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;
        String pname;
        int price;
        public void get(){
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the pcode:");
                pcode =sc.nextInt();
                System.out.println("Enter the pname:");
                pname =sc.next();
                System.out.println("Enter the pname:");
                price =sc.nextInt();}
        public void master(){
                System.out.println("poduct code :"+pcode);
                System.out.println("poduct pname :"+pname);
                System.out.println("poduct price :"+price);}
        public static void main(String[] args){
                Product p1 = new Product();
                Product p2 = new Product();
                Product p3 = new Product();
                System.out.println("poduct details1:");
                p1.get();
                System.out.println("poduct details2:");
                p2.get();
                System.out.println("poduct details3:");
                p3.get();
                System.out.println("poduct details1 first:");
                p1.master();
                System.out.println("poduct details1 second:");
                p2.master();
                System.out.println("poduct details1 third:");
                p1.master();}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\Tressa_Sajan\c01>javac Product.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\Tressa_Sajan\c01>java Product

....Product details1....

Enter the product code: 01
Enter the product name: Camera
Enter the product price: 2300

....Product details2....

Enter the product code: 02
Enter the product name: Pencil
Enter the product price: 2500

....Product details3....

Enter the product code: 03
Enter the product name: Pen
Enter the product price: 1500

|  Product details1 first  |

The product code: 1
The product pname: Camera
The product price: 2300

|  Product details1 second  |

The product code: 2
The product pname: Pencil
The product price: 2500

|  Product details1 third  |

The product code: 1
The product pname: Camera
The product price: 2300
```

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 program2{
        public static void main(String args[]){
         Scanner a= new Scanner(System.in);
         int i,j;
         System.out.print("\nEnter the number of rows: ");
         int r=a.nextInt();
         System.out.print("Enter the number of columns: ");
         int c=a.nextInt();
         System.out.println("Enter the elements of the matrix 1: ");
         int matrix1[][]=new int[r][c];
         for(i=0;i<r;i++){
                  for(j=0;j<c;j++){
        matrix1[i][j]=a.nextInt();        }}
        System.out.println("Matrix 1 is:");
        for(i=0;i<r;i++){
                  for(j=0;j<c;j++){
        System.out.print(matrix1[i][j]+"\t");   }
                   System.out.println("\n");       }
                  System.out.print("Enter the elements of the matrix 2:");
                  int matrix2[][]=new int[r][c];
         for(i=0;i<r;i++){
                  for(j=0;j<c;j++){
        matrix2[i] [j]=a.nextInt();        }}
        System.out.println("Matrix 2 is:");
        for(i=0;i<r;i++){
                  for(j=0;j<c;j++){
        System.out.print(matrix2[i][j]+"\t");  }
                  System.out.println("\n");}
        System.out.println("Sum of two matrix is:");
        int matrix3[][]=new int[r][c];
for(i=0;i<r;i++){
                  for(j=0;j<c;j++){
        matrix3[i][j]=matrix1[i][j]+matrix2[i][j];}}
                  for(i=0;i<r;i++){
                   for(j=0;j<c;j++){
        System.out.print(matrix3[i][j]+"\t");  }
                            System.out.println("\n");}}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO1>javac program2.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO1>java program2

Enter the number of rows: 2
Enter the number of columns: 2
Enter the elements of the matrix 1:
2
7
8
5
Matrix 1 is:
2       7

8       5

Enter the elements of the matrix 2:6
7
4
8
Matrix 2 is:
6       7

4       8

Sum of two matrix is:
8       14

12      13
```

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 complex3{
        public static void main(String args[]){
                Scanner a=new Scanner(System.in);
                System.out.print("\nEnter first real part: ");
                int r1=a.nextInt();
                System.out.print("Enter first complex part: ");
                int c1=a.nextInt();
                System.out.print("\nfirst complex no: "+r1+" + "+c1+"i\n");
                System.out.print("\nEnter second real part: ");
                int r2=a.nextInt();
                System.out.print("Enter second complex part: ");
                int c2=a.nextInt();
                System.out.println("\nsecond complex no: "+r2+" + "+c2+"i\n");
                System.out.println("Sum is: "+(r1+r2)+" + "+(c1+c2)+"i");}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO1>javac complex3.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO1>java complex3

Enter first real part: 2
Enter first complex part: 3

first complex no: 2 + 3i

Enter second real part: 6
Enter second complex part: 7

second complex no: 6 + 7i

Sum is: 8 + 10i
```

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 program4{
        public static void main(String args[]){
         Scanner a= new Scanner(System.in);
         int i,j,flag=0;
         System.out.print("\nEnter the number of rows: ");
         int r=a.nextInt();
         System.out.print("Enter the number of columns: ");
         int c=a.nextInt();
         System.out.println("Enter the elements of the matrix 1: ");
         int matrix1[][]=new int[r][c];
         for(i=0;i<r;i++){
                for(j=0;j<c;j++){
        matrix1[i][j]=a.nextInt();        }}
        System.out.println("\nMatrix 1 is:");
        for(i=0;i<r;i++){
                for(j=0;j<c;j++){
        System.out.print(matrix1[i][j]+"\t");  }
                System.out.println(" ");        }
   System.out.println("\nTranspose of Matrix 1 is:");
        for(i=0;i<r;i++){
                for(j=0;j<c;j++){
        System.out.print(matrix1[j][i]+"\t");  }
                System.out.println(" ");}
   for(i=0;i<r;i++){
                for(j=0;j<c;j++){
                        if(matrix1[i][j]!=matrix1[j][i])          {
                                        flag=1;
                                        break;}}}
                if(flag==0)      {
                                                System.out.print("\nMatrix is symmetric\n");}
                else{
                        System.out.print("\nMatrix is non symmetric\n");}}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO1>javac program4.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO1>java program4

Enter the number of rows: 2
Enter the number of columns: 2
Enter the elements of the matrix 1:
5
4
7
8

Matrix 1 is:
5       4
7       8

Transpose of Matrix 1 is:
5       7
4       8

Matrix is non symmetric
```

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 Program5{
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.print("\nEnter the size of the Array: ");
		int s = sc.nextInt();
		String str[] = new String[s];
		int i;
		System.out.println("Enter the Elements of the Array: ");
		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("\nThe Sorted Array element: ");
		for (i = 0; i < str.length; i++) {
			System.out.println(str[i]);}}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO1>javac Program5.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO1>java Program5

Enter the size of the Array: 4
Enter the Elements of the Array:
5
7
8
6

The Sorted Array element:
5
6
7
8
```

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.*;
public class Array{
	public static void main(String[] args){
	Scanner tre=new Scanner(System.in);
	int i;
	int val;
	int flag=0;
	System.out.print("\nEnter the size of the array: ");
	int size=tre.nextInt();
	int a[]=new int[size];
	System.out.println("\nEnter the elements: ");
	for(i=0;i<size;i++){
		a[i]=tre.nextInt();}
	System.out.println("\nArray Elements are: ");
		for( i=0;i<size;i++){
			System.out.print(a[i]+" ");}
		System.out.println(" ");
		System.out.print("\nEnter the value to be search: ");
		val=tre.nextInt();
		for(i=0;i<size;i++){
			if(a[i]==val){
				flag=1;}}
		if(flag==1){
			System.out.println("\nvalue found.");}
		else{
			System.out.println("\nvalue is not found.");}}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C02>javac Array.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C02>java Array

Enter the size of the array: 4

Enter the elements:
1
7
8
5

Array Elements are:
1 7 8 5

Enter the value to be search: 1

value found.
```

