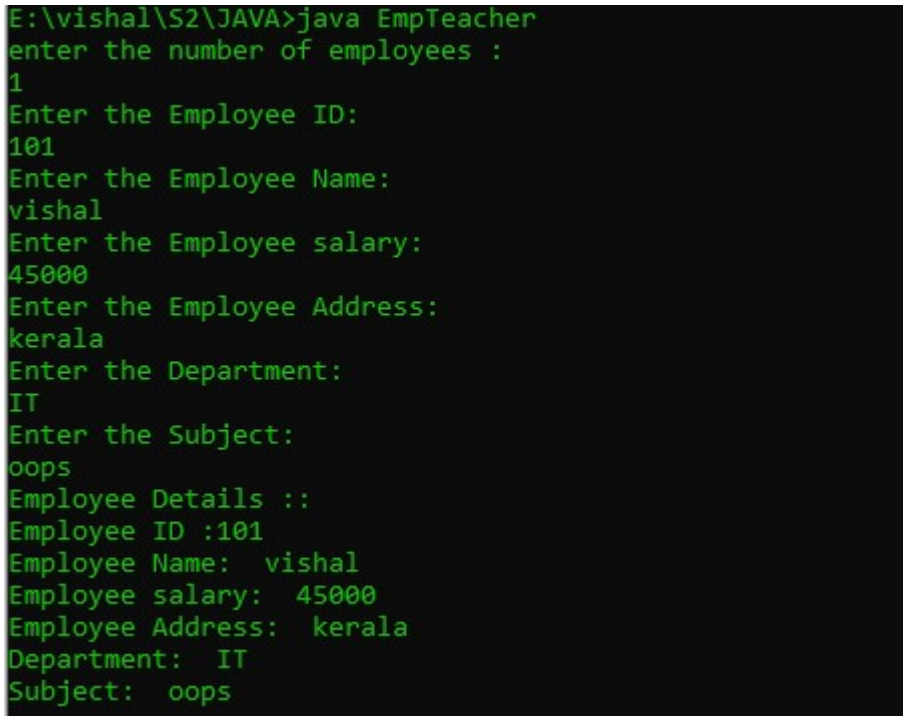
Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.*;
class Employee
{
        int Empid,Salary;
        String Name,Address;
        Scanner sc=new Scanner(System.in);

        Employee()
        {
                System.out.println("Enter the Employee ID:  ");
                Empid=sc.nextInt();
                System.out.println("Enter the Employee Name:  ");
                Name=sc.next();
                System.out.println("Enter the Employee salary:  ");
                Salary=sc.nextInt();
                System.out.println("Enter the Employee Address:  ");
                Address=sc.next();

        }
}

class Teacher extends Employee
{
        String Dep,Sub;
        Teacher()
        {

           System.out.println("Enter the Department:  ");
                Dep=sc.next();
```

```
        System.out.println("Enter the Subject:  ");
                Sub=sc.next();
        }

        void Display()
        {
                System.out.println("Employee ID :"+Empid);
                System.out.println("Employee Name:  "+Name);
                System.out.println("Employee salary:  "+Salary);
                System.out.println("Employee Address:  "+Address);
           System.out.println("Department:  "+Dep);
        System.out.println("Subject:  "+Sub);

        }
}
class EmpTeacher{


public static void main(String[] a)
        {
                int n,i;
                Scanner obj=new Scanner(System.in);
                System.out.println("enter the number of employees :");
                n=obj.nextInt();

                Teacher array[]=new Teacher[n];

                for(i=0;i<n;i++)
                {

                        array[i]=new Teacher();
                }
                System.out.println("Employee Details ::");
                for(i=0;i<n;i++)
                {
                        array[i].Display();

                }

        }
}
```

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

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.Scanner;
class person{
	String pname;
	String pgender;
	String paddress;
	int page;
	person(String name,String gender,String address,int age){
		pname=name;
		pgender=gender;
		paddress=address;
		page=age;
	}
}
class employee extends person{
	int empid;
	String cmpny_name;
	String qualificatiion;
	int salary;
	employee(String name,String gender,String address,int age,int eid,String cmpny,String
qualif,int sal){
		super(name,gender,address,age);
		empid=eid;
		cmpny_name=cmpny;
		qualificatiion=qualif;
		salary=sal;
	}
}
```

```
class teacher extends employee{
	int teacherid;
	String subject;
	String department;
	teacher(String name,String gender,String address,int age,int eid,String cmpny,String
qualif,int sal,int tid,String sub,String dep){
		super(name,gender,address,age,eid,cmpny,qualif,sal);
		teacherid=tid;
		subject=sub;
		department=dep;
	}
	void display(){
		System.out.println("\n");
		System.out.println("Person name:"+pname);
		System.out.println("Person gender:"+pgender);
		System.out.println("Person address:\n"+paddress);
		System.out.println("Person age:"+page);
		System.out.println("Employee id:"+empid);
		System.out.println("Company name: "+cmpny_name);
		System.out.println("Employee qualification: "+qualificatiion);
		System.out.println("Employee salary: "+salary);
		System.out.println("Teacher id: "+teacherid);
		System.out.println("Department: "+department);
		System.out.println("Subject taught: "+subject);
	}
}

public class SuperClass2{
	public static void main(String[] args){
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter the Limit:");
		int limit=sc.nextInt();
		teacher tcher[]=new teacher[limit];
		for(int i=0;i<limit;i++)
		{
			System.out.println("Enter the Person name:");
			String name=sc.next();
			System.out.println("Enter the Person gender:");
			String gender=sc.next();
			System.out.println("Enter the Person address:");
			String address=sc.next();
			System.out.println("Enter the Person age:");
			int age=sc.nextInt();
			System.out.println("Enter the Employee id:");
			int eid=sc.nextInt();
```

```
                        System.out.println("Enter the Company name:");
                        String cmpny=sc.next();
                        System.out.println("Enter the qualification:");
                        String qualif=sc.next();
                        System.out.println("Enter the Salary:");
                        int sal=sc.nextInt();
                        System.out.println("Enter the Teacher id:");
                        int tid=sc.nextInt();
                        System.out.println("Enter the Department:");
                        String dep=sc.next();
                        System.out.println("Enter the Subject:");
                        String sub=sc.next();
                        tcher[i]=new
teacher(name,gender,address,age,eid,cmpny,qualif,sal,tid,dep,sub);
                }
                for(int i=0;i<limit;i++)
                {
                        tcher[i].display();
                }
        }
}
```

Output Screenshot

```
E:\vishal\S2\JAVA>java SuperClass2
Enter the Limit:
1
Enter the Person name:
vishal
Enter the Person gender:
male
Enter the Person address:
kerala
Enter the Person age:
23
Enter the Employee id:
101
Enter the Company name:
TCS
Enter the qualification:
MCA
Enter the Salary:
45000
Enter the Teacher id:
121
Enter the Department:
CS
Enter the Subject:
C


Person name:vishal
Person gender:male
Person address:
kerala
Person age:23
Employee id:101
Company name: TCS
Employee qualification: MCA
Employee salary: 45000
Teacher id: 121
Department: C
Subject taught: CS
```

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

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 Screenshot

```
E:\vishal\S2\JAVA>java library
enter the total number :
1
enter the details of literature books :
Publisher name  :
vishal books
book name  :
the lovestory of vishal
enter the details of fiction books :
Publisher name  :
vishal matrix
book name  :
timetravel by vishal




details of literature books :
Publisher name is :vishal
book name is :the




details of fiction books :
Publisher name is :vishal
book name is :the
```

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

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 multiinher{
        public static void main(String[] args)
        {
        result obj=new result();
        obj.getS();
        obj.getSp();
        obj.dispS();
        obj.dispSp();
        }
}
```

Output Screenshot

