**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 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 price");

price=sc.nextInt();

}

public void put(){

System.out.println("detail of product");

System.out.println("code of product"+pcode);

System.out.println("name of product"+pname);

System.out.println("price of product"+price);

}

public static void main(String args[]){

product p1=new product();

product p2=new product ();

product p3=new product();

p1.get();

p2.get();

p3.get();

p1.put();

p2.put();

p3.put();

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

{

System.out.println("lowest price "+p1.price);

}

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

{

System.out.println("lowest price is"+p2.price);

}

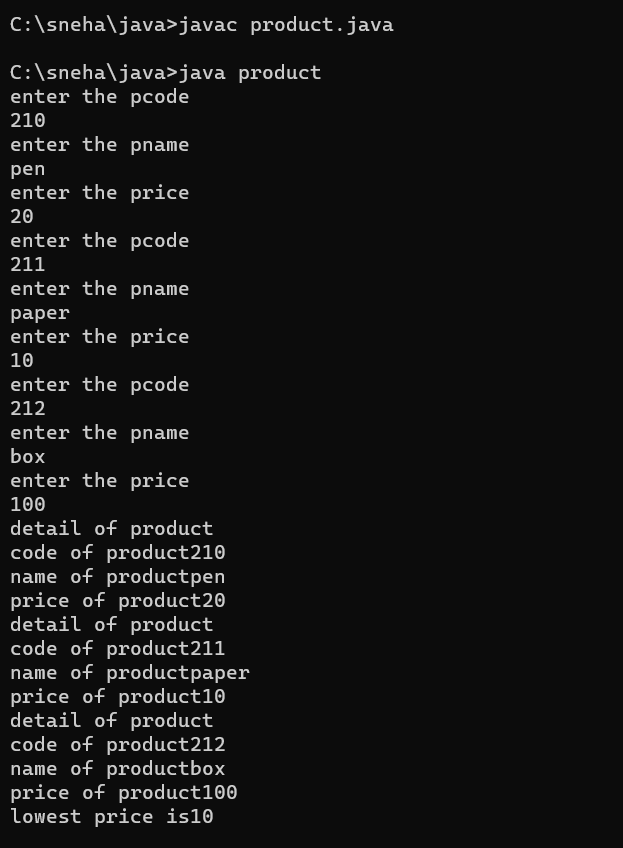
else

System.out.println("lowest price is"+p3.price);

}

}

**Output Screenshot**

****

**Result**

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

**Experiment No.: 2**

**Aim**

Read 2 matrices from the console and perform matrix addition.

**CO1**

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

**Procedure**

import java.util.Scanner;

public class MatrixAddition

{

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

// Read the dimensions of the matrices

System.out.print("Enter the number of rows: ");

int rows = s.nextInt();

System.out.print("Enter the number of columns: ");

int columns = s.nextInt();

// Create the matrices

int[][] matrix1 = new int[rows][columns];

int[][] matrix2 = new int[rows][columns];

int[][] sumMatrix = new int[rows][columns];

// Read the elements of the first matrix

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

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < columns; j++)

{

matrix1[i][j] = s.nextInt();

}

}

// Read the elements of the second matrix

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

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < columns; j++)

{

matrix2[i][j] = s.nextInt();

}

}

// Perform matrix addition

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

// Display the sum matrix

System.out.println("Sum of the matrices:");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

System.out.print(sumMatrix[i][j] + " ");

}

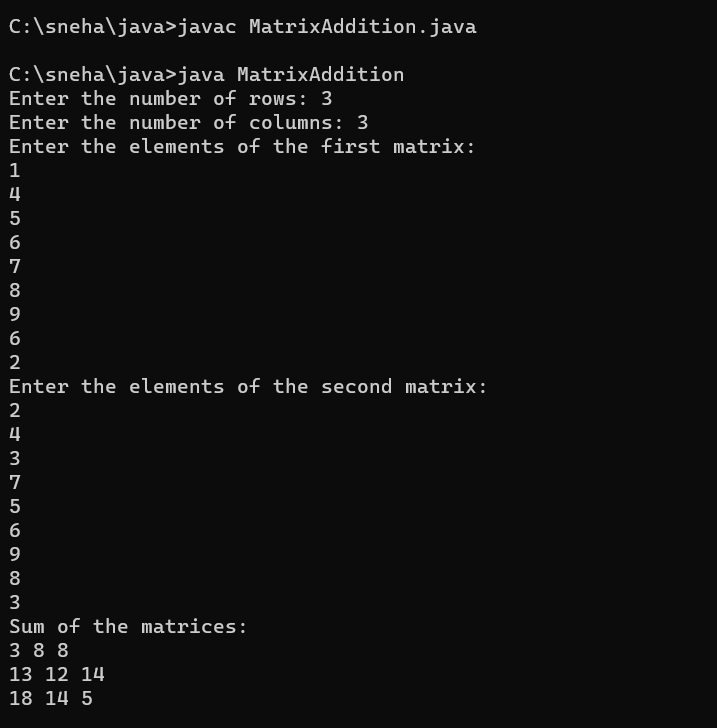
System.out.println();

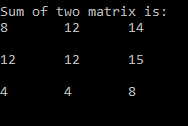
}

}

}

**Output Screenshot**





**Result**

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

**Experiment No.: 3**

**Aim**

Add complex numbers

**CO1**

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

**Procedure**

import java.util.\*;

public class Complex{

public static void main(String[] args)

{

Scanner Snr = new Scanner(System.in);

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

int r1 = Snr.nextInt();

System.out.println("Enter the coefficient of the first imaginary constant:");

int i1 = Snr.nextInt();

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

int r2 = Snr.nextInt();

System.out.println("Enter the coefficient of the second imaginary constant: ");

int i2 = Snr.nextInt();

System.out.println("The first imaginary number is "+r1+" + "+i1+"i");

System.out.println("The second imaginary number is "+r2+" + "+i2+"i");

int r3=r1+r2;

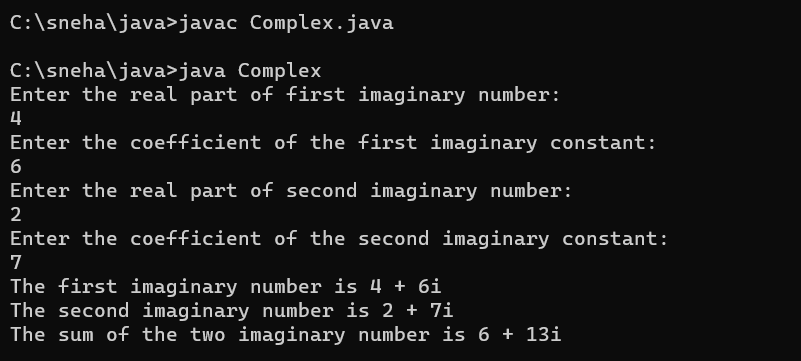
int i3=i1+i2;

System.out.println("The sum of the two imaginary number is "+ r3 +" + "+ i3 +"i");

}

}

**Output Screenshot**

****

**Result**

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

**Experiment No.: 4**

**Aim**

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

**CO1**

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

**Procedure**

import java.util.\*;

public class SymmetricMatrix{

public static void main(String[] args){

Scanner Snr = new Scanner(System.in);

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

int sz = Snr.nextInt();

int Arr[][] = new int[sz][sz];

int Arr1[][] = new int[sz][sz];

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

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

for(int j=0;j<sz;j++){

Arr[i][j] = Snr.nextInt();

}

}

System.out.println("The matrix: ");

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

for(int j=0;j<sz;j++){

System.out.print(Arr[i][j] + " ");

}

System.out.println(" ");

}

System.out.println("The Transpose of the matrix: ");

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

for(int j=0;j<sz;j++){

Arr1[i][j]=Arr[j][i];

}

}

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

for(int j=0;j<sz;j++){

System.out.print(Arr1[i][j] + " ");

}

System.out.println(" ");

}

int flag=0;

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

for(int j=0;j<sz;j++){

if(Arr[i][j] != Arr1[i][j]){

flag=1;

break;

}

}

}

if(flag==1){

System.out.println("The matrix is not symmetric");

}

else{

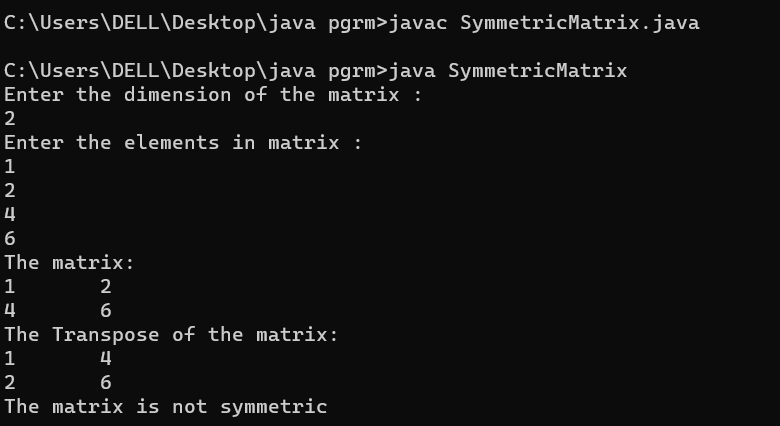
System.out.println("The matrix is symmetric");

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 5**

**Aim**

Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

**CO1**

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

**Procedure**

import java.util.\*;

public class Cpu

{

int price;

public void price\_input()

{

Scanner in = new Scanner(System.in);

System.out.println("Enter the price of CPU : ");

price = in.nextInt();

}

public void print()

{

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

}

public class Processor

{

int no\_of\_cores;

String man;

Processor()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the No:of cores : ");

no\_of\_cores = sc.nextInt();

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

man = sc.next();

}

void print()