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 StrManipulation{
	public static void main(String[] args){
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the String1:");
		String str1 = sc.nextLine();
		System.out.println("Enter the String2:");
		String str2 = sc.nextLine();
		System.out.println("The String concat is"+str1.concat(str2));
		System.out.println("The String length is"+str1.length());
		System.out.println("The String length is"+str2.length());
		if(str1.length()==str2.length()){
			System.out.println("String are same");}
		else{
			System.out.println("String are not same");}
		System.out.println("String are same:"+str1.equals(str2));
		System.out.println("String compare Uppercase:"+str1.toUpperCase());
		System.out.println("String compare Lowercase:"+str1.toLowerCase());
		System.out.println("The substring of s1"+str1.substring(3));}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C02>javac StrManipulation.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C02>java StrManipulation

Enter the String1:Stefy
Enter the String2:Mol
The String concat is: StefyMol
The String length is: 5
The String length is: 3
String are not same
String are same:false
String compare Uppercase:STEFY
String compare Lowercase:stefy
The substring of s1fy
```

Result

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

Experiment No.: 8

Aim: To create a class for Employees having attributes eNo, eName eSalary. 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.*;
class employee{
        int eno;
        String n;
        int p;
        void get(){
                Scanner tre=new Scanner(System.in);
                System.out.print("Enter the NO: ");
                eno=tre.nextInt();
                System.out.print("Enter the Name: ");
                n=tre.next();
                System.out.print("Enter the price: ");
                p=tre.nextInt();
                System.out.println(" ");        }
        void put(){
                System.out.println("NO:"+eno);
                System.out.println("NAME:"+n);
                System.out.println("PRICE:"+p);        }
        public static void main(String args[]){
                Scanner tre=new Scanner(System.in);
                int i,limit,val;
                System.out.print("\nEnter the limit: ");
                limit=tre.nextInt();
                employee e[]=new employee[limit];
                System.out.println(" ");
                for(i=0;i<limit;i++){
                System.out.println(" ");
                System.out.println("ENTER THE DEATAILS OF EMPLOYEE: "+(i+1));
                        e[i]=new employee();
                        e[i].get();}
                for(i=0;i<limit;i++){
                        System.out.println(" ");
                        System.out.println("DEATAILS OF EMPLOYEE: "+(i+1));
                        e[i].put();}
                System.out.println(" ");
                System.out.println("****Enter the value to be search**** ");
                val=tre.nextInt();
                for(i=0;i<limit;i++){
                        if(e[i].eno==val){
                                e[i].put();
```

```
                    break;}}
            else{
                    System.out.println("INVALID EMPLOYEE NO :) ");}}}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C02>javac employee.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C02>java employee

Enter the limit: 2


ENTER THE DEATAILS OF EMPLOYEE: 1
Enter the NO: 1
Enter the Name: Toy
Enter the price: 2500


ENTER THE DEATAILS OF EMPLOYEE: 2
Enter the NO: 2
Enter the Name: Ball
Enter the price: 500


DEATAILS OF EMPLOYEE: 1
NO:1
NAME:Toy
PRICE:2500

DEATAILS OF EMPLOYEE: 2
NO:2
NAME:Ball
PRICE:500

****Enter the value to be search****
2
NO:2
NAME:Ball
PRICE:500
```

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 areas1{
public void shape(int r){
                double area_of_circle=22/7*r*r;
                System.out.print("Area of circle is:"+area_of_circle);}
public void shape(int b,int h){
                double area_of_triangle=0.5*b*h;
                System.out.println("Area of triangle is: "+area_of_triangle);          }
public void shape(double y){
        double area_of_square=y*y;
                System.out.println("Area of square is: "+area_of_square);   }
public void shape(double l,double w){
        double area_of_rectangle=l*w;
                System.out.println("Area of rectangle is: "+area_of_rectangle);}
public void shape(double x,int f){
        double area_of_parallelogram=x*f;
                System.out.println("Area of parallelogram is: "+area_of_parallelogram);}
public void shape(double p,int q,double s){
        double area_of_trapezium=(0.5*(p+q))*s;
                System.out.println("Area of trapezium is: "+area_of_trapezium);}
public void shape(int t,double u){
        double area_of_ellipse=3.14*t*u;
                System.out.println("\nArea of ellipse is: "+area_of_ellipse);}
public static void main(String[] args){
                Scanner a=new Scanner(System.in);
                areas z=new areas();
                System.out.println("\n AREA OF CIRCLE ");
                System.out.print("Enter radius of circle: ");
                int r=a.nextInt();
                z.shape(r);
                System.out.print("\n\n AREA OF TRIANGLE ");
                System.out.print("\nEnter base of triangle: ");
                int b=a.nextInt();
                System.out.print("Enter height of triangle: ");
                int h=a.nextInt();
                z.shape(b,h);
                System.out.print("\n\n AREA OF SQUARE ");
                System.out.print("\nEnter side of square: ");
                double y=a.nextDouble();
                z.shape(y);
                System.out.print("\n\n AREA OF RECTANGLE ");
```

```
System.out.print("\nEnter length of rectangle:");
double l=a.nextDouble();
System.out.print("Enter width of rectangle:");
double w=a.nextDouble();
z.shape(l,w);
System.out.print("\n\n AREA OF PARALLELOGRAM ");
System.out.print("\nEnter base of parallelogram:");
double x=a.nextDouble();
System.out.print("Enter height of parallelogram:");
int f=a.nextInt();
z.shape(x,f);
System.out.print("\n\n AREA OF TRAPEZIUM ");
System.out.print("\nEnter length of parallel side 1 of trapezium:");
double p=a.nextDouble();
System.out.print("Enter length of parallel side 2 of trapezium:");
int q=a.nextInt();
System.out.print("Enter height of trapezium:");
double s=a.nextDouble();
z.shape(p,q,s);
System.out.print("\n\n AREA OF ELLIPSE ");
System.out.print("\nEnter minor axis of ellipse:");
int t=a.nextInt();
System.out.print("Enter major axis of ellipse:");
double u=a.nextDouble();
z.shape(t,u); }}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>javac areas1.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>java areas1

 AREA OF CIRCLE
Enter radius of circle: 2
Area of circle is:12.0

 AREA OF TRIANGLE
Enter base of triangle: 2
Enter height of triangle: 3
Area of triangle is:3.0

 AREA OF SQUARE
Enter side of square: 4
Area of square is:16.0

 AREA OF RECTANGLE
Enter length of rectangle:2
Enter width of rectangle:3
Area of rectangle is:6.0

 AREA OF PARALLELOGRAM
Enter base of parallelogram:2
Enter height of parallelogram:4
Area of parallelogram is:8.0

 AREA OF TRAPEZIUM
Enter length of parallel side 1 of trapezium:2
Enter length of parallel side 2 of trapezium:3
Enter height of trapezium:4
Area of trapezium is:10.0