```
E:\vishal\S2\JAVA>java multiinher
enter the name :
vishal
enter the roll number :
49
enter the mark 1 :
45
enter the mark2 :
78
enter the sports item which the student participated in :
football
enter the rank of the student :
2
Name of the Student :vishal
Roll Number of the Student :49
Mark 1 the Student :45
Mark 2 of the Student :78
Total Mark of the Student :123.0
Percentage of the Student :246.0
Sports item :football
The Rank  :2
```

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

Implement object-oriented concepts like inheritance, overloading and interfaces

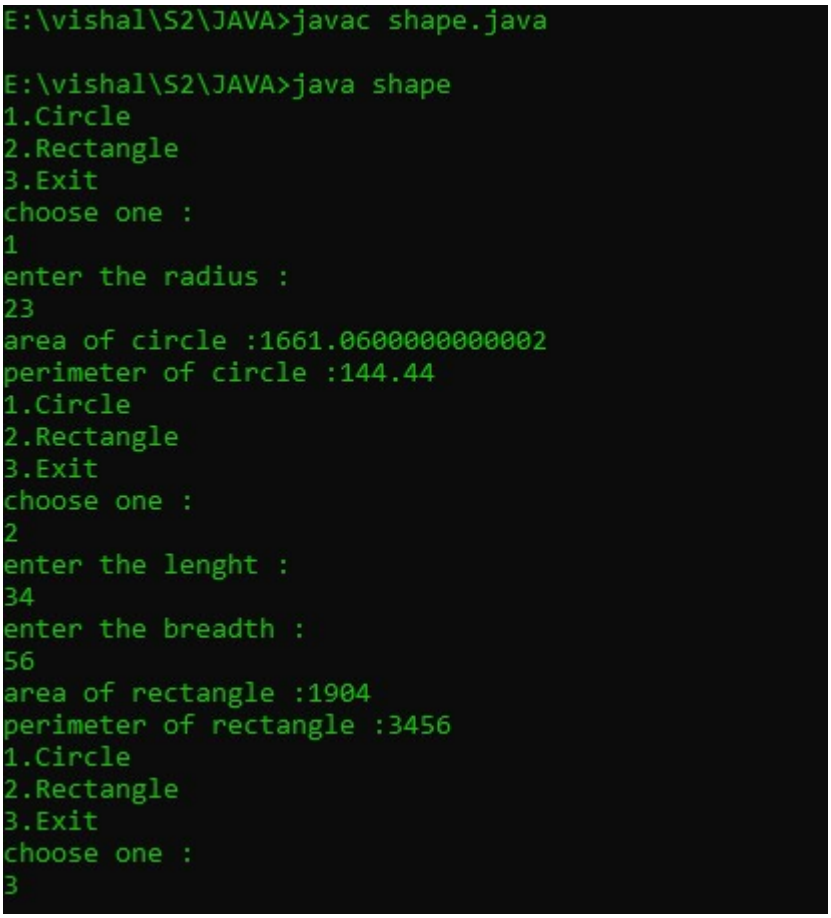
Procedure

```
import java.util.*;

interface prototype{
	public void getdata();
	public void area();
	public void perimeter();
}

class circle implements prototype
{
	Scanner obj=new Scanner(System.in);
	int radius;
	double z=3.14;

	public void getdata()
	{
		System.out.println("enter the radius :");
		radius=obj.nextInt();
	}
	public void area()
	{
		System.out.println("area of circle :"+z*(radius*radius));
	}
	public void perimeter()
	{
		System.out.println("perimeter of circle :"+(2*z)*radius);
	}

}

class rectangle implements prototype
```

```
{
	Scanner obj=new Scanner(System.in);
	int l,b;
	public void getdata()
	{
		System.out.println("enter the lenght :");
		l=obj.nextInt();
		System.out.println("enter the breadth :");
		b=obj.nextInt();
	}
	public void area()
	{
		System.out.println("area of rectangle :"+l*b);
	}
	public void perimeter()
	{
		System.out.println("perimeter of rectangle :"+l+b);
	}
}

class shape{
	public static void main(String[] args)
	{
		int ch,u=0;

		circle cc=new circle();


		rectangle jj=new rectangle();

		while(u==0)
		{
			Scanner obj=new Scanner(System.in);
			System.out.println("1.Circle \n2.Rectangle \n3.Exit");
			System.out.println("choose one :");
			ch=obj.nextInt();
			switch(ch)
			{
				case 1:cc.getdata();
					cc.area();
					cc.perimeter();
					break;

				case 2:jj.getdata();
					jj.area();
```

```
                                    jj.perimeter();
                                    break;
                            case 3:System.exit(0);

                                default:
                                System.out.println("choose valid one :");
                                break;


                        }
                    }
                }
}
```

Output Screenshot



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 :

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

CO3

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure

```
import java.util.*;

interface product
{
        public void getdata();
        public void display();
        public void calc();
}

class calculate implements product
{
        int pid;
        String name;
        double quantity;
        double unit_price,total;

        public void getdata()
        {
                Scanner obj=new Scanner(System.in);
                System.out.println("enter the product id :");
                pid=obj.nextInt();
                System.out.println("enter the name :");
                name=obj.next();
                System.out.println("enter the quantity :");
                quantity=obj.nextDouble();
                System.out.println("enter the unit_price :");
```

```
                unit_price=obj.nextDouble();
        }

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

        public void display()
        {

        System.out.println(pid+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit_price+"\t\t"+total);

        }
}

class bill{
        public static void main(String[] args)
        {
                int i,odno,n;
                String date;
                double net_amnt =0;
                Scanner obj=new Scanner(System.in);
                System.out.println("enter order number :");
                odno=obj.nextInt();
                System.out.println("enter the date :");
                date=obj.next();
                System.out.println("enter the total number products :");
                n=obj.nextInt();

                calculate arr[]= new calculate[n];
                for(i=0;i<n;i++)
                {
                        arr[i] = new calculate();
                        arr[i].getdata();
                        arr[i].calc();
                }
                System.out.println("Order no : #"+odno);
                System.out.println("Date :"+date);

                System.out.println("Product Id          Name   Quantity          Unit Price
        Total");

        System.out.println("==========================================================
=====================");
```

```
                for(i=0;i<n;i++)
                {
                        arr[i].display();
                        net_amnt +=arr[i].total;

                }

        System.out.println("================================================
======================");
                System.out.println("\t\t\t\t\t\tNet Amount :"+net_amnt);



        }
}
```

Output Screenshot

```
E:\vishal\S2\JAVA>javac bill.java

E:\vishal\S2\JAVA>java bill
enter order number :
101
enter the date :
08/06/23
enter the total number products :
2
enter the product id :
121
enter the name :
apple
enter the quantity :
5
enter the unit_price :
14
enter the product id :
122
enter the name :
orange
enter the quantity :
7
enter the unit_price :
34
Order no : #101
Date :08/06/23
Product Id      Name    Quantity        Unit Price      Total
=========================================================================
121             apple           5.0             14.0            70.0
122             orange          7.0             34.0            238.0
=========================================================================
                                                Net Amount :308.0
```

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.

CO4

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

Procedure

Graphics.java

```
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();
```

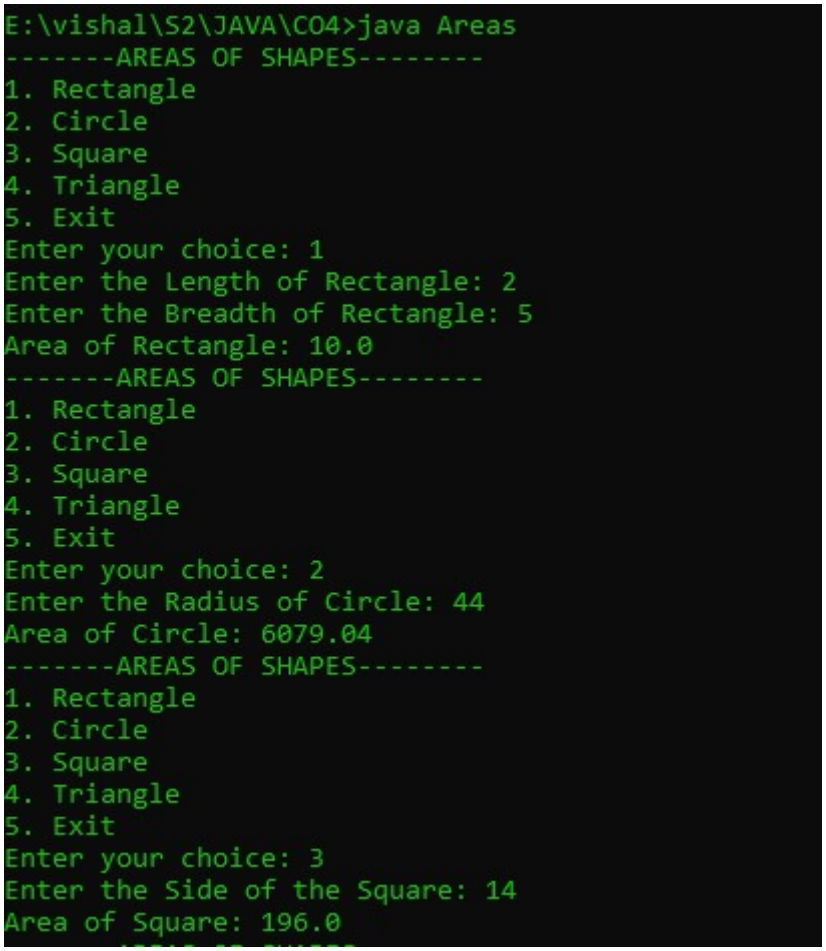
```
        area = s * s;
        return area;
    }
    public double TriangArea(){
        System.out.print("Enter the Width of the Triangle: ");
        double base = obj.nextDouble();
        System.out.print("Enter the Height of the Triangle: ");
        double height = obj.nextDouble();
      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

Shapes.java

```
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 Screenshot



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

CO4

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

Procedure

Arithmetic.java