{

System.out.println(" No:of cores : "+no\_of\_cores);

System.out.println(" Manufacturer : "+man);

}

}

static class RAM

{

static int mem\_size;

static String man;

RAM()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the memory size ofRAM : ");

mem\_size = sc.nextInt();

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

man = sc.next();

}

void print()

{

System.out.println(" Memory Size : "+mem\_size);

System.out.println(" Manufacturer : "+man);

}

}

public static void main(String args[])

{

Cpu c = new Cpu();

Cpu.Processor p = c.new Processor();

Cpu.RAM r = new Cpu.RAM();

c.price\_input();

p.print();

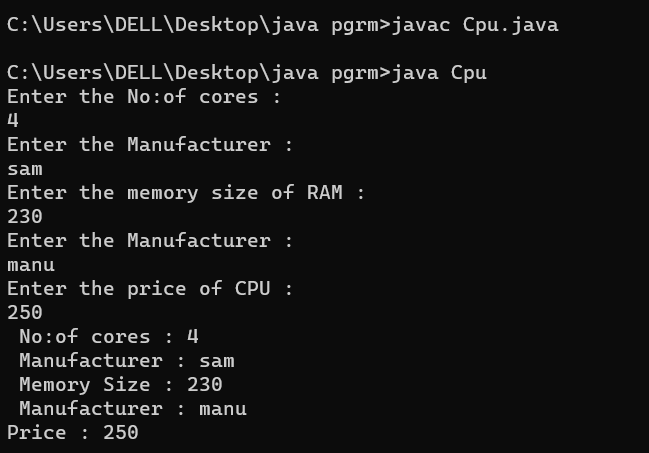
r.print();

c.print();

}

}

**Output Screenshot**

****

**Result**

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

**Experiment No.: 6**

**Aim**

Program to Sort strings

**CO2**

Implement arrays and strings.

**Procedure**

import java.util.\*;

public class sort {

public static void main(String[] args) {

Scanner value = new Scanner(System.in);

int i,j;

String temp;

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

int size = value.nextInt();

String array[] = new String[size];

System.out.println("Enter the elements of Array");

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

{

array[i] = value.nextLine();

}

System.out.println("The Array is: ");

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

{

System.out.println(array[i]);

}

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

{

for (j=i+1;j<size;j++)

if (array[i].compareTo(array[j])>0)

{

temp = array[i];

array[i] = array[j];

array[j] = temp;

}

}

System.out.println("The Sorted Array is: ");

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

{

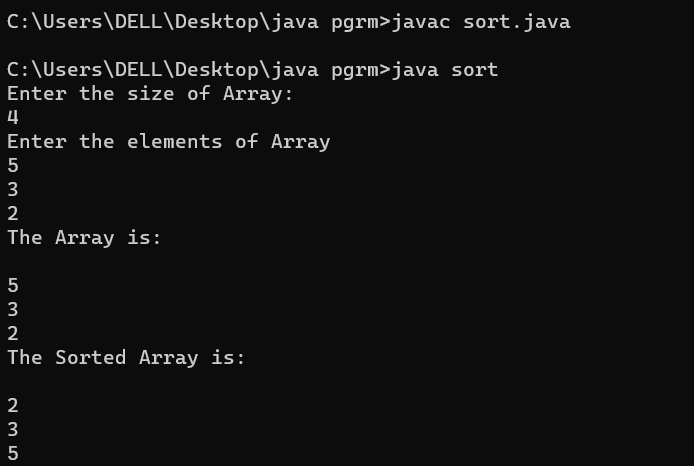
System.out.println(array[i]);

}

}

}

**Output Screenshot**

****

**Result**

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

**Experiment No.: 7**

**Aim**

Search an element in an array.

**CO2**

Implement arrays and strings.

**Procedure**

import java.util.\*;

public class FindvalBin{

public static void main(String[] args){

Scanner Snr = new Scanner(System.in);

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

int sz = Snr.nextInt();

int Arr[] = new int[sz];

System.out.println("Enter the elements in array: ");

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

{

Arr[i] = Snr.nextInt();

}

System.out.print("The array: ");

System.out.print("[ ");

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

{

System.out.print(Arr[i]+" ");

}

System.out.println("]");

int temp=0;

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

{

for(int j=i+1;j<sz;j++)

{

if(Arr[i] > Arr[j])

{

temp= Arr[i];

Arr[i] = Arr[j];

Arr[j] = temp;

}

}

}

System.out.print("The sorted array: ");

System.out.print("[ ");

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

{

System.out.print(Arr[i]+" ");

}

System.out.println("]");

System.out.println("Enter the value to be searched: ");

int val = Snr.nextInt();

int flag=0;

int mid=sz/2;

if(val == Arr[mid]){

int ps=sz+1;

System.out.println("Value is found and is in position "+ ps);

flag=1;

}

else if(val>Arr[mid])

{

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

if(Arr[i] == val){

int ps=i+1;

System.out.println("Value is found and is in position "+ ps);

flag=1;

}

}

}

else

{

for(int i=mid;i>0;i--){

if(Arr[i] == val){

int ps=i+1;

System.out.println("Value is found and is in position "+ ps);

flag=1;

}

}

}

if(flag!=1)

{

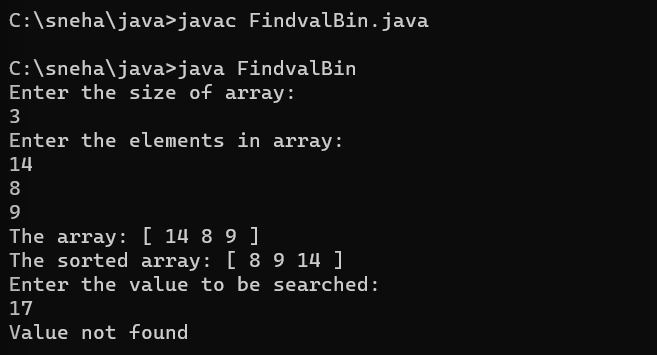
System.out.println("Value not found");

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 8**

**Aim**

Perform string manipulations

**CO2**

Implement arrays and strings.

**Procedure**

import java.util.Scanner;

import java.lang.\*;

public class Manipulation{

public static void main(String [] args){

int a;

String b,c;

Scanner sc = new Scanner(System.in);

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

b = sc.nextLine();

while(true)

{

System.out.println("\n MENU:\n 1.String Length.\2.Uppercase.\n3.Lowercase.\n4.Concatenate.\n 5.Character index.\n6.Exit.");

System.out.print("\n Enter your option : ");

a = sc.nextInt();

switch(a)

{

case 1:

System.out.println(" String length = "+b.length());

break;

case 2:

System.out.println(" String in uppercase = "+b.toUpperCase());

break;

case 3:

System.out.println(" String in lowercase = "+b.toLowerCase());

break;

case 4:

{

System.out.print(" Enter the string to be concatenate = ");

c = sc.next();

System.out.println(" Concatenated string = "+b.concat(c));

break;

}

case 5:

{

System.out.print(" Enter the Character to be searched in the given string = ");

c = sc.next();

System.out.println(" The character is found at "+(b.indexOf(c)+1)+".");

break;

}

case 6: System.exit(0);

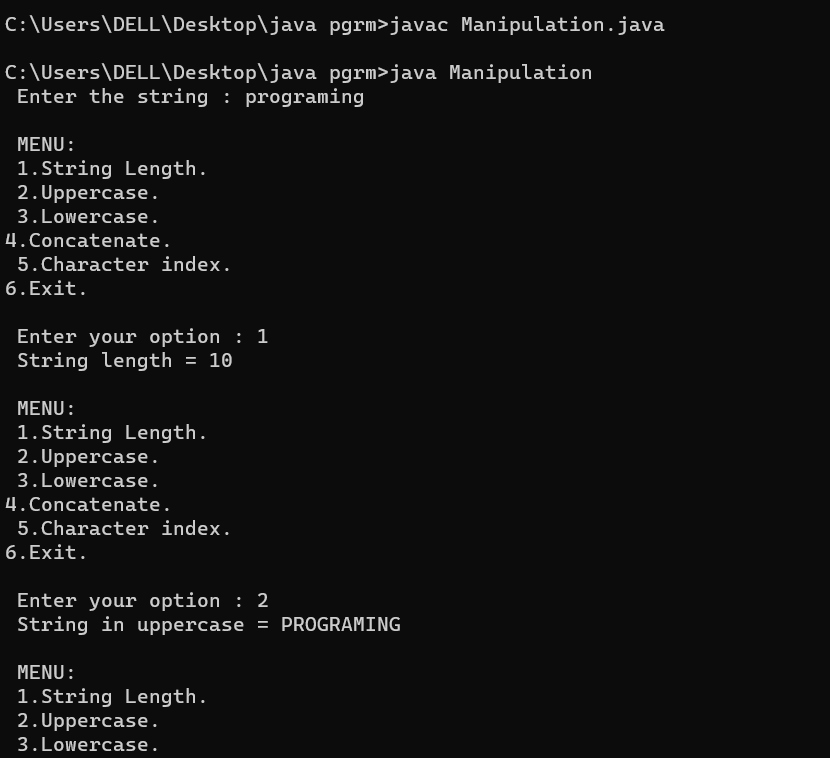
}

}

}

}

**Output Screenshot**

****

**Result**

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

**Experiment No.: 9**

**Aim**

Program to create a class for Employee having attributes eNo, eName , eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

**CO2**

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)