 AREA OF ELLIPSE
Enter minor axis of ellipse:4
Enter major axis of ellipse:2
Area of ellipse is:25.12
```

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 inherits the properties of class employees and contains its own data members department, Subjects taught and constructors to initialize these data members and also include a 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 Employe{
        int Empid;
        String Name;
        int Salary;
        String Address;
        Employe(int a,String b,int c,String d){
                Empid = a;
                Name = b;
                Salary = c;
                Address = d;}  }
class Teacher extends Employe{
        String Department;
        String Subject;
        Teacher(int h,String i,int j,String k,String l,String m){
                super(h,i,j,k);
                Department = l;
                Subject = m;}
public void Display(){
        System.out.println("Enter the Employe id: "+ Empid);
        System.out.println("Enter the Employe name:"+ Name);
        System.out.println("Enter the Employe Salary:"+ Salary);
        System.out.println("Enter the Employe address:"+ Address);
        System.out.println("Enter the teacher department:"+ Department);
        System.out.println("Enter the teacher subjecy:"+ Subject);}}
public class EmployeQ2{
        public static void main(String[] args){
                int i;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the limit of arrays..");
                int n = sc.nextInt();
                Teacher s[] = new Teacher[n];
                for(i=0;i<n;i++){
                        System.out.println("Enter the employe id :");
                        int a = sc.nextInt();
                        System.out.println("Enter the employe name :");
                        String b = sc.next();
```

```
            System.out.println("Enter the employe salary :");
            int c = sc.nextInt();
            System.out.println("Enter the employe address :");
            String d = sc.next();
            System.out.println("Enter the teacher department :");
            String e = sc.next();
            System.out.println("Enter the teacher subject :");
            String f = sc.next();
            s[i] = new Teacher(a,b,c,d,e,f);          }
System.out.println("/n/n printing deatails...");
for(i=0;i<n;i++){
            s[i].Display();}}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C02>javac EmployeQ2.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C02>java EmployeQ2

Enter the limit of arrays: 2

Enter the employe id: 1
Enter the employe name: Jaimol
Enter the employe salary: 2500
Enter the employe address: Pala
Enter the teacher department: Computer
Enter the teacher subject: Ai

Enter the employe id: 2
Enter the employe name: Stefy
Enter the employe salary: 2300
Enter the employe address: Changanacherry
Enter the teacher department: Mathematics
Enter the teacher subject: Geometry


 printing deatails:

Enter the Employe id: 1
Enter the Employe name:Jaimol
Enter the Employe Salary:2500
Enter the Employe address:Pala
Enter the teacher department:Computer
Enter the teacher subject:Ai

Enter the Employe id: 2
Enter the Employe name:Stefy
Enter the Employe Salary:2300
Enter the Employe address:Changanacherry
Enter the teacher department:Mathematics
Enter the teacher subject:Geometry
```

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 contains constructors and methods to display the data members. Use an 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{
        int age;
        String name;
        String gender;
        String address;
        person(String pname, int ag, String gen, String adds){
                age = ag;
                name = pname;
                gender = gen;
                address = adds;}}
class employee extends person{
        int empid;
        String cmpnyname;
        String qualification;
        int salary;
        employee(String pname, int a,String gen, String ads, int eid, String cname, String qual,
int sal){
                super(pname,ag,gen,adds);
                empid = eid;
                cmpnyname = cname;
                qualification = qual;
                salary = sal;}}
class teachers extends employee{
        String department;
        int tchr_id;
        teachers(String pname, int a, String gen, String ads, int eid, String cname, String qual,
int sal, int tid, String dept){
                super(pname,a,gen,ads,eid,cname,qual,sal);
                department = dept;
                tchr_id = tid;}
        void put(){
                System.out.println("Name: "+name);
                System.out.println("Age: "+age);
                System.out.println("Gender: "+gender);
```

```
                System.out.println("Address: "+address);
                System.out.println("Employee ID: "+empid);
                System.out.println("Company name: "+cmpny_name);
                System.out.println("Qualification: "+qualification);
                System.out.println("Salary: "+salary);
                System.out.println("Teacher ID: "+tchr_id);
                System.out.println("Department: "+department);}}
public class people{
        public static void main(String[] args){
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the number of entries required: ");
                int n = sc.nextInt();
                teachers t[] = new teachers[n];
                for(int i=0;i<n;i++){
                        System.out.println("Enter the Name: ");
                        String na = sc.next();
                        System.out.println("Enter the Age: ");
                        int a = sc.nextInt();
                        System.out.println("Enter the Gender: ");
                        String g = sc.next();
                        System.out.println("Enter the Address: ");
                        String ad = sc.next();
                        System.out.println("Enter the Employee ID: ");
                        int eid = sc.nextInt();
                        System.out.println("Enter the Company name: ");
                        String c = sc.next();
                        System.out.println("Enter the Qualification: ");
                        String q = sc.next();
                        System.out.println("Enter the Salary: ");
                        int s = sc.nextInt();
                        System.out.println("Enter the Teacher ID: ");
                        int tid = sc.nextInt();
                        System.out.println("Enter the Department: ");
                        String d = sc.next();
                        t[i] = new teachers(na,a,g,ad,eid,c,q,s,tid,d);}
                for(int i=0;i<n;i++){
                        t[i].put();}}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>javac newteacher.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>java newteacher

Enter number of entries:1

Enter the name:Sapna
Enter the gender:Female
Enter address:Changanacherry
Enter your age:22
Enter employee id:101
Enter company name:Sutherland
Enter qualification:Degree
Enter salary:2300
Enter subject taught:Ai
Enter department:Computer
Enter teacherid:203

 Employee name:Sapna
 Employee gender:Female
 Employee address:Changanacherry
 Employee age:22
 Employee id:101
 Employee company name:Sutherland
 Employee qualification:Degree
 Employee salary:2300
 Subject handled by the teacher:Ai
 Teacher department:Computer
 Teacher id:203
```

Result

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

Experiment No.:12

Aim: Write a program that 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 nam;
        void publisher(){
        Scanner a=new Scanner(System.in);
        System.out.print("Enter publisher name:");
         nam=a.next();}}
class book extends publisher{
        String tit,auth,pric;
        int price;
        void book(){
        Scanner a=new Scanner(System.in);
        System.out.print("Enter book title:");
        tit=a.next();
        System.out.print("Enter author name:");
        auth=a.next();
        System.out.print("Enter book price:");
        price=a.nextInt();}}
class fiction extends book{
        void fiction(){
                System.out.println("Publisher Name:"+nam);
                System.out.println("Book Name:"+tit);
                System.out.println("Author Name: "+auth);
                System.out.println("Price: "+pric);}}
class literature extends book{
        void literature(){
                System.out.println("\n\nPublisher Name:"+nam);
                System.out.println("Book Name:"+tit);
                System.out.println("Author Name: "+auth);
                System.out.println("Price: "+pric);}}
class details{
        public static void main(String[] args){
                int i;
        Scanner a=new Scanner(System.in);
        System.out.print("\nEnter no of entries:");
        int n=a.nextInt();
        fiction dd[]=new fiction[n];
        for(i=0;i<n;i++){
                System.out.println("Details of book "+(i+1));
```

```
            dd[i]=new fiction();
            dd[i].publisher();
            dd[i].book();}
    for(i=0;i<n;i++){
            System.out.println("\nDetails of fiction book "+(i+1));
            dd[i].fiction();}
    System.out.print("\nEnter no of literature books:");
    int n1=a.nextInt();
    literature ll[]=new literature[n];
    for(i=0;i<n1;i++){
            System.out.println("Details of book "+(i+1));
            ll[i]=new literature();
            ll[i].publisher();
            ll[i].book();}
    for(i=0;i<n1;i++){
            System.out.println("Details of fiction book "+(i+1));
            ll[i].literature();}}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>javac details.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>java details

Enter no of entries:1
Details of book 1
Enter publisher name:Penguin India
Enter book title:Sensualist
Enter author name:Ruskin
Enter book price:120000

Details of fiction book 1
Publisher Name:Penguin
Book Name:Sensualist
Author Name: Ruskin
Price: null
Enter no of literature books:1
Details of book 1
Enter publisher name:Penguin India
Enter book title:Sensualist
Enter author name:Ruskin
Enter book price:1250000
Details of fiction book 1

Publisher Name:Penguin
Book Name:Sensualist
Author Name: Ruskin
Price: null
```