```
package arithmetic;
import java.util.*;
interface maths
{
	public double add();
	public double subtract();
	public double multiply();
	public double division();
}

public class Arithmetic implements maths
{
	Scanner obj=new Scanner(System.in);
	int a1,a2,s1,s2,m1,m2,d1,d2;
	double total;
	public double add()
	{
		System.out.println("enter a number :");
		a1=obj.nextInt();
		System.out.println("enter a number :");
		a2=obj.nextInt();
		total=a1+a2;
		return total;
	}
	public double subtract()
	{
		System.out.println("enter a number :");
		s1=obj.nextInt();
		System.out.println("enter a number :");
		s2=obj.nextInt();
```

```
                total=s1-s2;
                return total;
        }
        public double multiply(){
                System.out.println("enter a number :");
                m1=obj.nextInt();
                System.out.println("enter a number :");
                m2=obj.nextInt();
                total=m1*m2;
                return total;
        }
        public double division()
        {
                System.out.println("enter a number :");
                d1=obj.nextInt();
                System.out.println("enter a number :");
                d2=obj.nextInt();
                total=d1/d2;
                return total;
        }
}
```

Maths.java

```
import arithmetic.Arithmetic;
import java.util.*;
public class maths
{
        public static void main(String args[])
        {
                Scanner obj=new Scanner(System.in);
                Arithmetic sc=new Arithmetic();
                int choice=0;
                while(choice!=5)
                {
                        System.out.println("------ARITHMETIC OPERATIONS-------
\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\n5.Exit");
                        System.out.println("enter a choice :");
                        choice=obj.nextInt();
                        switch(choice)
                        {
                                case 1:
                                System.out.println("the total is :"+sc.add());
                                break;
                                case 2:
                                System.out.println("the total is :"+sc.subtract());
```

```
                                break;
                                case 3:
                                System.out.println("the total is :"+sc.multiply());
                                break;
                                case 4:
                                System.out.println("the total is :"+sc.division());
                                break;
                                case 5:
            System.exit(0);
            break;
            default:
            System.out.println("Select a valid option!");
                        }
                    }
            }
}
```

Output Screenshot

```
E:\vishal\S2\JAVA\CO4>java maths
------ARITHMETIC OPERATIONS-------
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
enter a choice :
1
enter a number :
2
enter a number :
11
the total is :13.0
------ARITHMETIC OPERATIONS-------
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
enter a choice :
2
enter a number :
67
enter a number :
7
the total is :60.0
------ARITHMETIC OPERATIONS-------
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
enter a choice :
3
enter a number :
6
enter a number :
6
the total is :36.0
```

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.

CO4

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 Screenshot

```
E:\vishal\S2\JAVA\CO4>javac CheckLoginCredential.java

E:\vishal\S2\JAVA\CO4>java CheckLoginCredential
Enter username : vishal01
Enter password : qwerty
Login Successful !!!
The final statement

E:\vishal\S2\JAVA\CO4>java CheckLoginCredential
Enter username : vis
Enter password : hello
UsernameException: Username must be greater than 6 characters.....
        at CheckLoginCredential.main(CheckLoginCredential.java:33)
The final statement

E:\vishal\S2\JAVA\CO4>
```

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

CO4

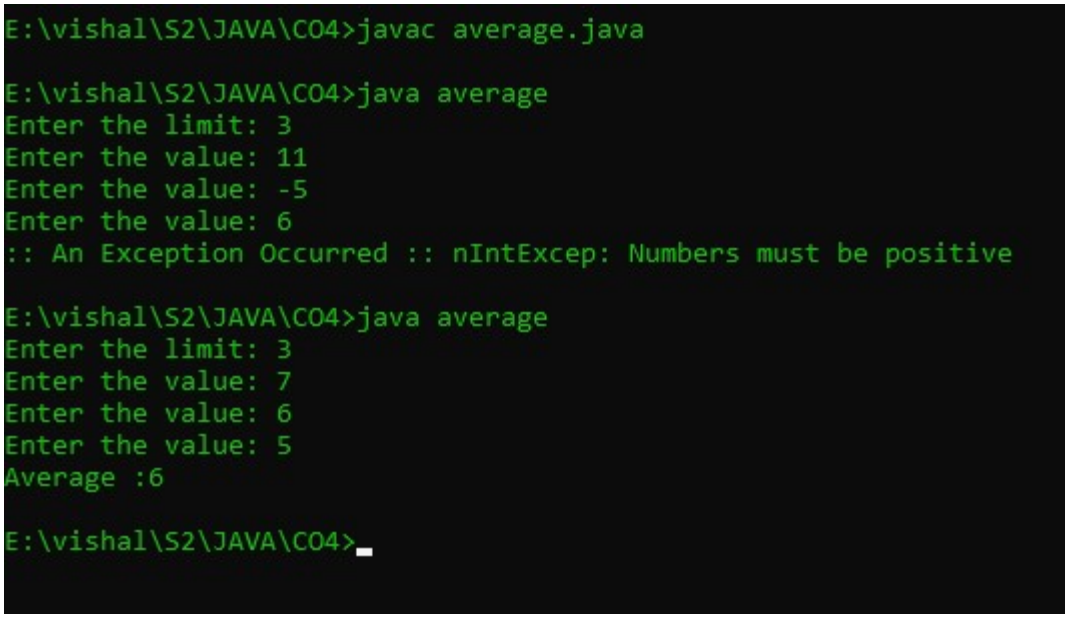
Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

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 Screenshot



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

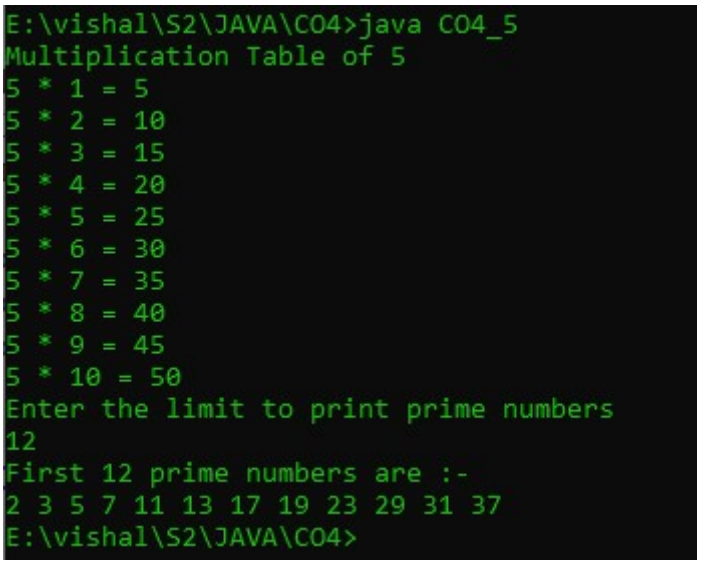
Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;
class ThreadA extends Thread{
   public void run( ) {
int n = 5;
   for (inti = 1; i<= 10; ++i)
System.out.println(n + " * " + i +
       " = " + n * i);
   }
}
class ThreadB extends Thread{
   public void run( ){
   Scanner sc = new Scanner(System.in);
inti,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 CO4_5{
   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: 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

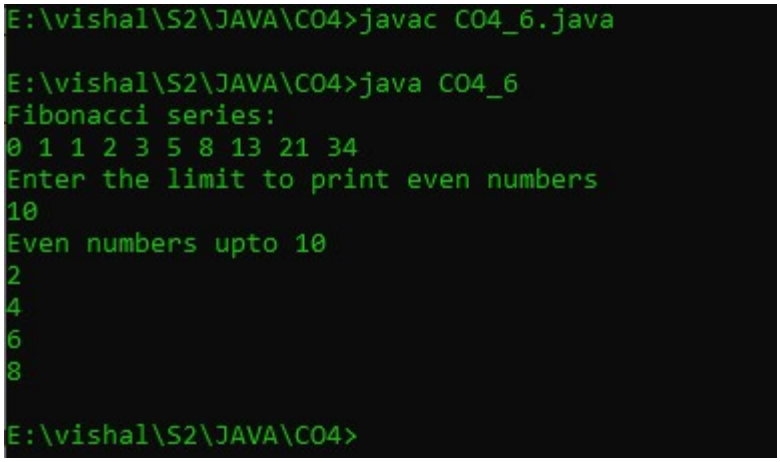
Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;
public class CO4_6 {
	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(inti=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);
		for(i=2;i<count;++i){
			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: 22

Aim

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

CO4

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

Procedure

```
import java.util.*;
public class CO4_8{
	public static void main( String args[]){
		intpos,ch;
		Scanner obj =new Scanner(System.in);
		Stack<String>sta =new Stack<String>();
		do{
			System.out.println("STACK");
			System.out.println("1.ADD ");
			System.out.println("2.REMOVE ");
			System.out.println("3.DISPLAY ");
			System.out.print("Select your Option: ");
			ch=obj.nextInt();
		switch(ch){
			case 1:
				System.out.print("Enter the Element to be inserted :- ");
				sta.add(obj.next()) ;
				System.out.println("Inserted to the Stack :- ");
				break;
			case 2:
				System.out.print("Enter the position of  element removed from stack: ");
				pos=obj.nextInt();
				sta.remove(pos);
				break;
			case 3:
				System.out.println(sta);
		}
		} while(ch!=0);
	}
}
```

Output Screenshot