{

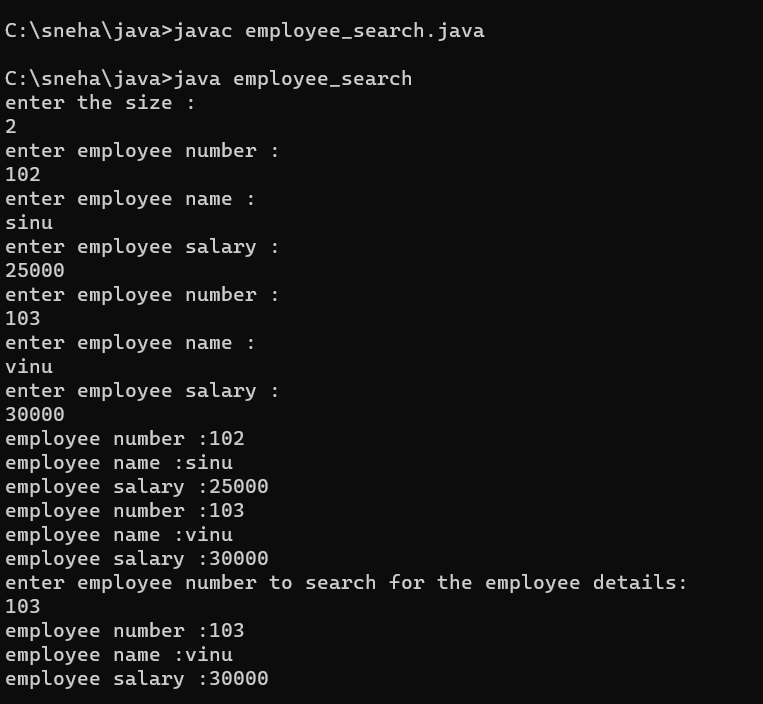
System.out.println("not found");

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 10**

**Aim**

Area of different shapes using overloaded functions

**CO3**

Implement object-oriented concepts like inheritance, overloading and interfaces

**Procedure**

import java.util.Scanner;

class Shape1{

public void area(double r){

System.out.println("Area of Circle "+ (3.14\*r\*r));

}

public void area(int b,int h){

System.out.println("Area of Triangle "+ (.5\*(b\*h)));

}

public void area(int a){

System.out.println("Area of Square "+ (a\*a));

}

public void area(int x,float y)

{

System.out.println("Area Of Reactangle "+(x\*y));

}

public static void main(String args []){

int b,c,x;

double a;

float y;

Scanner f=new Scanner(System.in);

Shape1 s =new Shape1();

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

a=f.nextDouble();

s.area(a);

System.out.println("Enter the Base of Triangle :");

c=f.nextInt();

System.out.println("Enter the Height of Triangle :");

b=f.nextInt();

s.area(c,b);

System.out.println("Enter the Length of Reactangle :");

x=f.nextInt();

System.out.println("Enter the width of Rectangle :");

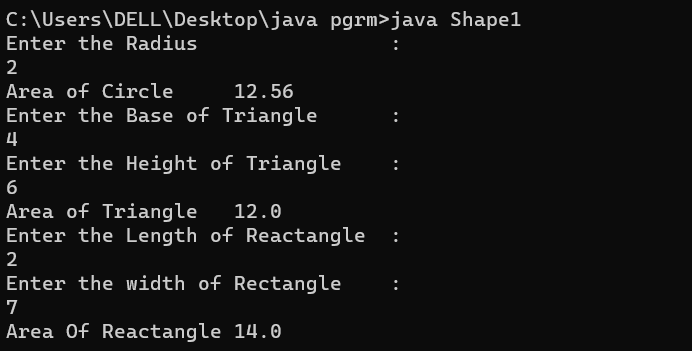
y=f.nextFloat();

s.area(x,y);

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 11**

**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.Scanner;

class Employee {

int empid;

String name;

int salary;

String address;

Employee(int a, String b, int c, String d) {

empid = a;

name = b;

salary = c;

address = d;

}

}

class Teacher extends Employee {

String department;

String subject;

Teacher(int l, String m, int n, String o, String p, String q) {

super(l, m, n, o);

department = p;

subject = q;

}

public void display() {

System.out.println("\nEmployee Id :" + empid);

System.out.println("Employee Name :" + name);

System.out.println("Employee Salary :" + salary);

System.out.println("Employee Address :" + address);

System.out.println("Teacher Department:" + department);

System.out.println("Teacher Subject :" + subject);

}

}

public class EmpCO3 {

public static void main(String[] args) {

int i;

Scanner cin = new Scanner(System.in);

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

int n = cin.nextInt();

Teacher e[] = new Teacher[n];

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

System.out.print("Enter the Id of the Employee :");

int a = cin.nextInt();

System.out.print("Enter the Name of the Employee :");

String b = cin.next();

System.out.print("Enter the Salary of the Employee :");

int c = cin.nextInt();

System.out.print("Enter the Address of the Employee :");

String d = cin.next();

System.out.print("Enter the Department of the Teachers :");

String q = cin.next();

System.out.print("Enter the Subject of the Teachers :");

String f = cin.next();

e[i] = new Teacher(a, b, c, d, q, f);

}

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

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

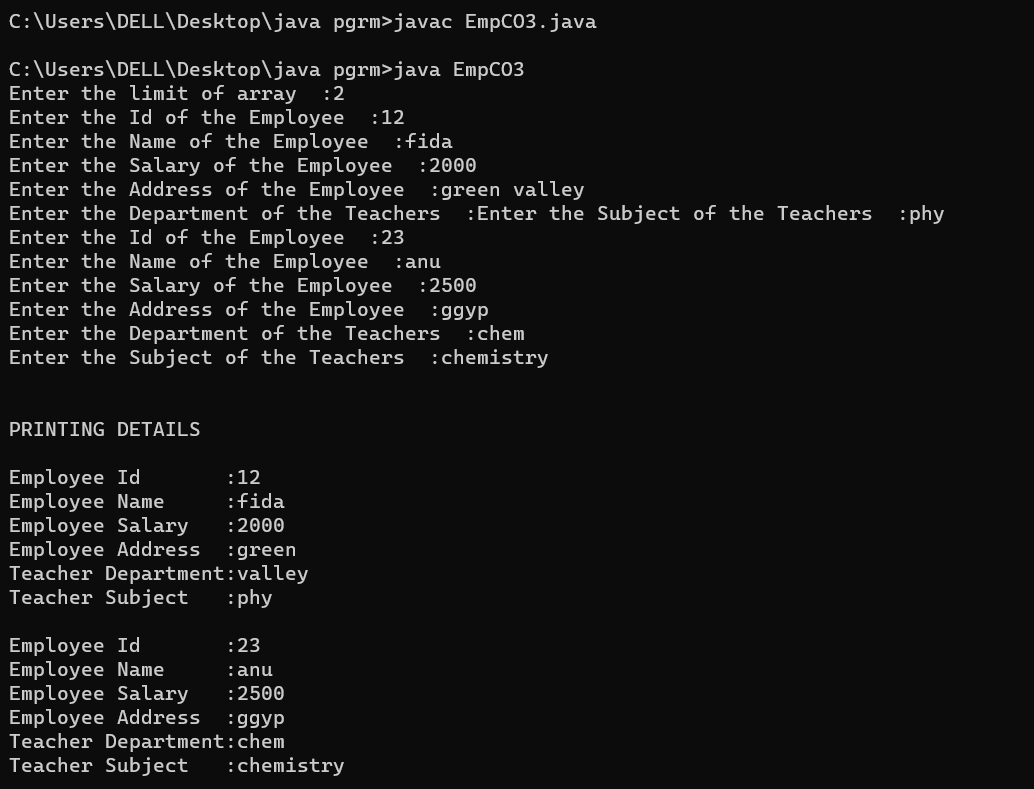
e[i].display();

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 12**

**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**



**Result**

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

**Experiment No.: 13**

**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 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: "+price);

System.out.println("Category:Fiction");

}

}

class literature extends book

{

void literature()

{

System.out.println("Publisher Name:"+nam);

System.out.println("Book Name:"+tit);

System.out.println("Author Name: "+auth);

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

System.out.println("Category:Literature");

}

}

class co3\_4\_bookdetails

{

public static void main(String[] args)

{

int i;

Scanner a=new Scanner(System.in);

System.out.println(" ");

System.out.print("Enter no of entries:");

int n=a.nextInt();

fiction dd[]=new fiction[n];

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

{

System.out.println(" ");

System.out.println("Details of book "+(i+1));

dd[i]=new fiction();

dd[i].publisher();

dd[i].book();

System.out.println(" ");

}

System.out.println(" ");

System.out.print("Enter no of literature books:");

int n1=a.nextInt();

literature ll[]=new literature[n];

for(i=0;i<n1;i++)

{

System.out.println(" ");

System.out.println("Details of book "+(i+1));

ll[i]=new literature();

ll[i].publisher();

ll[i].book();

}

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

{

System.out.println(" ");

System.out.println("Details of fiction book "+(i+1));

dd[i].fiction();

}

for(i=0;i<n1;i++)

{

System.out.println(" ");

System.out.println("Details of fiction book "+(i+1));

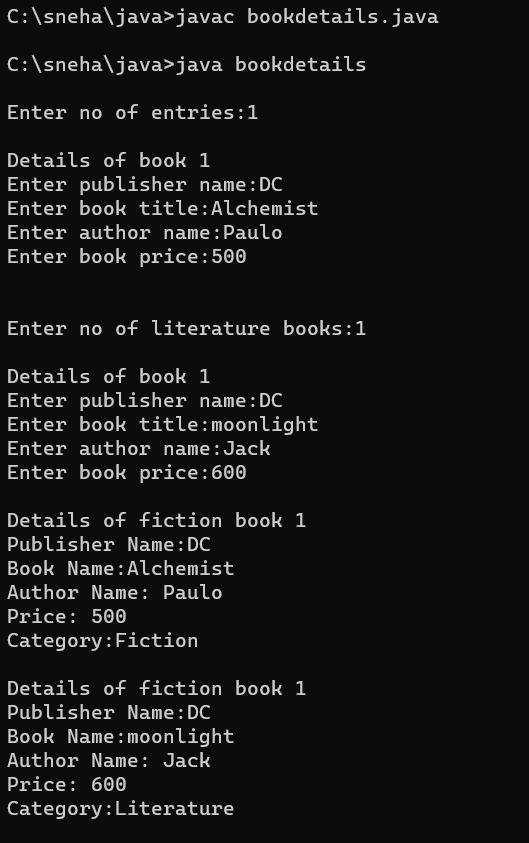
ll[i].literature();