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 stget();
  public void stput();}
interface sports{
        public void spget();
        public void spput();}
class result1 implements student,sports{
        String name,s;
        int m1,m2,tot,p;
        public void stget(){
        Scanner a=new Scanner(System.in);
        System.out.print("\nEnter the name: ");
         name=a.next();
        System.out.print("Enter mark 1: ");
        m1=a.nextInt();
        System.out.print("Enter mark 2: ");
         m2=a.nextInt();
         tot=m1+m2;}
        public void stput(){
        System.out.println("\n**************************");
        System.out.println("Name: "+name);
        System.out.println("Mark 1: "+m1);
        System.out.println("Mark 2: "+m2);
        System.out.println("Total: "+tot);
        System.out.println("**************************");}
        public void spget(){
        Scanner a=new Scanner(System.in);
        System.out.print("\nEnter the sports item participated: ");
        s=a.next();
        System.out.print("Enter the position: ");
        p=a.nextInt();}
        public void spput(){
        System.out.println("\n**************************");
        System.out.println("Sports item participated: "+s);
        System.out.println("Position attained: "+p);
        System.out.println("**************************\n");}
        public static void main(String args[]){
                result1 obj=new result1();
```

```
obj.stget();
obj.stput();
obj.spget();
obj.spput();}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>javac result1.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>java result1

Enter the name: Tinu
Enter mark 1: 50
Enter mark 2: 80

**************************
Name: Tinu
Mark 1: 50
Mark 2: 80
Total: 130
**************************

Enter the sports item participated: Longjump
Enter the position: 2

*************************
Sports item participated: Longjump
Position attained: 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 implement the above interface. Create a menu driven program to find the area and perimeter of objects.

CO3

Implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
interface proto{
        public void get();
        public void area();
        public void per();}
class circle implements proto{
        double r,pe;
        double a;
        public void get(){
        Scanner a=new Scanner(System.in);
        System.out.print("\nEnter radius of circle: ");
         r=a.nextInt();}
         public void area(){
         a=3.14*r*r;
         System.out.println("**************************");
         System.out.println("Area of circle: "+a);
         System.out.println("**************************");}
          public void per(){
         pe=2*3.14*r;
         System.out.println("**************************");
         System.out.println("Perimeter of circle: "+pe);
         System.out.println("**************************");}}
class rectangle implements proto{
        int l,b,ar,p;
        public void get(){
        Scanner a=new Scanner(System.in);
        System.out.print("\nEnter length of rectngle: ");
         l=a.nextInt();
         System.out.print("Enter breadth of rectngle: ");
         b=a.nextInt();}
         public void area(){
         ar=l*b;
         System.out.println("**************************");
         System.out.println("Area of rectngle: "+ar);
         System.out.println("**************************");}
         public void per(){
         p=2*(l+b);
```

```
        System.out.println("**************************");
        System.out.println("Perimeter of rectngle: "+p);
        System.out.println("**************************");}}
class abc{
public static void main(String[] args)
        {int ch;
        do{
        Scanner a=new Scanner(System.in);
        circle c=new circle();
        rectangle r=new rectangle();
        System.out.println("Choose your choice: ");
        System.out.println("1.Area of circle \n2.Perimeter of circle \n3.Area of rectangle
\n4.Perimeter of rectangle \n5.Exit");
        ch=a.nextInt();
        switch (ch){
        case 1:
                c.get();
                c.area();
                break;
        case 2:
                c.get();
                c.per();
                break;
        case 3:
                r.get();
                r.area();
                break;
        case 4:
                r.get();
                r.per();
                break;
        case 5:
                break;
        default:
                System.out.println("***choose valid options***");
                break;}}
        while(ch!=5);}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>javac abc.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>java abc
Choose your choice:
1.Area of circle
2.Perimeter of circle
3.Area of rectangle
4.Perimeter of rectangle
5.Exit
1

Enter radius of circle: 2
**************************
Area of circle: 12.56
**************************
Choose your choice:
1.Area of circle
2.Perimeter of circle
3.Area of rectangle
4.Perimeter of rectangle
5.Exit
2

Enter radius of circle: 3
**************************
Perimeter of circle: 18.84
**************************
Choose your choice:
1.Area of circle
2.Perimeter of circle
3.Area of rectangle
4.Perimeter of rectangle
5.Exit
3

Enter length of rectngle: 2
Enter breadth of rectngle: 3
**************************
Area of rectngle: 6
**************************
Choose your choice:
1.Area of circle
2.Perimeter of circle
3.Area of rectangle
4.Perimeter of rectangle
5.Exit
```

Result

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

Experiment No.:15

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

Order No.
Date :

ProductId	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 bill{
        public void get();}
interface calculate{
        public void cal();}
class order implements bill,calculate{
        String name;
        int n,id,qy,price,total;
        public void get(){
                Scanner a=new Scanner(System.in);
                System.out.print("\nEnter product id: ");
                id=a.nextInt();
                System.out.print("Enter product name: ");
                name=a.next();
                System.out.print("Enter the quantity: ");
                qy=a.nextInt();
                System.out.print("Enter the unit price for the product: ");
                price=a.nextInt();}
        public void cal(){
         total=price*qy;}
        public void display(){
                System.out.print(id+ "\t\t" +name+"\t" +qy+"\t\t" +price+"\t" +total+"\n");}
        public static void main(String[] args){
                String date;
                int i,n,ord;
        Scanner a=new Scanner(System.in);
        System.out.print("\nEnter order no: ");
        ord=a.nextInt();
        System.out.print("Enter date: ");
        date=a.next();
        System.out.print("Enter the number of ordered products: ");
        n=a.nextInt();
        order odd[]=new order[n];
```

```
for(i=0;i<n;i++){
        odd[i]=new order();
        odd[i].get();
        odd[i].cal();}
System.out.println("----------------------------Bill----------------------------");
System.out.println("Order: "+ord);
System.out.println("Date: "+date);
System.out.println("------------------------------------------------------------");
System.out.println("PRODUCT ID \tNAME  \t\tQuantity  Unit price \tTotal");
for(i=0;i<n;i++){
        odd[i].display();}
System.out.println("\n------------------------------------------------------------");}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>javac order.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\CO3>java order

Enter order no: 1
Enter date: 25/11/1999
Enter the number of ordered products: 1

Enter product id: 01
Enter product name: CricketBat
Enter the quantity: 20
Enter the unit price for the product: 1200
----------------------------Bill----------------------------
Order: 1
Date: 25/11/1999
------------------------------------------------------------
PRODUCT ID      NAME            Quantity  Unit price    Total
1               CricketBat      20              1200    24000

------------------------------------------------------------
```

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. Use java.util package and Collection framework.

Procedure

Shapes.java

```
package Graphics;
import java.util.*;
public class Shapes implements Area{
double lr,lb,ra,th,tb,ta,saa,sa,cr,cc;
public void getRect(){
Scanner ab= new Scanner(System.in);
System.out.print("Enter the length of the rectangle: ");
lr=ab.nextInt();
System.out.print("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.print("Enter the height of the Triangle: ");
th=cb.nextInt();
System.out.print("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.print("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.print("Enter the radius of the Circle: ");
cc=sc.nextInt();}
public void Circle(){
cr=3.14*cc*cc;
System.out.println("Area of Square is "+cr);}
public static void main(String[] args){
```

```
Shapes o = new Shapes();
o.getRect();
o.Rectangle();
o.getTri();
o.Triangle();
o.getSqr();
o.Square();
o.getCrl();
o.Circle();}}
```

Area.java

```
package Graphics;
interface Area{
public void Rectangle();
public void Triangle();
public void Square();
public void Circle();
public void getRect();
public void getTri();
public void getSqr();
public void getCrl();}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4\Q1>javac Area.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4\Q1>javac -d . Area.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4\Q1>javac -d . Shapes.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4\Q1>java Graphics.Shapes
Enter the length of the rectangle: 2
Enter the breadth of the rectangle: 3
Area of Rectangle is 6.0
Enter the height of the Triangle: 2
Enter the base of the Triangle: 3
Area of Triangle angle is 3.0
Enter the Side of the Square: 4
Area of Square is 16.0
Enter the radius of the Circle: 2
Area of Square is 12.56

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4\Q1>
```

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. Use java.util package and Collection framework.