```
E:\vishal\S2\JAVA\CO4>java CO4_8
STACK
1.ADD
2.REMOVE
3.DISPLAY
Select your Option: 1
Enter the Element to be inserted :- 22
Inserted to the Stack :-
STACK
1.ADD
2.REMOVE
3.DISPLAY
Select your Option: 1
Enter the Element to be inserted :- 44
Inserted to the Stack :-
STACK
1.ADD
2.REMOVE
3.DISPLAY
Select your Option: 3
[22, 44]
STACK
1.ADD
2.REMOVE
3.DISPLAY
Select your Option: 2
Enter the position of  element removed from stack: 1
STACK
1.ADD
2.REMOVE
3.DISPLAY
Select your Option: 3
[22]
STACK
1.ADD
2.REMOVE
3.DISPLAY
Select your Option:
```

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

Experiment: 23

Aim

Using generic method perform Bubble sort.

CO4

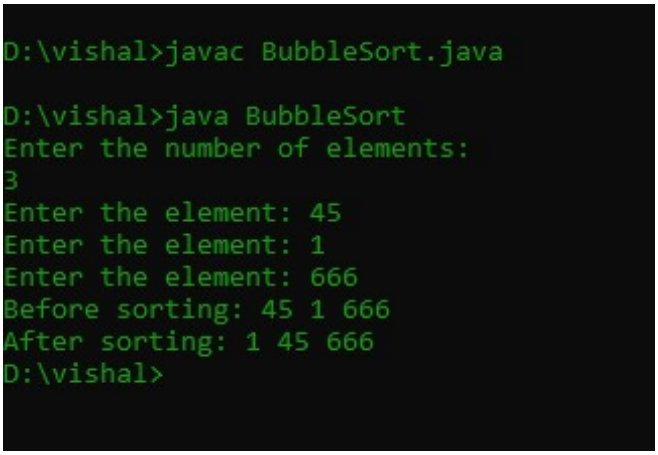
Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

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 Screenshot



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.

CO4

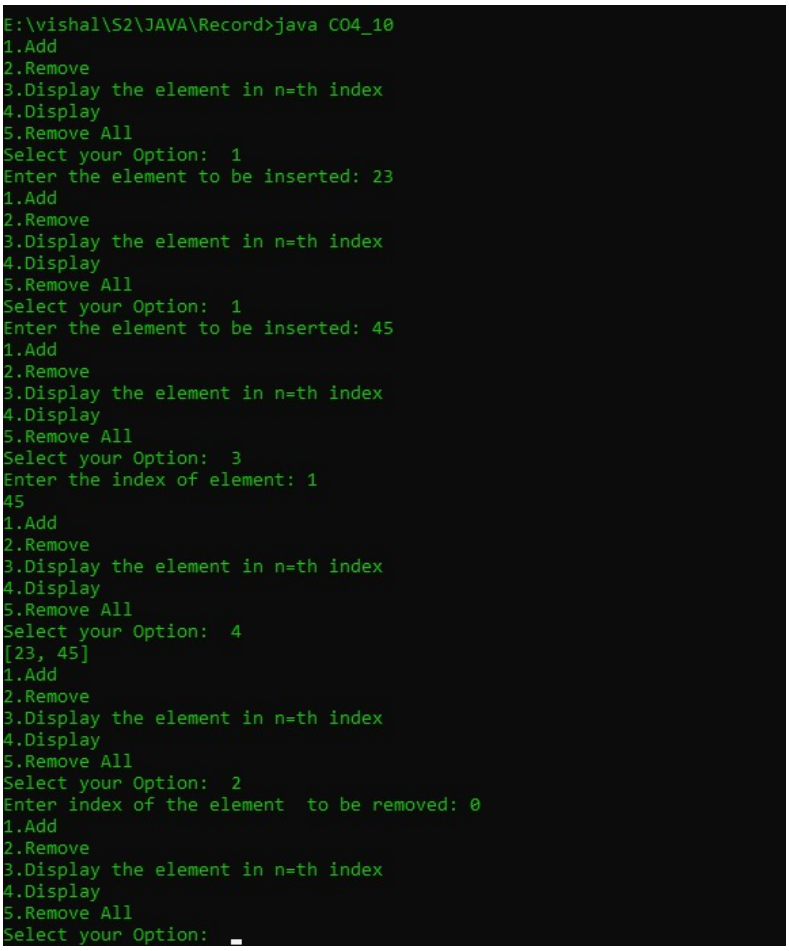
Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework

Procedure

```
import java.util.*;
class CO4_10{
        public static void main(String args[]){
                Scanner value = new Scanner(System.in);
                intch;
                String str;
                int n;
                ArrayList<String> list = new ArrayList<String>();
                do{
                        System.out.println("1.Add ");
System.out.println("2.Remove ");
System.out.println("3.Display the element in n=th index ");
System.out.println("4.Display ");
System.out.println("5.Remove All ");
System.out.print("Select your Option:  ");
ch=value.nextInt();
                switch(ch){
                        case 1:
System.out.print("Enter the element to be inserted: ");
str=value.next();
list.add(str);
                                break;
                        case 2:
                                System.out.print("Enter index of the element  to be removed: ");
                                n=value.nextInt();
                                list.remove(n);
                                break;
                        case 3:
                                System.out.print("Enter the index of element: ");
                                n=value.nextInt();
                                System.out.println(list.get(n));
                                break;
```

```
                        case 5:
list.removeAll(list);
                                break;
                        case 4 :
                                System.out.println(list);
                                break;
        }
                }
                while(ch!=0);
        }
}
```

Output Screenshot



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

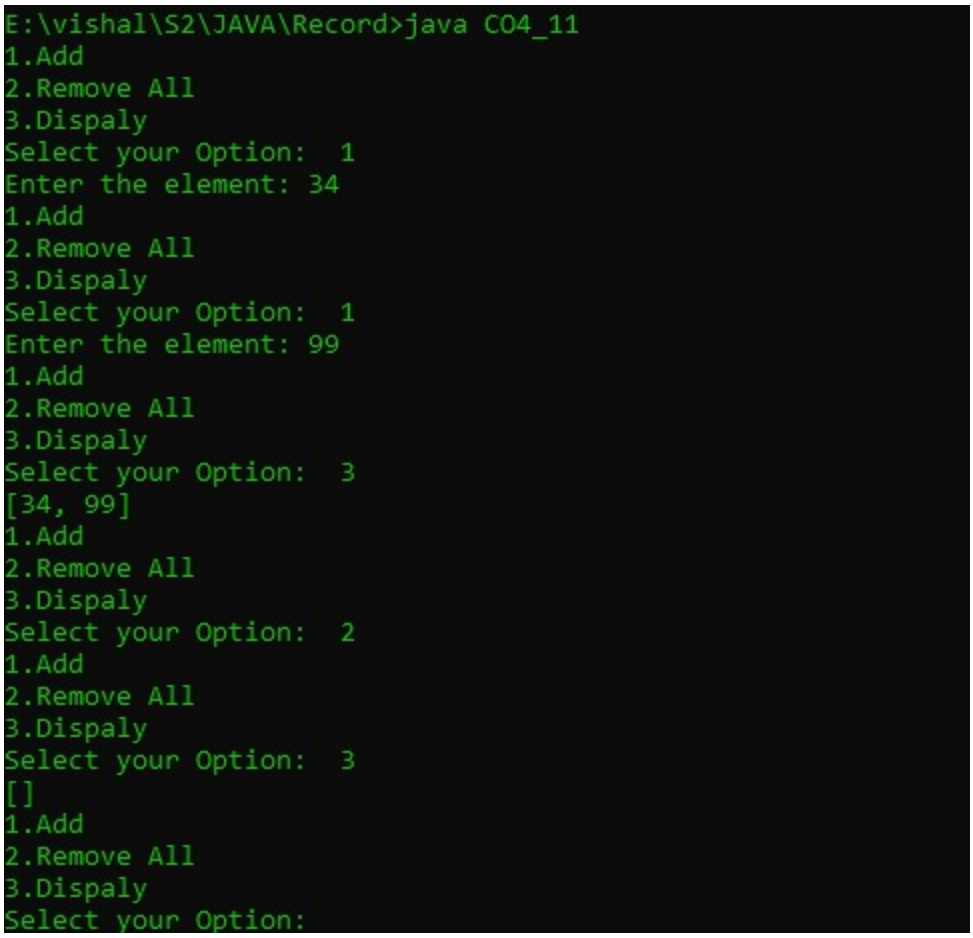
CO4

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

Procedure