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 14**

**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.Scanner;

interface Student {

void getAcademicMarks();

}

interface Sports {

void getSportsMarks();

}

class Result implements Student, Sports {

private int academicMarks;

private int sportsMarks;

public void getAcademicMarks() {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the marks obtained for the examination: ");

academicMarks = sc.nextInt();

}

public void getSportsMarks() {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the marks/position obtained for Sports: ");

sportsMarks = sc.nextInt();

}

public void displayScores() {

System.out.println("\nAcademic Score: " + academicMarks);

System.out.println("Sports Score: " + sportsMarks);

}

}

public class stuCO3 {

public static void main(String[] args) {

Result res = new Result();

res.getAcademicMarks();

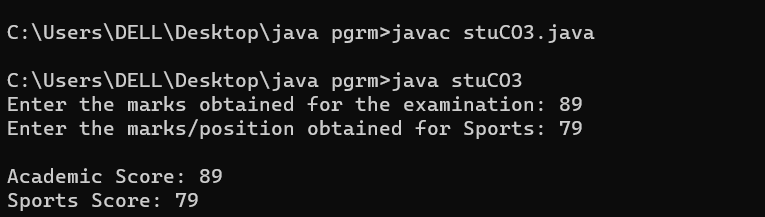
res.getSportsMarks();

res.displayScores();

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 15**

**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;

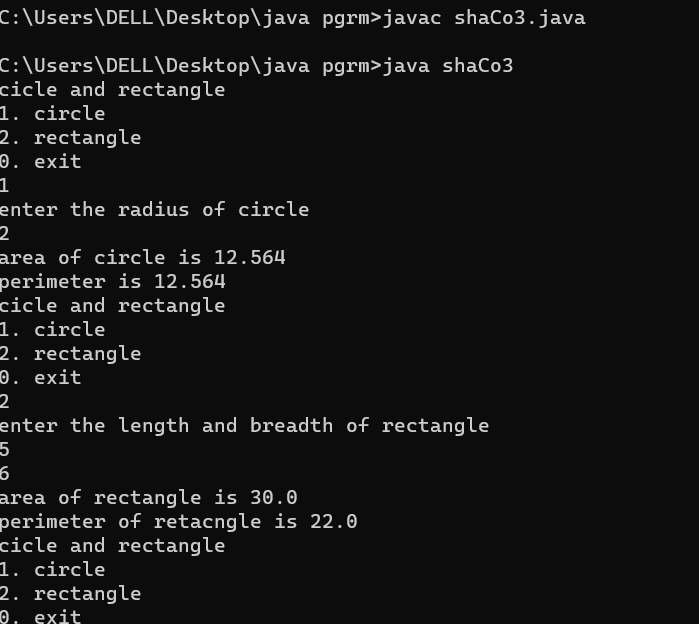
}

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 16**

**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 bill

{

public void get();

}

interface calculate

{

public void cal();

}

class co3\_7\_bill implements bill,calculate

{

String name;

int n,id,qy,price,total;

public void get()

{

Scanner a=new Scanner(System.in);

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

id=a.nextInt();

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

name=a.next();

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

qy=a.nextInt();

System.out.println("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;

float net=0;

Scanner a=new Scanner(System.in);

System.out.print("Enter order no:");

ord=a.nextInt();

System.out.println("Enter date:");

date=a.next();

System.out.println("Enter the number of ordered products:");

n=a.nextInt();

co3\_7\_bill odd[]=new co3\_7\_bill[n];

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

{

odd[i]=new co3\_7\_bill();

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 \tQuantity Unit price \tTotal");

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

{

odd[i].display();

net=net+odd[i].total;

}

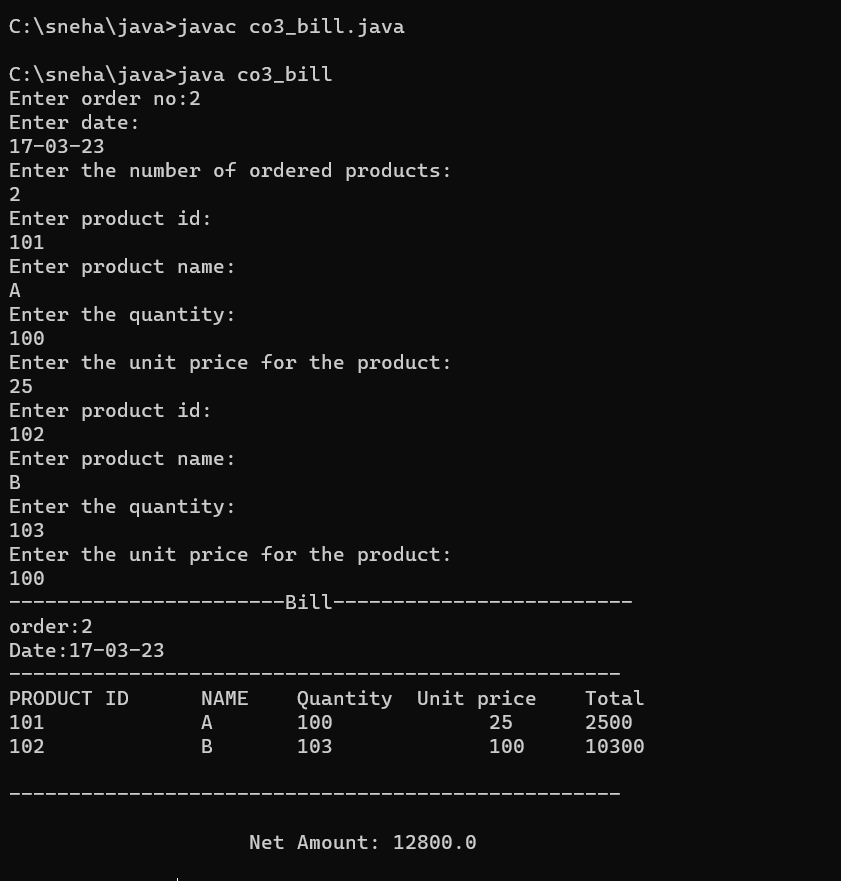
System.out.println("\n---------------------------------------------------");

System.out.println("\n Net Amount: "+net);

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 17**

**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**

//areaGraphics.java

package graphics;

import java.util.\*;

interface shapes

{

public double area\_of\_rect();

public double area\_of\_triangle();

public double area\_of\_square();

public double area\_of\_circle();

}

public class areaGraphics implements shapes

{

Scanner sc=new Scanner(System.in);

int l,b,ba,h,s,r;

double area;

public double area\_of\_rect()

{

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

l=sc.nextInt();

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

b=sc.nextInt();

area=l\*b;

return area;

}

public double area\_of\_triangle()

{

System.out.println("Enter the base of triangle:");

int ba=sc.nextInt();

System.out.println("Enter the height of triangle:");

int h=sc.nextInt();

area=0.5\*ba\*h;

return area;

}

public double area\_of\_square()

{

System.out.println("Enter the side of square:");

int s=sc.nextInt();

area=s\*s;

return area;

}

public double area\_of\_circle()

{

System.out.println("Enter the radius of circle:");

int r=sc.nextInt();

area=3.14\*r\*r;

return area;

}

}

// co41\_area\_of\_shapes.java

import graphics.areaGraphics;

import java.util.\*;

public class co41\_area\_of\_shapes

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

areaGraphics obj=new areaGraphics();

int ch=0;

while(ch !=5)

{

System.out.println("\*\*\*Area of shapes\*\*\*");

System.out.println("1.Area of rectangle");

System.out.println("2.Area of triangle");

System.out.println("3.Area of square");

System.out.println("4.Area of circle");

System.out.println("5.Exit");

System.out.println(" ");

System.out.println("Enter your choice:");

ch=sc.nextInt();

switch(ch)

{

case 1:

System.out.println("Area of Rectangle: " + obj.area\_of\_rect() );

break;

case 2:

System.out.println("Area of Triangle: " + obj.area\_of\_triangle() );

break;

case 3:

System.out.println("Area of Square: " + obj.area\_of\_square() );

break;

case 4:

System.out.println("Area of Circle: " + obj.area\_of\_circle() );

break;

case 5:

System.exit(0);

break;

default:

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

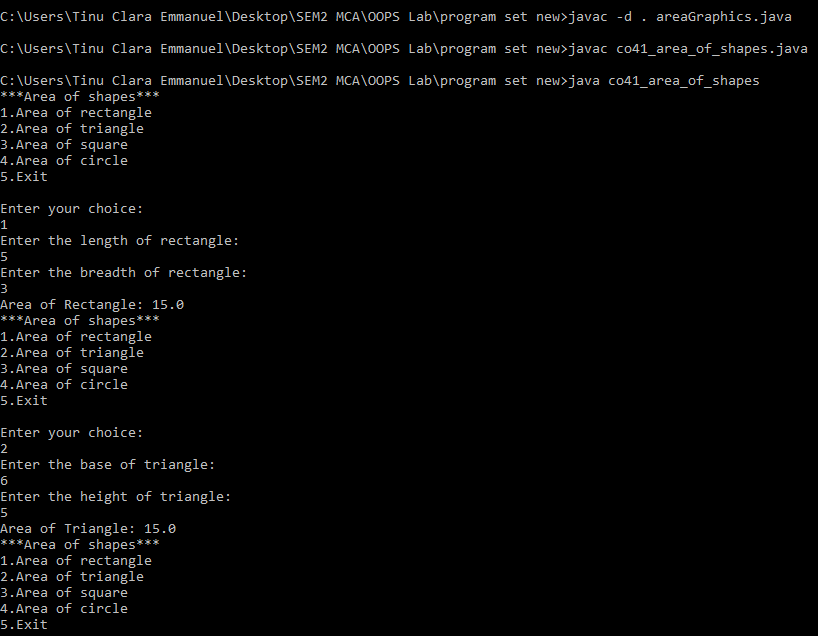
}

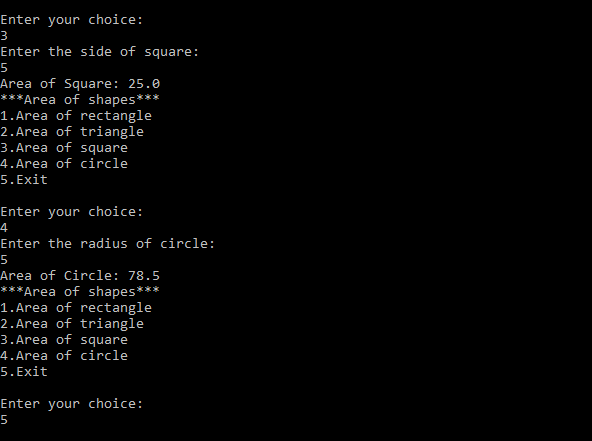
}

}

}

**Output Screenshot**





**Result**

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

**Experiment No.: 18**

**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 graphics;

import java.util.\*;

interface operations

{

public void values();

public double add();

public double subtract();

public double multiply();

public double division();

}

public class arithmetic implements operations

{

Scanner sc=new Scanner(System.in);

int n,m;

double result;

public void values()

{

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

n=sc.nextInt();

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

m=sc.nextInt();

}

public double add()

{

result=n+m;

return result;

}

public double subtract()

{

result=n-m;

return result;

}

public double multiply()

{

result=n\*m;

return result;

}

public double division()

{

result=n/m;

return result;

}

}

// co42\_operations.java

import graphics.arithmetic;

import java.util.\*;

public class co42\_operations

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

arithmetic obj=new arithmetic();

int ch=0;

while(ch !=5)

{

System.out.println("\*\*\*ARITHMETIC OPERATIONS\*\*\*");

System.out.println("1.ADDITION");

System.out.println("2.SUBTRACTION");

System.out.println("3.MULTIPLICATION");

System.out.println("4.DIVISION");

System.out.println("5.Exit");

System.out.println(" ");

System.out.println("Enter your choice:");

ch=sc.nextInt();

switch(ch)

{

case 1:

obj.values();

System.out.println("RESULT OF ADDITION: " + obj.add());

break;

case 2:

obj.values();

System.out.println("RESULT OF SUBTRACTION: " + obj.subtract());

break;

case 3:

obj.values();

System.out.println("RESULT OF MULTIPLICATION: " + obj.multiply());

break;

case 4:

obj.values();

System.out.println("RESULT OF DIVISION: " + obj.division());

break;

case 5:

System.exit(0);

break;

default:

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

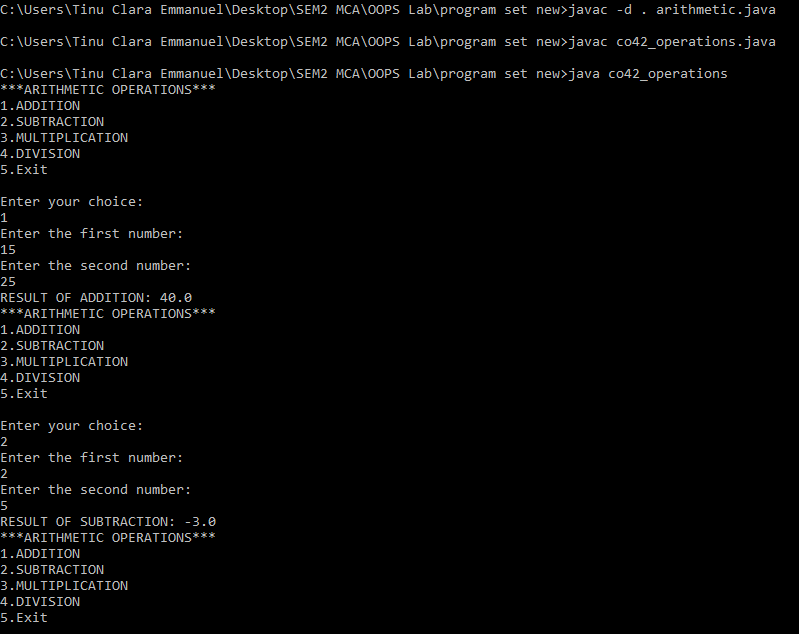
}

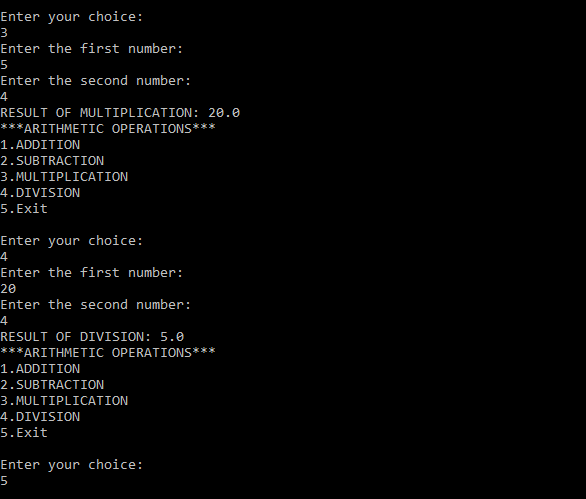
}

}

}

**Output Screenshot**





**Result**

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

**Experiment No.: 19**

**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 user\_excep extends Exception

{

public user\_excep(String str)

{

super(str);

}

}

class pass\_excep extends Exception

{

public pass\_excep(String str)

{

super(str);

}

}

public class validate

{

public static void main(String[] args)

{

Scanner Snr = new Scanner(System.in);

String user, pass;

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

user = Snr.next();

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

pass = Snr.next();

int uslen = user.length();

try

{

if(uslen < 8)

{

throw new user\_excep("User Name should be more than 8 characters");

}

if(!pass.equals("root"))

{

throw new pass\_excep("Incorrect Password");

}

else

{

System.out.println(":: LOGIN SUCCESSFUL ::");

}

}

catch (user\_excep u)

{

u.printStackTrace();

}

catch (pass\_excep p)

{

p.printStackTrace();

}

}

}

**Output Screenshot**

**Result**

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

**Experiment No.: 20**

**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 co44\_exception{

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){

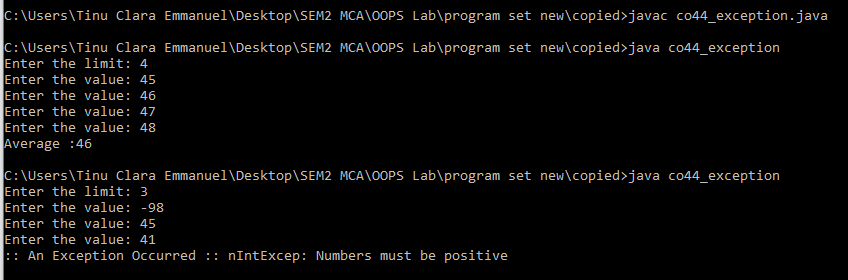
System.out.println(":: An Exception Occurred :: "+ e);

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 21**

**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 MultiT extends Thread{

public void run(){

Scanner Snr = new Scanner(System.in);

System.out.println(":: MULTIPLICATION TABLE ::");

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

int num = Snr.nextInt();

System.out.println("Multiplication table of " + num + " : ");

for(int i=1;i<=10;i++){

System.out.println(num+ " \* " + i + " = " + num\*i);

}

}

}

class PrimeNo extends Thread

{

public void run()

{

Scanner Snr = new Scanner(System.in);

System.out.println(":: PRIME NUMBERS ::");

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

int sz = Snr.nextInt();

System.out.println("Required Prime Numbers: ");

for(int i=1;i<=sz;i++){

if(i==1 || i==0){

continue;

}

else{

int flag= 1;

for(int j=2;j<=i/2;j++)

{

if(i%j == 0)

{

flag=0;

break;

}

}

if(flag == 1)

{

System.out.println(i);

}

}

}

}

}

public class co45\_multiplication{

public static void main(String[] args) throws InterruptedException

{

MultiT m = new MultiT();

m.start();

m.sleep(2500);

PrimeNo p = new PrimeNo();

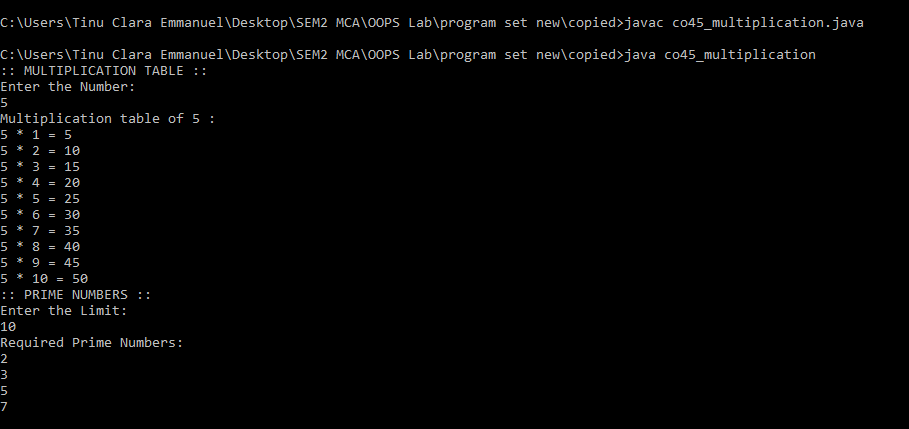
p.start();

p.sleep(2500);