Procedure

basic.java

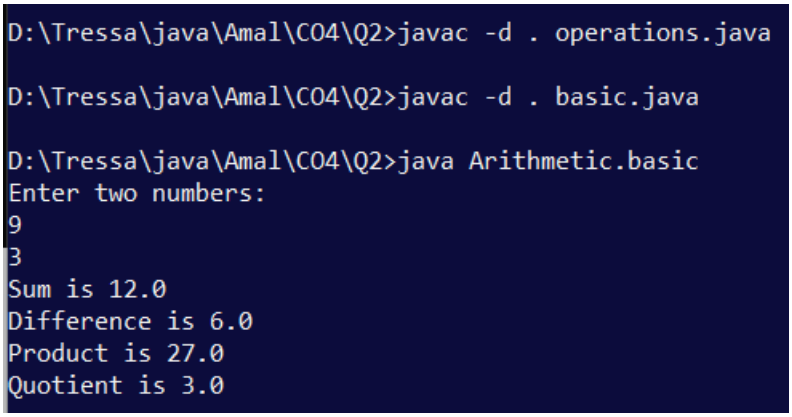
```
package Arithmetic;
import java.util.*;
public class basic implements operations{
   double a,b,ad,dif,mult,div;
   public void input(){
      Scanner ab=new Scanner(System.in);
      System.out.println("Enter two numbers:");
      a=ab.nextInt();
      b=ab.nextInt();}
   public void add(){
   ad=a+b;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

Operation.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 user name and password.

CO4

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

Procedure

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

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac CheckLoginCredential.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java CheckLoginCredential
Enter username :: Tressa
Enter password :: hello
Login Successful !!!
The finally statement is executed
```

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. Use java.util package and Collection framework.

Procedure

```
import java.util.Scanner;
import java.util.InputMismatchException;
public class avg{
   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

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac avg.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java avg
Enter the limit to calculate average: 4
Enter number:50
Enter number:12
Enter number:48
Enter number:78
Average: 47.0
```

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 a 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 (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 mtable{
   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

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac mtable.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java mtable
Multiplication Table of 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
Enter the limit to print prime numbers
19
First 19 prime numbers are :-
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>
```

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. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
public class Mythread6{
        public static void main(String args[]){
                Runnable r1 = new Runnable1();
                Thread t1 = new Thread(r1);
                t1.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.print("\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.print("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

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java Mythread6
Fibonacci series: 0112358132134
Enter the limit to print even numbers: 15
Even numbers upto15
2
4
6
8
10
12
14
```

Result

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

Experiment No.:22

Aim: 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 stack{
	public static void main( String args[]){
		int pos,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

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac stack.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java stack
STACK
1.ADD
2.REMOVE
3.DISPLAY
Select your Option: 1
Enter the Element to be inserted :- 2
Inserted to the Stack :-
STACK
1.ADD
2.REMOVE
3.DISPLAY
Select your Option: 1
Enter the Element to be inserted :- 3
Inserted to the Stack :-
STACK
1.ADD
2.REMOVE
3.DISPLAY
Select your Option: 1
Enter the Element to be inserted :- 4
Inserted to the Stack :-
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
[2, 4]
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 No.:23

Aim: Using generic methods perform Bubble sort.

CO4

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

Procedure

```
import java.util.Scanner;
class bubbleSort {
	public void sort(int a[],int n ){
		int t;
		for(int i=0;i<n-1;i++){
			for(int j=0;j<n-i-1;j++){
				if(a[j]>a[j+1]){
					t=a[j];
					a[j]=a[j+1];
					a[j+1]=t;}}}}
public static void main (String args[]){
	int n;
	Scanner obj =new Scanner(System.in);
	System.out.print("Enter the size: ");
	n=obj.nextInt();
	int b[]=new int[n];
	System.out.println("Enter the elements: ");
	for(int i=0;i<n;i++){
		b[i]=obj.nextInt();}
	System.out.println("Array before sort: ");
	for(int i=0;i<n;i++){
		System.out.print(b[i] + "\t");}
	CO4_9 rt =new CO4_9();
	rt.sort(b,n);
	System.out.println("\n" + "Array after sort: ");
	for(int i=0;i<n;i++){
		System.out.print(b[i] + "\t");}}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac bubbleSort.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java bubbleSort
Enter the size: 4
Enter the elements:
2
4
5
7
Array before sort:
2       4       5       7
Array after sort:
2       4       5       7
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>
```

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. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
class Arraylist{
	public static void main(String args[]){
		Scanner value = new Scanner(System.in);
		int ch;
		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

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac Arraylist.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java Arraylist
1.Add
2.Remove
3.Display the element in n=th index
4.Display
5.Remove All
Select your Option:  1
Enter the element to be inserted: 1
1.Add
2.Remove
3.Display the element in n=th index
4.Display
5.Remove All
Select your Option:  1
Enter the element to be inserted: 2
1.Add
2.Remove
3.Display the element in n=th index
4.Display
5.Remove All
Select your Option:  1
Enter the element to be inserted: 3
1.Add
2.Remove
3.Display the element in n=th index
4.Display
5.Remove All
Select your Option:  3
Enter the index of element: 2
3
1.Add
2.Remove
3.Display the element in n=th index
4.Display
5.Remove All
Select your Option:  4
[1, 2, 3]
1.Add
2.Remove
3.Display the element in n=th index
4.Display
5.Remove All
Select your Option:
```

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. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
public class Linkedlist{
        public static void main(String[] jai){
        Scanner sc = new Scanner(System.in);
        LinkedList<String> list=new LinkedList<String>();
        System.out.print("\nEnter the limit: ");
        int limit = sc.nextInt();
        System.out.println("Enter the elements: ");
        for(int i=0;i<limit;i++){
                        list.add(sc.next());}
        System.out.println("list : "+list);
        list.clear();
        System.out.println("After removing all the elements: "+list);}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac Linkedlist.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java Linkedlist

Enter the limit: 3
Enter the elements:
2
4
5
list : [2, 4, 5]
After removing all the elements: []
```

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. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
import java.util.Stack;
public class stackdemo{
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                int n,i;
                Stack<String> obj = new Stack<String>();
                System.out.println("Enter the limit: ");
                n = sc.nextInt();
                for(i=0;i<n;i++){
                        System.out.println("Enter the stack elements: ");
                        obj.add(sc.next());}
                System.out.println("Stack is: " + obj);
                System.out.println("Enter the position of element to be popped: ");
                int p = sc.nextInt();
                System.out.println("New Stack: "+ obj);}}
```

Output Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac removeOBJ.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java removeOBJ
Enter the Size of Stack: 4
Enter the elements: 1
Enter the elements: 2
Enter the elements: 3
Enter the elements: 4
Stack = [1, 2, 3, 4]
Enter the position: 2
Stack = [1, 2, 4]
```

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. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
class priorityQueue{
	public static void main(String args[]){
		Scanner value = new Scanner(System.in);
		PriorityQueue<Integer> pq = new PriorityQueue<>();
		int ch;
		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: int val;
					System.out.print("Enter the 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 Screenshot