```
import java.util.*;
	public class CO4_11{
		public static void main (String [] args){
			LinkedList<String> list = new LinkedList<String>();
			Scanner value = new Scanner(System.in);
			intch;
		do{
			System.out.println("1.Add ");
			System.out.println("2.Remove All ");
			System.out.println("3.Dispaly");
			System.out.print("Select your Option:  ");
			ch = value.nextInt();
			switch(ch){
				case 1:
					String val;
					int v;
					System.out.print("Enter the element: ");
					val = value.next();
					list.add(val);
					break;
				case 2:
					list.clear();
					break;
				case 3:
					System.out.println(list);
					break;
			}
		}
		while(ch !=0);
		}
	}
```

Output Screenshot



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

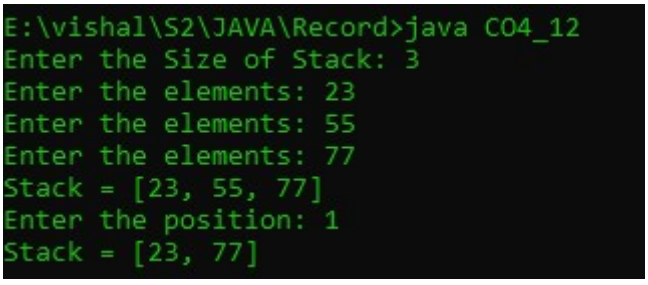
CO4

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

Procedure

```
import  java.util.*;
public class CO4_12 {
   public static void main(String[] args) {
      Scanner obj = new Scanner (System.in);
      Stack<Integer>st = new Stack<>();
System.out.print("Enter the Size of Stack: " );
int n = obj.nextInt();
      for (inti =0 ; i<n;i++){
                              System.out.print("Enter the elements: " );
                              st.add(obj.nextInt());
                     }
System.out.println("Stack = "+st);
      Scanner sc = new Scanner(System.in);
System.out.print("Enter the position: ");
int x = sc.nextInt();
st.remove(x);
System.out.println("Stack = "+st);
   }
}
```

Output Screenshot



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

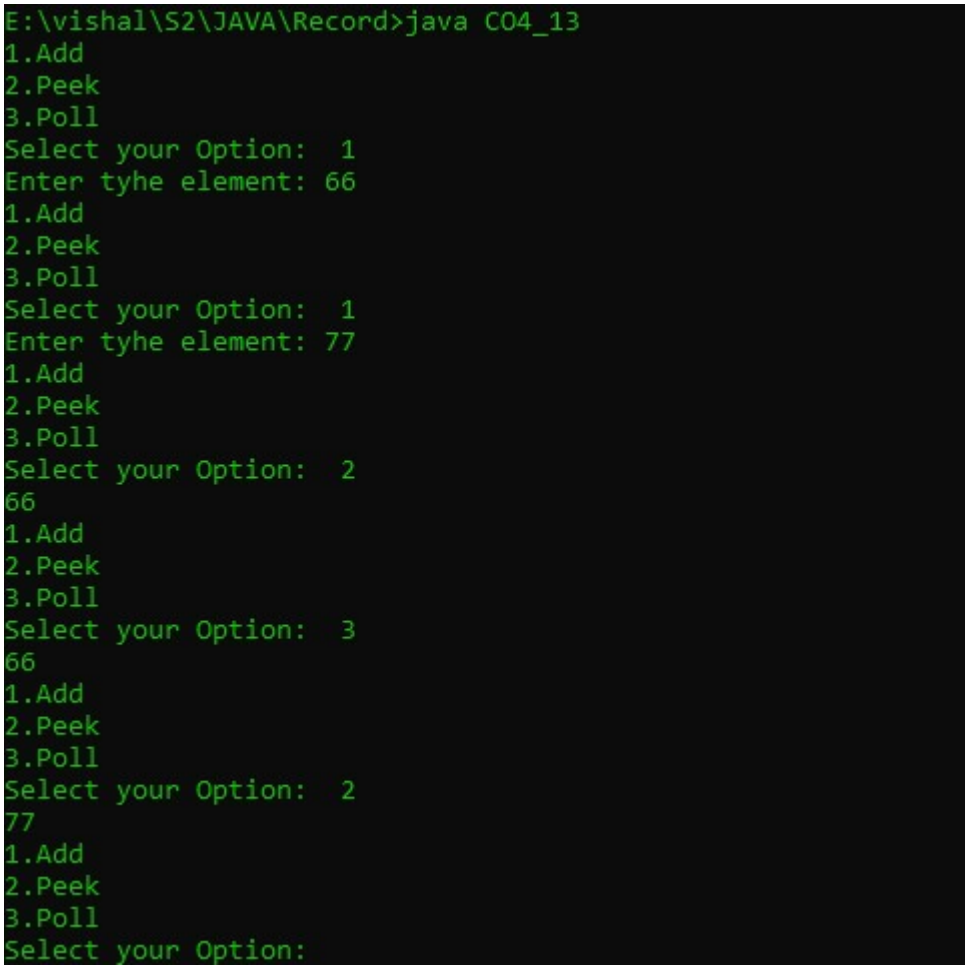
CO4

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

Procedure

```
import java.util.*;
class CO4_13{
	public static void main(String args[]){
		Scanner value = new Scanner(System.in);
		PriorityQueue<Integer>pq = new PriorityQueue<>();
		intch;
		do{
			System.out.println("1.Add ");
			System.out.println("2.Peek ");
			System.out.println("3.Poll ");
			System.out.print("Select your Option:  ");
			ch = value.nextInt();
			switch(ch){
				case 1:
					intval;
					System.out.print("Enter tyhe element: ");
					val = value.nextInt();
					pq.add(val);
					break;
				case 2:
					System.out.println(pq.peek());
					break;
				case 3:
					System.out.println(pq.poll());
					break;
			}
		}
		while(ch !=0);
	}
}
```

Output Screnshot



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

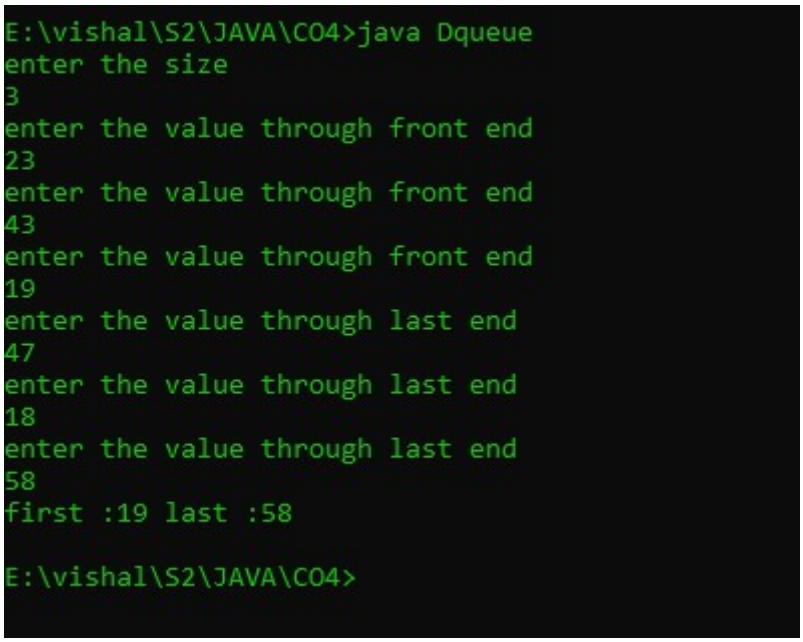
CO4

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 Screenshot



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.

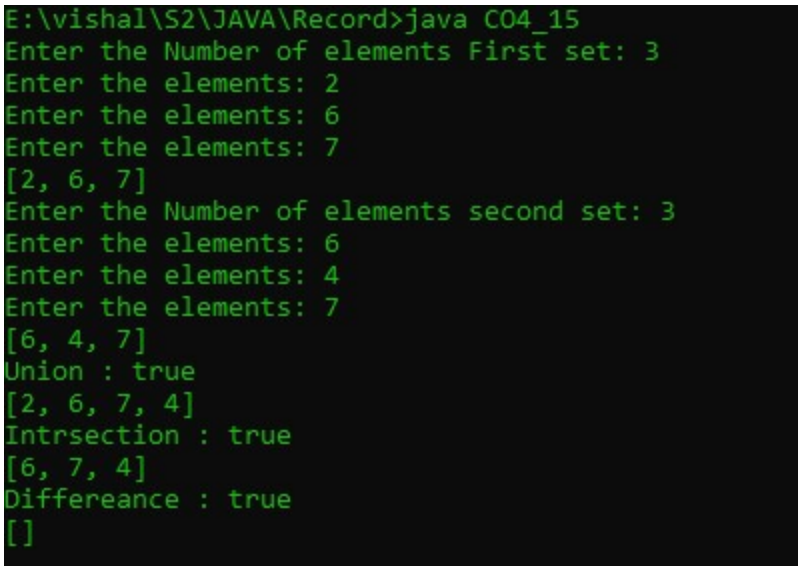
CO4

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

Procedure