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 22**

**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.\*;

class Fibonacci implements Runnable{

public void run(){

Scanner Snr = new Scanner(System.in);

int n1 = 0, n2 = 1, n3;

System.out.println(":: FIBONACCI NUMBERS ::");

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

int sz = Snr.nextInt();

System.out.println("The Required Fibonacci Number(s): ");

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

System.out.print(n1+ " ");

n3=n1+n2;

n1=n2;

n2=n3;

}

}

}

class EvenNo implements Runnable{

public void run(){

int uplmt, lwlmt;

Scanner Snr = new Scanner(System.in);

System.out.println("\n:: EVEN NUMBERS ::");

System.out.println("Enter the Lower Limit: ");

lwlmt = Snr.nextInt();

System.out.println("Enter the Upper Limit: ");

uplmt = Snr.nextInt();

System.out.println("The Required Even Number(s): ");

for(int i=lwlmt; i<=uplmt; i++){

if(i%2 == 0){

System.out.print(i + " ");

}

}

}

}

public class co46\_finonacci{

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

Fibonacci obj1 = new Fibonacci();

Thread a = new Thread(obj1);

a.start();

a.sleep(2500);

EvenNo obj2 = new EvenNo();

Thread b = new Thread(obj2);

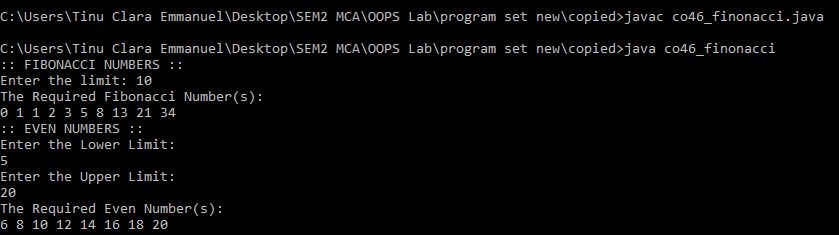
b.start();

b.sleep(2500);

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 23**

**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 co48\_generic\_stack

{

public static void main( String args[])

{

int pos,ch=0;

Scanner obj =new Scanner(System.in);

Stack<String> sta =new Stack<String>();

while(ch !=4)

{

System.out.println("1)ADD ");

System.out.println("2)REMOVE ");

System.out.println("3)DISPLAY ");

System.out.println("4)EXIT");

System.out.println("Select your Option ");

ch=obj.nextInt();

switch(ch)

{

case 1:

System.out.println(" Enter the Element to be inserted :- ");

sta.add(obj.next()) ;

System.out.println(sta);

break;

case 2:

System.out.println(" Enter the position of element removed from stack");

pos=obj.nextInt();

sta.remove(pos-1);

System.out.println(sta);

break;

case 3:

System.out.println(sta);

break;

case 4:

System.exit(0);

break;

default:

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

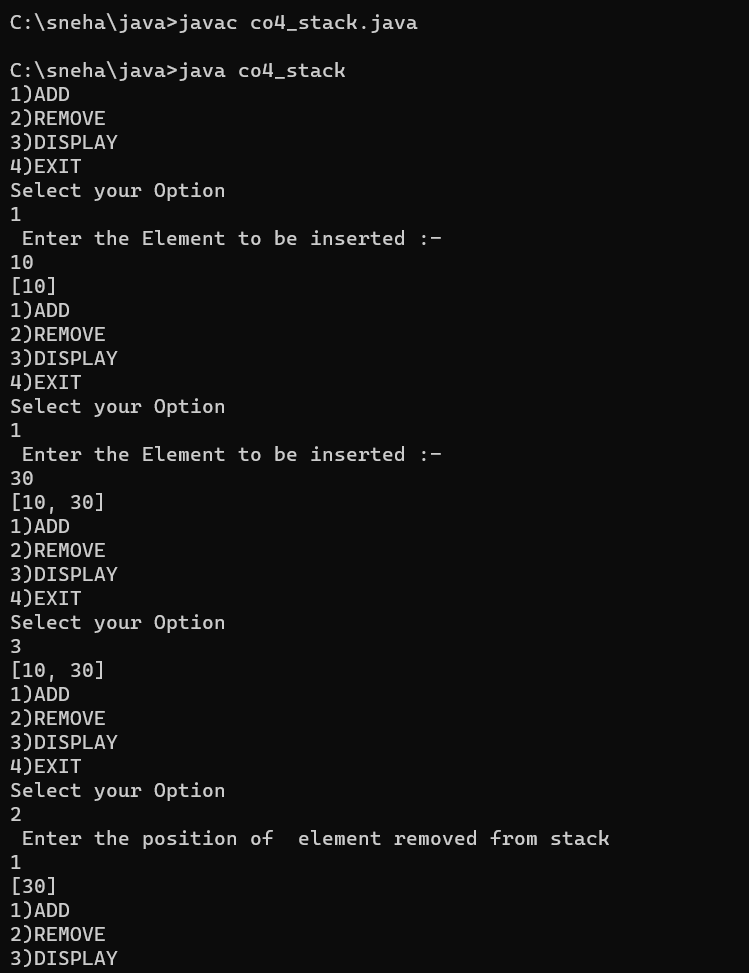
}

}

}

}

**Output Screenshot**





**Result**

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

**Experiment No.: 24**

**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.Scanner;

class bubble

{

int n=4;

int ar[]=new int[n];

Scanner cin=new Scanner(System.in);

void insert()

{

int i;

System.out.println("Enter the elements of array");

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

{

ar[i]=cin.nextInt();

}

}

void sort()

{

int i,j,temp;

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

{

for(j=i+1;j<n;j++)

{

if(ar[j] < ar[i])

{

temp = ar[i];

ar[i] = ar[j];

ar[j] = temp;

}

}

}

}

void display()

{

int i;

System.out.println("printing sorted array");

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

{

System.out.println(ar[i]);

}

}

}

public class co49\_bubblesort

{

public static void main(String args[])

{

bubble b=new bubble();

b.insert();

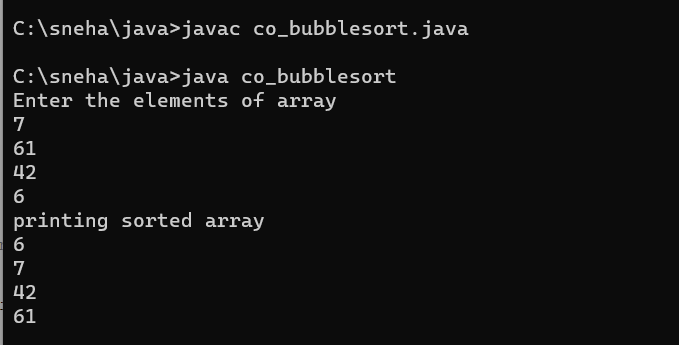
b.sort();

b.display();

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 25**

**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.\*;

public class arraylist{

public static void main(String []args)

{

ArrayList<String> arr=new ArrayList<>();

arr.add("alpha");

arr.add("beta");

arr.add("gamma");

System.out.println("the element of the arraylist is -"+arr);

Collections.sort(arr);

System.out.println("the sorted array"+arr);

Collections.addAll(arr,"beta","meta","meta");

System.out.println("add new items in the list"+arr);

Collections.sort(arr,Collections.reverseOrder());

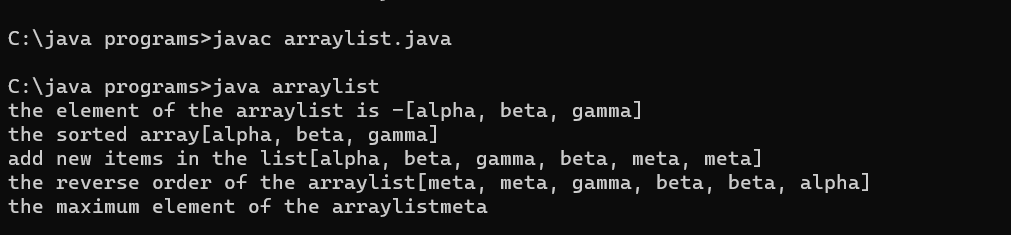
System.out.println("the reverse order of the arraylist"+arr);

System.out.println("the maximum element of the arraylist"+ Collections.max(arr));

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 26**

**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.\*;

import java.util.LinkedList;

public class linkedlist{

public static void insertBeg(LinkedList<String> list){

Scanner Snr = new Scanner(System.in);

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

String line = Snr.next();

list.addFirst(line);

}

public static void insertEnd(LinkedList<String> list){

Scanner Snr = new Scanner(System.in);

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

String line = Snr.next();

list.addLast(line);

}

public static void delBeg(LinkedList<String> list){

System.out.println(list.removeFirst() + " removed");

}

public static void delEnd(LinkedList<String> list){

System.out.println(list.removeLast() + " removed");

}

public static void gtfr(LinkedList<String> list){

System.out.println(list.getFirst());

}

public static void gtend(LinkedList<String> list){

System.out.println(list.getLast());

}

public static void delAll(LinkedList<String> list){

list.clear();

}

public static void main(String[] args){

Scanner Snr =new Scanner(System.in);

LinkedList<String> list = new LinkedList<String>();

int choice = 1;

while(choice != 9){

System.out.println("");

System.out.println(":: LINKED LIST ::");