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac priorityQueue.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java priorityQueue
1.Add
2.Peek
3.Poll
Select your Option:  1
Enter the element: 2
1.Add
2.Peek
3.Poll
Select your Option:  1
Enter the element: 3
1.Add
2.Peek
3.Poll
Select your Option:  1
Enter the element: 4
1.Add
2.Peek
3.Poll
Select your Option:  2
2
1.Add
2.Peek
3.Poll
Select your Option:  3
2
1.Add
2.Peek
3.Poll
Select your Option:
```

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

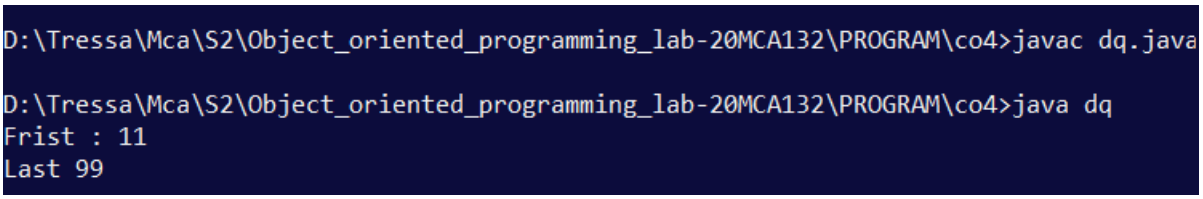
CO4

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

Procedure

```
import java.util.*;
import java.util.Deque;
public class dq{
        public static void main (String [] args){
                Deque<Integer> deque = new ArrayDeque<>();
                deque.addFirst(11);
                deque.addLast(99);
                int frist = deque.removeFirst();
                int last = deque.removeLast();
                System.out.println("Frist : " +frist + "\n"+"Last " + last);
                }}
```

Output Screenshot



Result

The program was executed and the result was successfully obtained. Thus, CO4 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. 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>();
		int n,u,l;
		System.out.print("Enter the Number of elements First set: " );
		n=obj.nextInt();
		for (int i =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 (int i =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:

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac CO4_15.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java CO4_15
Enter the Number of elements First set: 3
Enter the elements: 1
Enter the elements: 2
Enter the elements: 3
[1, 2, 3]
Enter the Number of elements second set: 4
Enter the elements: 5
Enter the elements: 6
Enter the elements: 7
Enter the elements: 8
[5, 6, 7, 8]
Union : true
[1, 2, 3, 5, 6, 7, 8]
Intrsection : true
[5, 6, 7, 8]
Differeance : true
[]
```

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 sets.

CO4:

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

Procedure:

```
import java.util.*
public class CO4_16 {
   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:

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac CO4_16.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java CO4_16
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. 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:

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac CO4_17.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java CO4_17
Initial Map: {Bob=30, Alice=25, Charlie=35, Dave=40}
Map after changing value for 'Alice': {Bob=30, Alice=26, Charlie=35, Dave=40}
Map after removing 'Charlie': {Bob=30, Alice=26, Dave=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. Use java.util package and Collection framework.

Procedure:

```
import java.util.*;
import java.util.stream.*;
public class CO4_18 {
  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 (HashMap to TreeMap) " + treeMap); }}
```

Output Screenshot:

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>javac CO4_18.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\co4>java CO4_18
HashMap = {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}
TreeMap (HashMap to 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:

1) **CO5P1.java // Java File**

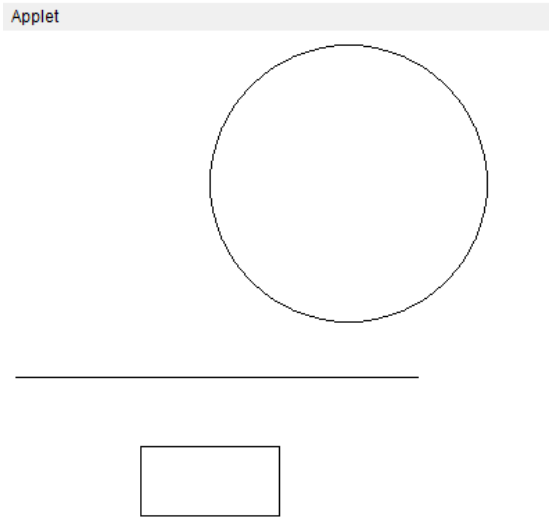
```
import java.applet.Applet;
import java.awt.Graphics;

public class CO5P1 extends Applet{
        public void paint(Graphics g){
                g.drawArc(150,10,200,200,0,360);
                g.drawLine(10,250,300,250);
                g.drawRect(100,300,100,50);}}
```

2) **CO5P1.html // HTML File**

```
<html>
        <body>
                <applet code="CO5P1.java" 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.: 34

Aim:

Program to find a maximum of three numbers using AWT.

CO5:

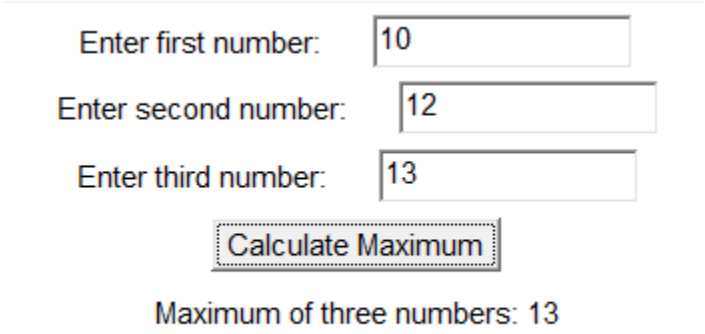
Develop applications to handle events using applets.

Procedure:

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class CO5P2 extends Frame implements ActionListener {
	private TextField num1Field, num2Field, num3Field;
	private Label resultLabel;
	public CO5P2() {
		setTitle("Maximum of Three Numbers");
		setSize(300, 200);
		setLayout(new FlowLayout());
		Label num1Label = new Label("Enter first number: ");
		num1Field = new TextField(10);
		Label num2Label = new Label("Enter second number: ");
		num2Field = new TextField(10);
		Label num3Label = new Label("Enter third number: ");
		num3Field = new TextField(10);
		Button calculateButton = new Button("Calculate Maximum");
		calculateButton.addActionListener(this);
		resultLabel = new Label("Result will be displayed here.");
		add(num1Label);
		add(num1Field);
		add(num2Label);
		add(num2Field);
		add(num3Label);
		add(num3Field);
		add(calculateButton);
		add(resultLabel);
		setVisible(true);}
	@Override
	public void actionPerformed(ActionEvent e) {
		int num1 = Integer.parseInt(num1Field.getText());
		int num2 = Integer.parseInt(num2Field.getText());
		int num3 = Integer.parseInt(num3Field.getText());

		int max = Math.max(num1, Math.max(num2, num3));
		resultLabel.setText("Maximum of three numbers: " + max);}
	public static void main(String[] args) {
		new CO5P2();}}
```

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:

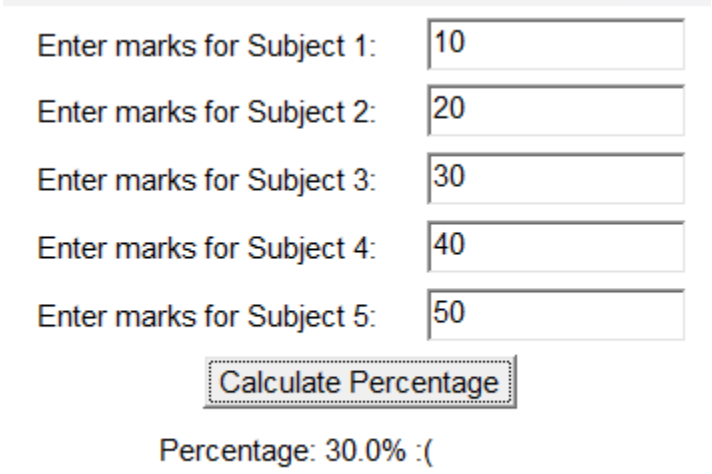
Develop applications to handle events using applets.

Procedure:

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
public class CO5P3 extends Frame implements ActionListener {
   private TextField[] marksFields;
   private Button calculateButton;
   private Label resultLabel;
   public CO5P3() {
      setTitle("Percentage Calculator");
      setSize(300, 250);
      setLayout(new FlowLayout());
      marksFields = new TextField[5];
      for (int i = 0; i < marksFields.length; i++) {
         Label label = new Label("Enter marks for Subject " + (i + 1) + ": ");
         marksFields[i] = new TextField(10);
         add(label);
         add(marksFields[i]);}
      calculateButton = new Button("Calculate Percentage");
      calculateButton.addActionListener(this);
      resultLabel = new Label("Result will be displayed here.");
      addWindowListener(new WindowAdapter() {
         public void windowClosing(WindowEvent e) {
            dispose();} });
      add(calculateButton);
      add(resultLabel);
      setVisible(true);}
   public void actionPerformed(ActionEvent e) {
      try {
         int totalMarks = 0;
         for (TextField marksField : marksFields) {
            int marks = Integer.parseInt(marksField.getText());
            totalMarks += marks;  }
         double percentage = (double) totalMarks / (marksFields.length * 100) * 100;
         resultLabel.setText("Percentage: " + percentage + "%");
         if (percentage > 50) {
            resultLabel.setText(resultLabel.getText() + " :)"); // Happy face
```

```
        } else {
            resultLabel.setText(resultLabel.getText() + " :("); // Sad face }
    } catch (NumberFormatException ex) {
        resultLabel.setText("Invalid input. Please enter valid marks.");}}
public static void main(String[] args) {
    new CO5P3();}}
```