```
import java.util.*;
public class CO4_15{
	public static void main(String args[]){
		Scanner obj = new Scanner (System.in);
		Set <Integer>Srt =new LinkedHashSet<Integer>();
		Set <Integer> Srt2 =new LinkedHashSet<Integer>();
		Set <Integer> Union =new LinkedHashSet<Integer>();
		Set <Integer> inter =new LinkedHashSet<Integer>();
		intn,u,l;
		System.out.print("Enter the Number of elements First set: " );
		n=obj.nextInt();
		for (inti =0 ; i<n;i++){
			System.out.print("Enter the elements: " );
			Srt.add(obj.nextInt());
		}
		System.out.println(Srt);
		System.out.print("Enter the Number of elements second set: " );
		l=obj.nextInt();
		for (inti =0 ; i<l;i++){
			System.out.print("Enter the elements: " );
			Srt2.add(obj.nextInt());
		}
		System.out.println(Srt2);
		Union=Srt;
		System.out.println("Union : " + Union.addAll(Srt2));
			System.out.println(Union);
		inter=Srt;
		System.out.println("Intrsection : " + inter.retainAll(Srt2));
		System.out.println(inter);
		System.out.println("Differeance : " + Srt.removeAll(Srt2));
		System.out.println(Srt);
	}}
```

Output Screenshot



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.

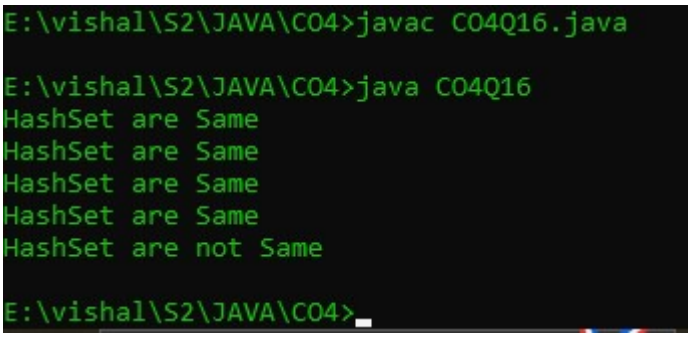
CO4

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

Procedure

```
import java.util.HashSet;
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) ? "HashSet are Same" : "HashSet are not Same");
      } }}
```

Output Screenshot



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.

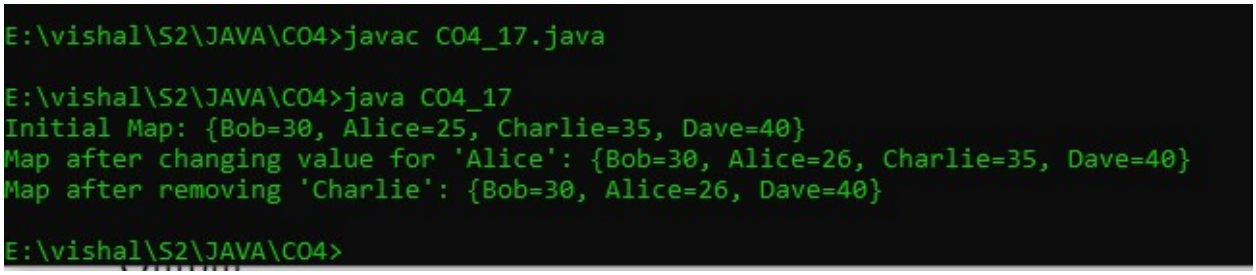
CO4

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

Procedure

```
import java.util.HashMap;
import java.util.Map;
public class CO4_17 {
   public static void main(String[] args) {
      Map<String, Integer> map = new HashMap<>();
      map.put("Alice", 25);
      map.put("Bob", 30);
      map.put("Charlie", 35);
      map.put("Dave", 40);
      System.out.println("Initial Map: " + map);
      map.put("Alice", 26);
      System.out.println("Map after changing value for 'Alice': " + map);
      map.remove("Charlie");
      System.out.println("Map after removing 'Charlie': " + map);
   }
}
```

Output Screenshot



Result

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

Experiment: 32

Aim

Program to Convert HashMap to TreeMap

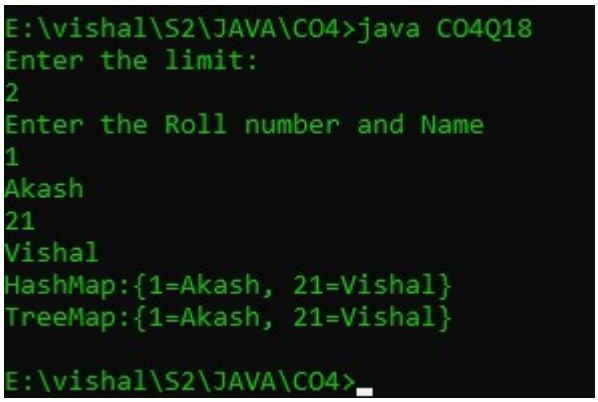
CO4

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 args[]) {

      Map<String, String> map = new HashMap<>();
      System.out.println("Enter the limit:");
      Scanner inp = new Scanner(System.in);
      int n= inp.nextInt();
      System.out.println("Enter the Roll number and Name");
      while(n!=0) {

         String e= inp.next();
         String s= inp.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 Screenshot



Result

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

Experiment: 33

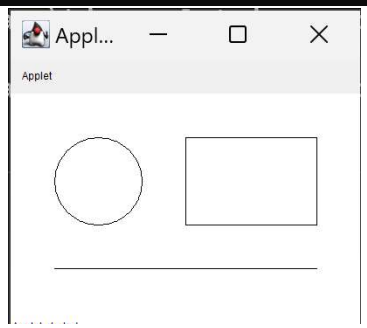
Aim
Program to draw Circle, Rectangle, Line in Applet.

CO5
Develop applications to handle events using applets.

Procedure

co5q1.java
```
import java.applet.Applet;
import java.awt.*;
public class co5q1 extends Applet {
   public void paint(Graphics g) {
g.drawOval(50, 50, 100, 100);
g.drawRect(200, 50, 150, 100);
g.drawLine(50, 200, 350, 200);
   }}
```
co5q1.html
```
<html>
<head><title>Applet</title></head><body>
<applet code="co5q1.class" width="400" height="400"></applet>
</body></html>
```

Output Screenshot



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

Experiment: 34

Aim

Program to find maximum of three numbers using AWT.

CO5

Develop applications to handle events using applets.

Procedure

co5q2.java

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class co5q2 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("Display Maximum:");
   b1.setBounds(175,65,50,40);
   add(t1);
   add(t2);
   add(t3);
   add(t4);
   add(b1);
   b1.addActionListener(this);
}
public void actionPerformed(ActionEvent e){
inti,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);
```

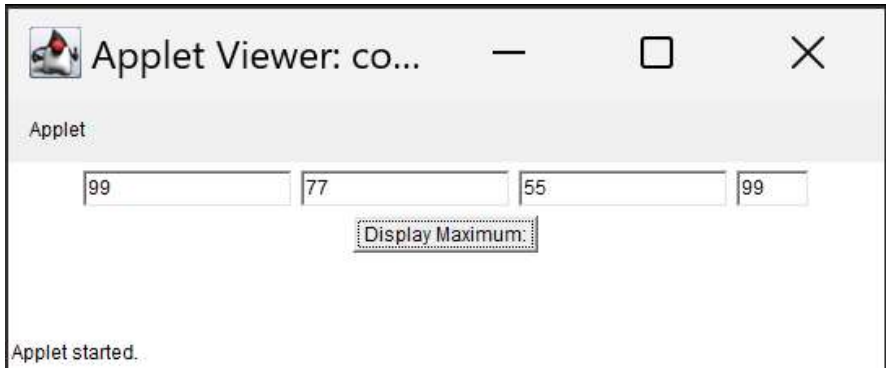
```
        else
            t4.setText(""+j);
        }
    else
        t4.setText(""+i);
    }
}
```

co5q2.html

```
<html>
<body>
<applet code="co5q2.class" width="500" height="500"></applet>
</body>
</html>
```

Output Screenshot

```
E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>javac co5_q2.java

E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>appletviewer co5_q2.html
```



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.

CO5

Develop applications to handle events using applets.