System.out.println("1. Insert at Beginning");

System.out.println("2. Insert at End");

System.out.println("3. Delete from Beginning");

System.out.println("4. Delete from End");

System.out.println("5. Get the First Element");

System.out.println("6. Get the Last Element");

System.out.println("7. Delete all elements");

System.out.println("8. Display");

System.out.println("9. Exit");

System.out.print("Enter your choice: ");

choice = Snr.nextInt();

switch(choice){

case 1:

insertBeg(list);

break;

case 2:

insertEnd(list);

break;

case 3:

delBeg(list);

break;

case 4:

delEnd(list);

break;

case 5:

gtfr(list);

break;

case 6:

gtend(list);

break;

case 7:

delAll(list);

break;

case 8:

System.out.println(list);

break;

case 9:

System.exit(0);

break;

default:

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

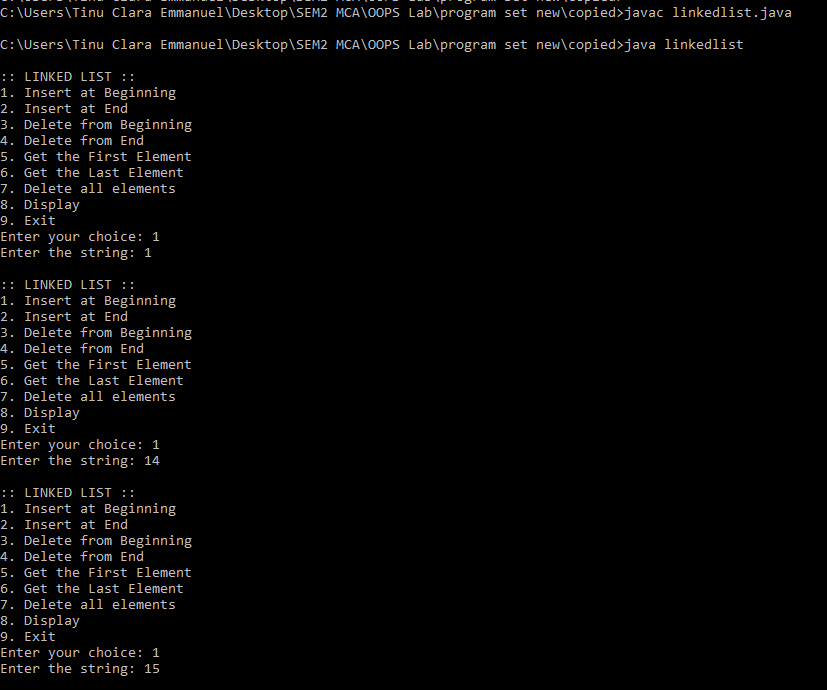
}

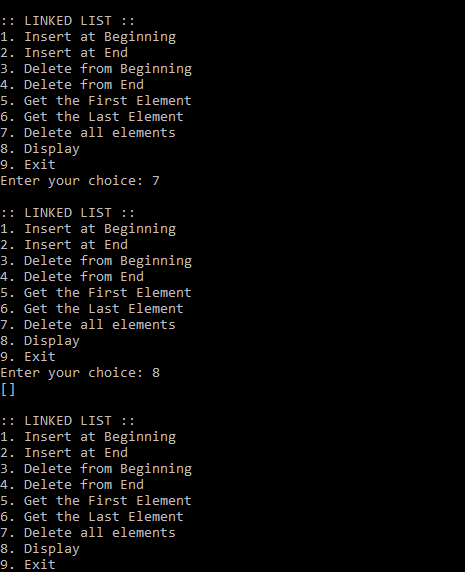
}

}

}

**Output Screenshot**





**Result**

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

**Experiment No.: 27**

**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**

**Aim:**

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

**Course Outcome(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 stack{

public static void insert(Stack<String> stk){

Scanner Snr = new Scanner(System.in);

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

String line = Snr.next();

stk.add(line);

}

public static void delete(Stack<String> stk){

Scanner Snr = new Scanner(System.in);

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

int pos = Snr.nextInt();

pos = pos-1;

System.out.println(stk.remove(pos) + " removed");;

}

public static void display(Stack<String> stk){

System.out.println("Stack: " + stk);

}

public static void main(String[] args){

Scanner Snr =new Scanner(System.in);

Stack<String> stk = new Stack<String>();

int choice = 0;

while(choice != 4){

System.out.println(" ");

System.out.println(":: STACK OPERATIONS ::");

System.out.println("1. Insert");

System.out.println("2. Delete");

System.out.println("3. Display");

System.out.println("4. Exit");

System.out.print("Enter your choice: ");

choice = Snr.nextInt();

switch(choice){

case 1:

insert(stk);

break;

case 2:

delete(stk);

break;

case 3:

display(stk);

break;

case 4:

System.exit(0);

break;

default:

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

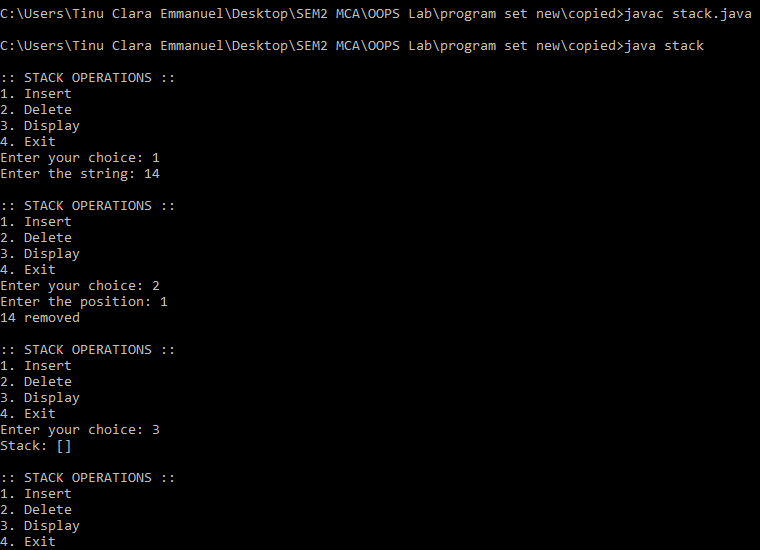
}

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 28**

**Aim**

Program to demonstrate the 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.\*;

import java.util.PriorityQueue;

public class priorqueue{

public static void insert(PriorityQueue<Integer> pq){

Scanner Snr = new Scanner(System.in);

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

int num = Snr.nextInt();

pq.add(num);

}

public static void last(PriorityQueue<Integer> pq){

System.out.println(pq.peek());

}

public static void del(PriorityQueue<Integer> pq){

System.out.println(pq.poll() + " removed");

}

public static void display(PriorityQueue<Integer> pq){

System.out.println("Priority Queue: " + pq);

}

public static void main(String[] args){

Scanner Snr =new Scanner(System.in);

PriorityQueue<Integer> pq = new PriorityQueue<Integer>();

int choice = 0;

while(choice != 5){

System.out.println(" ");

System.out.println(":: PRIORITY QUEUE OPERATIONS ::");

System.out.println("1. Insert");

System.out.println("2. Print top element");

System.out.println("3. Print and Delete top element");

System.out.println("4. Display");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

choice = Snr.nextInt();

switch(choice){

case 1:

insert(pq);

break;

case 2:

last(pq);

break;

case 3:

del(pq);

break;

case 4:

display(pq);

break;

case 5:

System.exit(0);

break;

default:

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

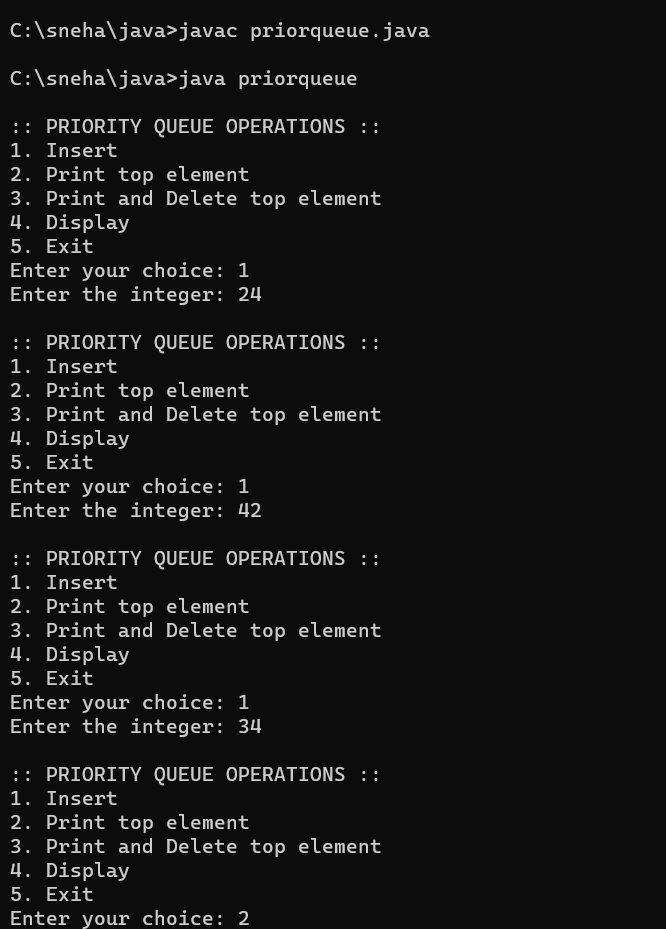
}

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 29**

**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.ArrayDeque;

import java.util.Deque;

public class deque{

public static void insertBeg(Deque<Integer> dq){

Scanner Snr = new Scanner(System.in);

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

int num = Snr.nextInt();

dq.addFirst(num);