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.

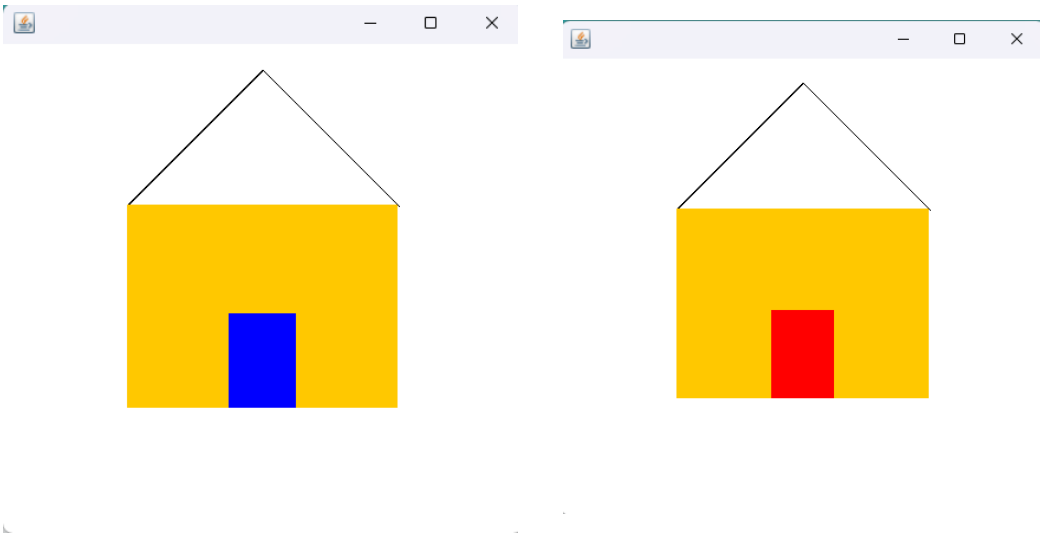
CO5:

Develop applications to handle events using applets.

Procedure:

```
import java.awt.*;
import java.awt.event.*;
import java.awt.Graphics;
public class Home extends Frame{
        private Color col = Color.red;
        Home(){
                addMouseListener(new MouseAdapter(){
                        public void mouseClicked(MouseEvent m){
                        col = Color.blue;
                        repaint();
                        }});
                setSize(400,400);
                setLayout(null);
                setVisible(true);
                addWindowListener(new WindowAdapter(){
                        public void windowClosing(WindowEvent e){
                                dispose();}});  }
        public void paint(Graphics g){
                g.drawLine(100,150,200,50);
                g.drawLine(200,50,300,150);
                g.setColor(Color.orange);
                g.fillRect(100,150,200,150);
                g.setColor(col);
                g.fillRect(175,230,50,70);}
        public static void main(String args[]){
                Home h = new Home();}}
```

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.

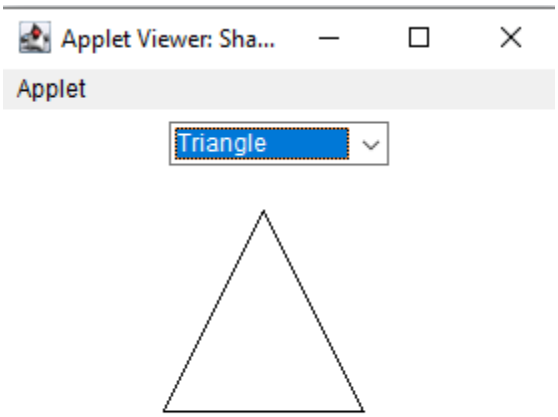
Procedure:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class ShapeChoice extends Applet implements ItemListener{
        int a;
        Choice ch;
        public void init(){
                ch = new Choice();
                ch.addItem("Select a shape");
                ch.addItem("Circle");
                ch.addItem("Rectangle");
                ch.addItem("Square");
                ch.addItem("Triangle");
                add(ch);
                ch.addItemListener(this);}
        public void itemStateChanged(ItemEvent e){
                a=ch.getSelectedIndex();
                repaint();}
        public void paint(Graphics g){
                if (a==1)
                        g.drawOval(50,50,100,100);
                if (a==2)
                        g.drawRect(50,50,200,100);
                if (a==3)
                        g.drawRect(50,50,100,100);
                if (a==4){
                        g.drawLine(130,50,80,150);
                        g.drawLine(130,50,180,150);
                        g.drawLine(80,150,180,150);
                }}}
/*<html>
</head><title>Shape Choice</title></head>
<body>
<applet code="ShapeChoice.class" width="500" height="500"></applet>
</body>
</html>*/
```

Output Screenshot

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

C:\java\bin>appletviewer ShapeChoice.java
```



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:

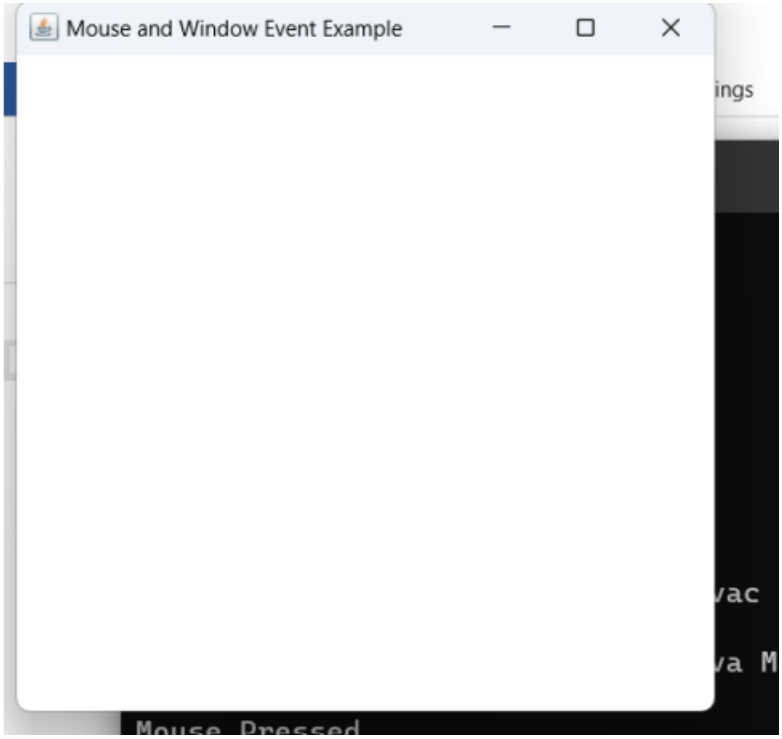
Develop applications to handle events using applets.