Procedure

co5q3.java

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class co5q3 extends Applet implements ActionListener{
   public int per =0;
      Label l1 = new Label("Subject 1: ");
      Label l2 = new Label("Subject 2: ");
      Label l3 = new Label("Subject 3: ");
      Label l4 = new Label("Subject 4: ");
      Label l5 = new Label("Subject 5: ");
      Label l6 = new Label("Percentage: ");
TextField t1 = new TextField(3);
TextField t2 = new TextField(3);
TextField t3 = new TextField(3);
TextField t4 = new TextField(3);
TextField t5 = new TextField(3);
TextField t6 = new TextField(3);
      Button b1 = new Button("CALCULATE");
   public co5q3(){
      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);
```

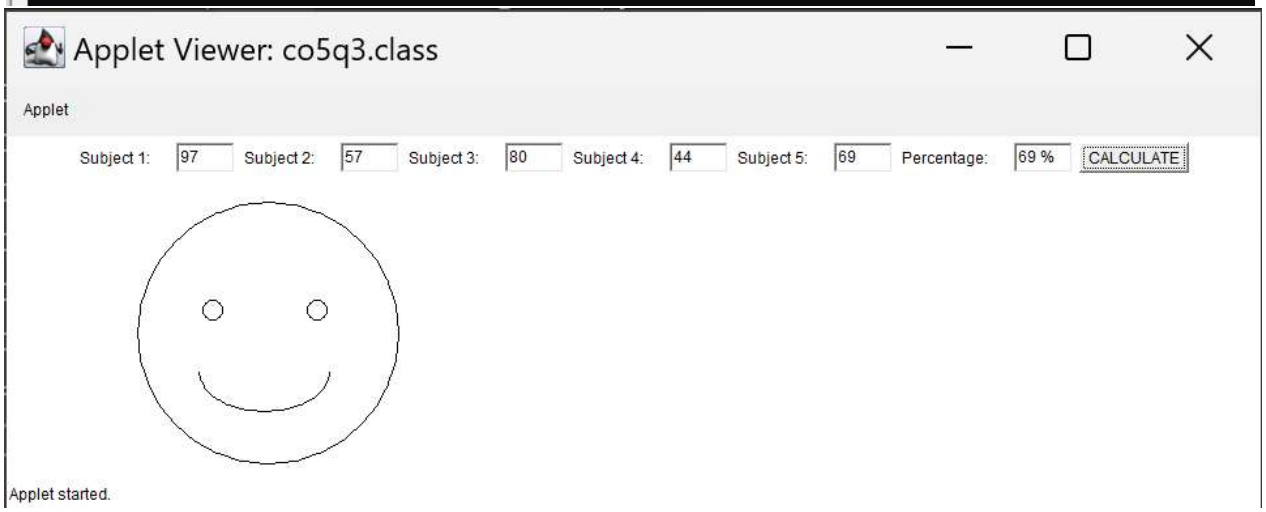
```
      b1.addActionListener(this);
   }
   public void actionPerformed(ActionEvent e){
int m1 = Integer.parseInt(t1.getText());
int m2= Integer.parseInt(t2.getText());
int m3= Integer.parseInt(t3.getText());
int m4= Integer.parseInt(t4.getText());
int m5= Integer.parseInt(t5.getText());
   if(e.getSource()==b1){
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.drawOval(100,50,200,200);
g.drawOval(150,125,15,15);
g.drawOval(230,125,15,15);
g.drawArc(147,150,100,60,0,-180);
   }
   else if(per>0 && per<50){
g.drawOval(100,50,200,200);
g.drawOval(150,125,15,15);
g.drawOval(230,125,15,15);
g.drawArc(147,170,100,60,0,180);
   }
}
public static void main(String args[]){
   new co5q3();
}
}
```

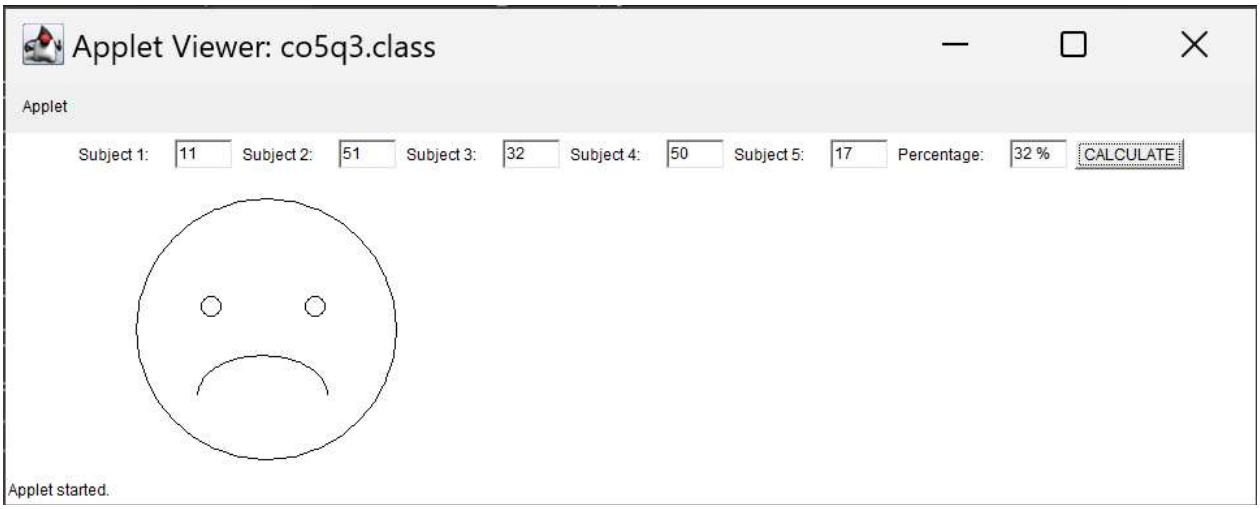
co5q3.html

```
<html>
<body>
<applet code="co5q3.class"width="1000"height="1000"></applet></div>
</body>
</html>
```

Output Screenshot

```
E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>javac co5_q3.java

E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>appletviewer co5_q3.html
```





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.

CO5

Develop applications to handle events using applets.

Procedure

co5q4.java

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.*;
public class co5q4 extends Applet{
   private booleanisDoorBlue = 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);
   }
}
```

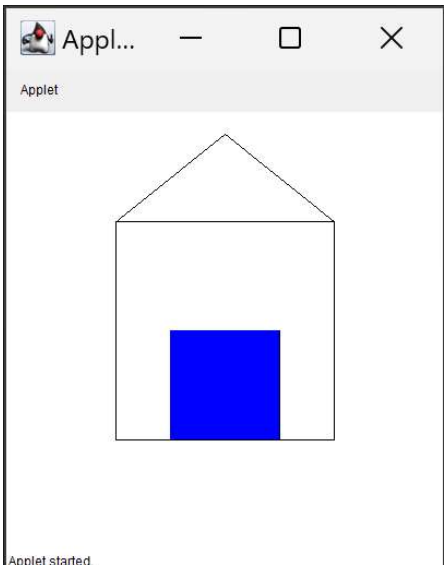
co5q4.html

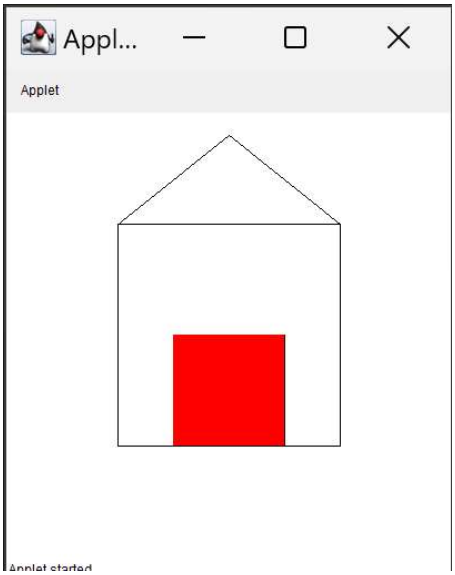
```
<html>
```

```
<body>
<applet code="co5q4.class" width="500" height="500"></applet>
</body>
</html>
```

Output Screenshot

```
E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>javac co5_q4.java

E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>appletviewer co5_q4.html
```





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.

CO5

Develop applications to handle events using applets.

Procedure

co5q5.java

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class co5q5 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.drawOval(100,100,150,150);
		if(s==2)
			g.drawRect(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);
		}
```

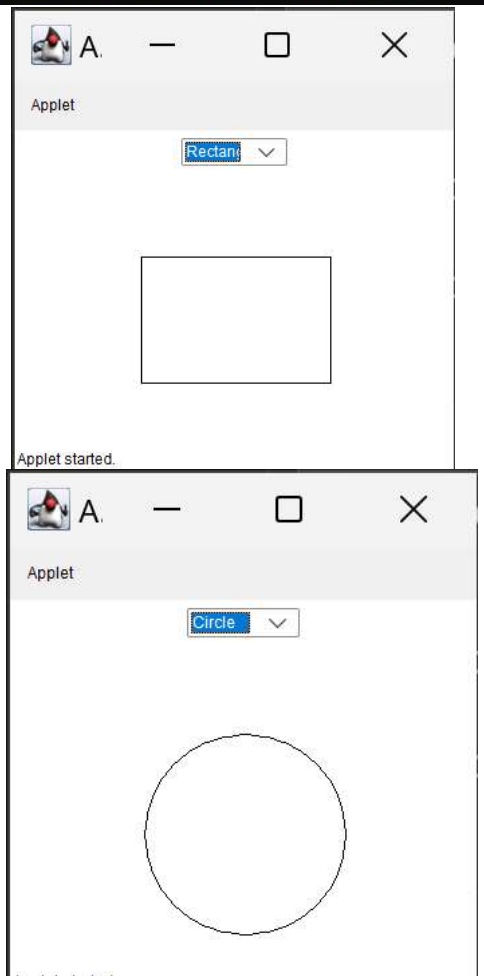
```
                if(s==4)
                        g.drawRect(100,100,100,100);
        }
}
```

co5q5.html

```
<html>
<body>
<applet code="co5q5.class" width="500" height="500"></applet>
</body>
</html>
```

Output Screenshot