Procedure:

```
import java.awt.*;
import java.awt.event.*;
public class MouseEventExample extends Frame implements MouseListener,
WindowListener {
MouseEventExample() {
addMouseListener(this);
addWindowListener(this);
setSize(400, 400);
setTitle("Mouse and Window Event Example");
setVisible(true);}
public void mouseClicked(MouseEvent e) {
System.out.println("Mouse Clicked at (" + e.getX() + ", " + e.getY() + ")");}
public void mouseEntered(MouseEvent e) {
System.out.println("Mouse Entered");}
public void mouseExited(MouseEvent e) {
System.out.println("Mouse Exited");}
public void mousePressed(MouseEvent e) {
System.out.println("Mouse Pressed");}
public void mouseReleased(MouseEvent e) {
System.out.println("Mouse Released");}
public void windowClosing(WindowEvent e) {
System.out.println("Window Closing");
dispose(); // Close the window}
public void windowActivated(WindowEvent e) {}
```

```
public void windowClosed(WindowEvent e) {}
public void windowDeactivated(WindowEvent e) {}
public void windowDeiconified(WindowEvent e) {}
public void windowIconified(WindowEvent e) {}
public void windowOpened(WindowEvent e) {}
public static void main(String[] args) {
new MouseEventExample();}}
```

Output Screenshot



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:

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 read{
   public static void main(String[] args) throws IOException {
        FileWriter writer = new FileWriter("co6q2.txt",true);
        writer.write("new file is created");
        writer.close();
        FileReader reader = new FileReader("co6q2.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();} }
```

Output Screenshot:

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>javac read.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>java read
Data read from the file
new file is created
```

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:

Develop 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("file1.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:

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>javac file.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>java file
Successfully copied one file to another
```

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:

Develop applications using files and networking concepts.

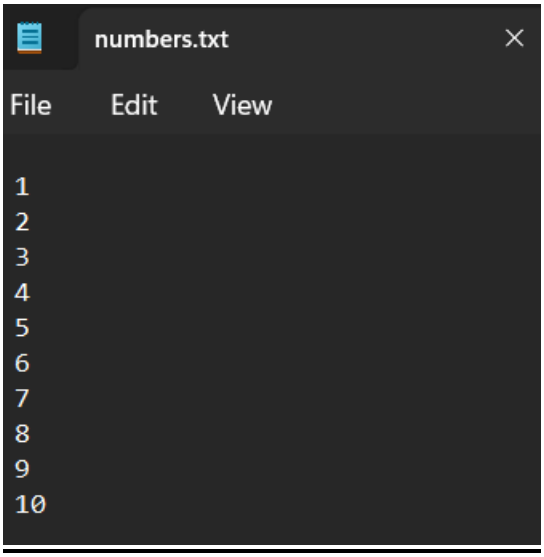
Procedure:

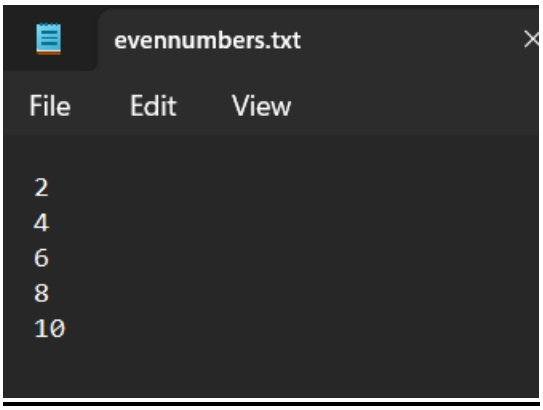
```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
public class oddeven {
public static void main(String[] args) {
try {
FileReader fr = new FileReader ("numbers.txt");
BufferedReader br = new BufferedReader(fr);
File file1 = new File("oddnumbers.txt");
FileWriter fw1 = new FileWriter(file1);
File file2 = new File("evennumbers.txt");
FileWriter fw2 = new FileWriter(file2);
String num;
while ((num = br.readLine()) != null) {
if (Integer.parseInt(num) % 2 == 0) {
fw2.write(num + "\n");}
else {
fw1.write(num + "\n");}}
fw1.close();
fw2.close();}
catch (Exception e) {
System.out.println("Error");}}}
```

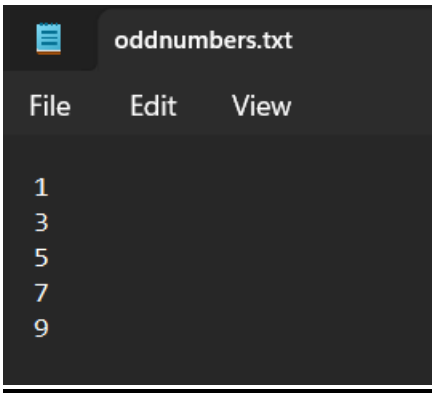
Output Screenshot:

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>javac oddeven.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>java oddeven
```







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:

Develop applications using files and networking concepts.

Procedure:

- **CO6P6_Client.java**

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;
public class CO6P6_Client{
        public static void main(String []args) throws IOException{
                Scanner sc = new Scanner(System.in);
                DatagramSocket dataSocket = new DatagramSocket();
                InetAddress ip = InetAddress.getLocalHost();
                System.out.println("Host Name with IP Address :"+ip);
                byte buff[] = null;
                while(true){
                        System.out.print("Chat: ");
                        String chat = sc.next();
                        buff = chat.getBytes(); // Converting inputed char to Bytes
                                        //DatagramPacket(Bytes, Byte Length, Host IP,
Port Num); Setting up the Packet to be Send
                        DatagramPacket packetSend = new DatagramPacket(buff,
buff.length, ip, 1234);
                        dataSocket.send(packetSend); // Sending the Packet

                        if(chat.equals("bye"))
                                break;}}}
```

- **CO6P6_Server.java**

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
public class CO6P6_Server{
        public static void main(String[] args) throws IOException{
// Step 1 : Create a socket to listen at port 1234
                DatagramSocket ds = new DatagramSocket(1234);
                byte[] receive = new byte[65535];
                DatagramPacket DpReceive = null;
                while (true){
                        // Step 2 : create a DatgramPacket to receive the data.
                        DpReceive = new DatagramPacket(receive, receive.length);
                        // Step 3 : revieve the data in a byte buffer.
```

```
                    ds.receive(DpReceive);
                    System.out.println("Client:-" + data(receive));
                    // Exit the server if the client sends "bye"
                    if (data(receive).toString().equals("bye")){
                            System.out.println("Client sent bye.....EXITING");
                            break;}
                    // Clear the buffer after every message.
                    receive = new byte[65535];}}
// A utility method to convert the byte array
// data into a string representation.
public static StringBuilder data(byte[] a){
        if (a == null)
                return null;
        StringBuilder ret = new StringBuilder();
        int i = 0;
        while (a[i] != 0){
                ret.append((char) a[i]);
                i++;}
        return ret;}}
```

Output Screenshot:

Client Side

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>javac q6_Client.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>java q6_Client
Host Name with IP Address :DESKTOP-ODRD5I6/127.0.0.1
Chat: hello
Chat: hai
Chat: Jaimol
Chat: bye

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>
```

Server Side

```
D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>javac q6_Server.java

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>java q6_Server
Client:-hello
Client:-hai
Client:-Jaimol
Client:-bye
Client sent bye.....EXITING

D:\Tressa\Mca\S2\Object_oriented_programming_lab-20MCA132\PROGRAM\C06>
```

Result:

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