```
E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>javac co5q5.java

E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>appletviewer co5q5.html
```



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.

CO5

Develop applications to handle events using applets.

Procedure

co5q6.java

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class co5q6 extends Applet implements MouseListener, MouseMotionListener, WindowListener {
   private String mouseEventMessage = "";
   private String windowEventMessage = "";
   @Override
   public void init() {
addMouseListener(this);
addMouseMotionListener(this);
      Frame frame = (Frame) this.getParent().getParent();
frame.addWindowListener(this);
   }
   @Override
   public void paint(Graphics g) {
g.drawString("Mouse Events: " + mouseEventMessage, 20, 20);
g.drawString("Window Events: " + windowEventMessage, 20, 40);
   }
   @Override
   public void mouseClicked(MouseEvent e) {
mouseEventMessage = "Mouse Clicked at (" + e.getX() + ", " + e.getY() + ")";
      repaint();
   }
   @Override
   public void mousePressed(MouseEvent e) {
mouseEventMessage = "Mouse Pressed at (" + e.getX() + ", " + e.getY() + ")";
      repaint();
   }
   @Override
   public void mouseReleased(MouseEvent e) {
mouseEventMessage = "Mouse Released at (" + e.getX() + ", " + e.getY() + ")";
```

```
        repaint();
    }
    @Override
    public void mouseEntered(MouseEvent e) {
mouseEventMessage = "Mouse Entered at (" + e.getX() + ", " + e.getY() + ")";
        repaint();
    }
    @Override
    public void mouseExited(MouseEvent e) {
mouseEventMessage = "Mouse Exited at (" + e.getX() + ", " + e.getY() + ")";
        repaint();
    }
    @Override
    public void mouseDragged(MouseEvent e) {
mouseEventMessage = "Mouse Dragged at (" + e.getX() + ", " + e.getY() + ")";
        repaint();
    }
    @Override
    public void mouseMoved(MouseEvent e) {
mouseEventMessage = "Mouse Moved at (" + e.getX() + ", " + e.getY() + ")";
        repaint();
    }
    @Override
    public void windowOpened(WindowEvent e) {
windowEventMessage = "Window Opened";
        repaint();
    }
    @Override
    public void windowClosing(WindowEvent e) {
windowEventMessage = "Window Closing";
        repaint();
    }
    @Override
    public void windowClosed(WindowEvent e) {
windowEventMessage = "Window Closed";
        repaint();
    }
    @Override
    public void windowIconified(WindowEvent e) {
windowEventMessage = "Window Iconified";
        repaint();
    }
    @Override
    public void windowDeiconified(WindowEvent e) {
windowEventMessage = "Window Deiconified";
```

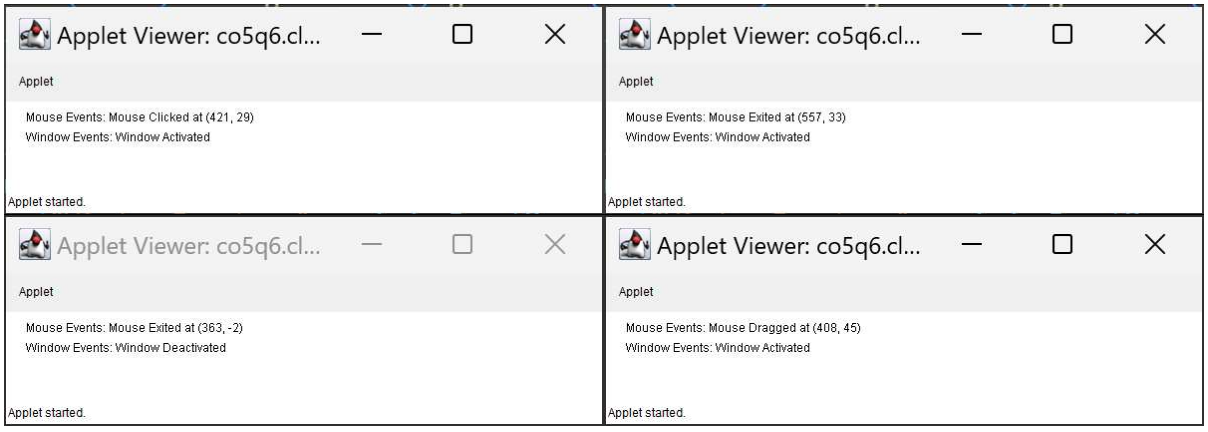
```
        repaint();
    }
    @Override
    public void windowActivated(WindowEvent e) {
windowEventMessage = "Window Activated";
        repaint();
    }
    @Override
    public void windowDeactivated(WindowEvent e) {
windowEventMessage = "Window Deactivated";
        repaint();
    }
}
```

co5q6,html

```
<html>
<body>
<applet code="co5q6.class" width="500" height="500"></applet>
</body>
</html>
```

Output Screenshot

```
E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>javac co5_q6.java

E:\vishal\S2\JAVA\java-20230719T011821Z-001\java\bin>appletviewer co5_q6.html
```



Result

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

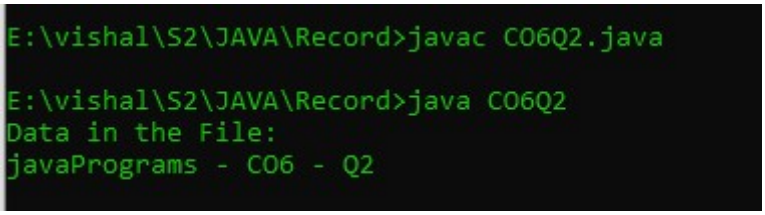
CO6

Develop 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 CO6Q2 {
   public static void main(String[] args) throws IOException {
FileWriter writer = new FileWriter("file.txt",true);
writer.write("javaPrograms - CO6 - Q2");
writer.close();
FileReader reader = new FileReader("file.txt");
BufferedReader br= new BufferedReader(reader);
      String line;
System.out.println("Data in the File: ");
      while ((line = br.readLine()) != null) {
System.out.println(line);
         }
reader.close();
      }
   }
```

Output Screenshot



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.

CO6

Develop applications using files and networking concepts.

Procedure

```
import java.io.*;
import java.util.*;
class CO6Q3 {
 public static void main(String arg[]) throws Exception {
  Scanner sc = new Scanner(System.in);
System.out.print("Enter Source Filename: ");
  String sfile = sc.next();
System.out.print("Enter Destination Filename: ");
  String dfile = sc.next();
FileReader fin = new FileReader(sfile);
FileWriterfout = new FileWriter(dfile, true);
int c;
  while ((c = fin.read()) != -1) {
fout.write(c);
  }
System.out.println("Copying finish... Successfully Copied " + sfile + " to " + dfile );
fin.close();
fout.close();
 }
}
```

Output Screenshot

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.

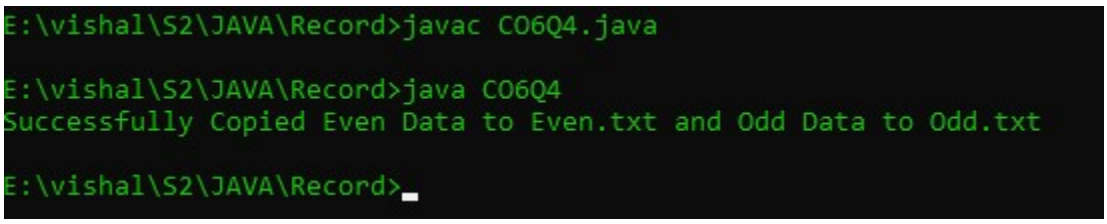
CO6

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 ("inputFile.txt");
FileOutputStreamoutputOdd = new FileOutputStream ("Odd.txt");
FileOutputStreamoutputEven = new FileOutputStream ("Even.txt");
inti;
       while((i = source.read()) != -1){
          if(i%2==0) {
outputEven.write(i);
          }
          else {
outputOdd.write(i);
          }}
System.out.println("Successfully Copied Even Data to Even.txt and Odd Data to Odd.txt");
source.close();
outputEven.close();
outputOdd.close();
   }}
```

Output Screenshot



Result

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

Experiment: 42

Aim

Client Server communication using DatagramSocket - UDP

CO6

Develop applications using files and networking concepts.

Procedure

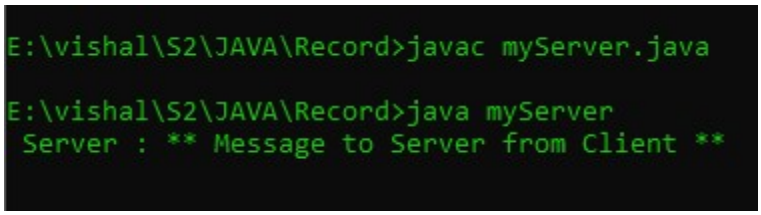
myServer.java

```
import java.io.*;
import java.net.*;
public class myServer {
   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();
   }
}
```

myClient.java

```
import java.io.*;
import java.net.*;
public class myClient {
   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();
DatagramPacketdatagramPacket=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.