}

public static void insertEnd(Deque<Integer> dq){

Scanner Snr = new Scanner(System.in);

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

int num = Snr.nextInt();

dq.addLast(num);

}

public static void delBeg(Deque<Integer> dq){

System.out.println(dq.removeFirst() + " removed");

}

public static void delEnd(Deque<Integer> dq){

System.out.println(dq.removeLast() + " removed");

}

public static void main(String[] args){

Scanner Snr =new Scanner(System.in);

Deque<Integer> dq = new ArrayDeque<Integer>();

int choice = 0;

while(choice != 6){

System.out.println(" ");

System.out.println(":: DEQUE OPERATIONS ::");

System.out.println("1. Insert at Beginning");

System.out.println("2. Insert at End");

System.out.println("3. Delete from Beginning");

System.out.println("4. Delete from End");

System.out.println("5. Display");

System.out.println("6. Exit");

System.out.print("Enter your choice: ");

choice = Snr.nextInt();

switch(choice){

case 1:

insertBeg(dq);

break;

case 2:

insertEnd(dq);

break;

case 3:

delBeg(dq);

break;

case 4:

delEnd(dq);

break;

case 5:

System.out.println(dq);

break;

case 6:

System.exit(0);

break;

default:

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

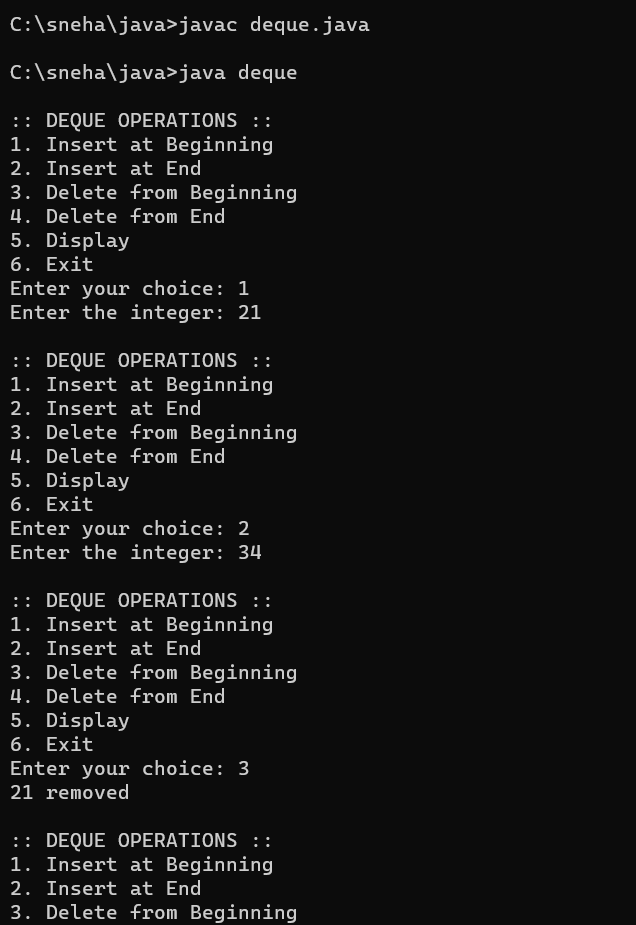
}

}

}

}

**Output Screenshot**



**Result**

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

**Experiment No.: 30**

**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.\*;

import java.util.LinkedHashSet;

import java.util.Set;

public class set{

public static void insert(Set<Integer> st){

Scanner Snr = new Scanner(System.in);

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

int line = Snr.nextInt();

st.add(line);

}

public static void del(Set<Integer> st){

Scanner Snr = new Scanner(System.in);

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

int pos = Snr.nextInt();

pos = pos-1;

System.out.println(st.remove(pos) + " removed");;

}

public static void display(Set<Integer> st){

System.out.println("Stack: " + st);

}

public static void delAll(Set<Integer> st){

st.clear();

System.out.println("Set successfully cleared");

}

public static void main(String[] args){

Set<Integer> st = new LinkedHashSet<Integer>();

Scanner Snr = new Scanner(System.in);

int choice1 = 0, choice2 = 0, choice3 = 0;

while(choice1 != 5){

System.out.println(":: SET OPERATIONS ::");

System.out.println("1. Insert");

System.out.println("2. Delete");

System.out.println("3. Display");

System.out.println("4. Clear All");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

choice1 = Snr.nextInt();

switch(choice1){

case 1:

insert(st);

break;

case 2:

del(st);

break;

case 3:

display(st);

break;

case 4:

delAll(st);

break;

case 5:

System.exit(0);

break;

default:

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

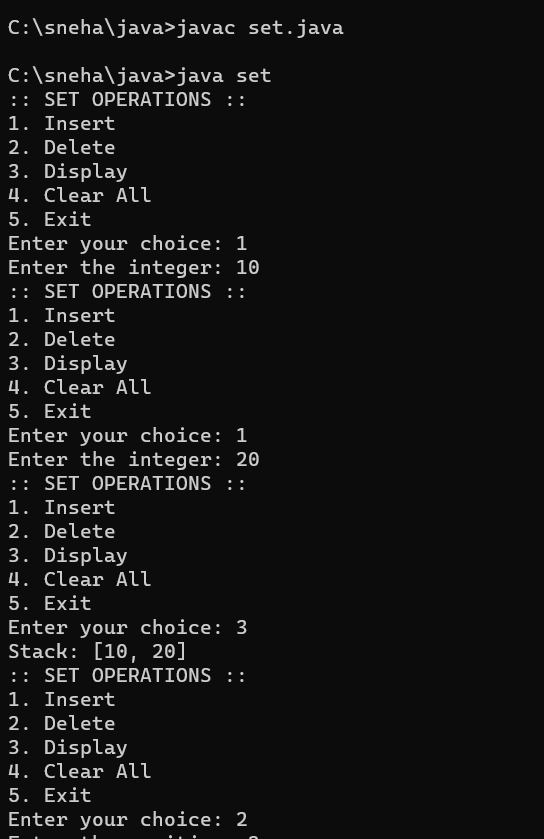
}

}

}

}

**Output Screenshot**



**Result**

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

**Course Outcome 5 (CO5):**

**1. Program to draw Circle, Rectangle, Line in Applet.**

**applet.java**

import java.applet.\*;

import java.awt.\*;

public class applet extends Applet{

public void paint(Graphics g){

g.drawLine(20,20,200,20);

g.drawRect(20,100,200,40);

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

}

}

**applet.html**

<html>

<head></head>

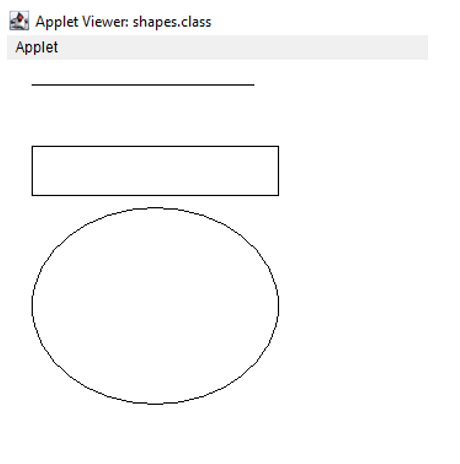
<title>APPLET</title>

<body>

<applet code="applet.class" height="300" width="300"></applet>

</body>

</html>



**2. Program to find maximum of three numbers using AWT.**

import java.awt.\*;

import java.awt.Graphics;

import java.applet.\*;

import java.awt.event.\*;

public class maxThree extends Applet implements ActionListener

{

Label l1,l2,l3,l4;

TextField tf1,tf2,tf3,tf4;

Button btn;

public void init()

{

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

tf1=new TextField();

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

tf2=new TextField();

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

tf3=new TextField();

l4 = new Label("MAXIMUM:");

tf4= new TextField();

btn= new Button("Submit");

setLayout(null);

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

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

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

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

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

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

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

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

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

add(l1);

add(l2);

add(l3);

add(l4);

add(tf1);

add(tf2);

add(tf3);

add(tf4);

add(btn);

btn.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==btn)

{

float n1,n2,n3,max;

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

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

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

if(n1>n2 && n1>n3)

{

max=n1;

}

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

{

max=n2;

}

else

{

max=n3;

}

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

}

}

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

{

maximum m = new maximum();

m.setTitle("Maximum");

m.setSize(150,150);

m.setVisible(true);

}\*/

}

<html>

<head>

<title>Maximum of Three</title>

</head>

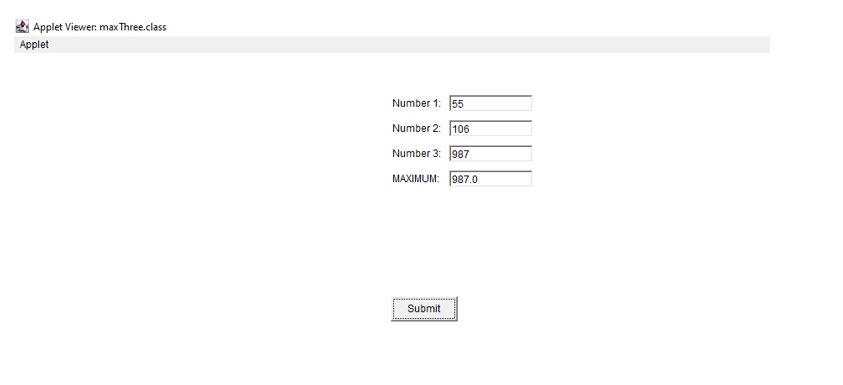
<body>

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

</applet>

</body>

</html>



**3. 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.**

import java.applet.\*;

import java.awt.\*;

import java.awt.Graphics;

import java.awt.event.\*;

public class Main extends Applet implements ActionListener {

Label l1,l2,l3,l4,l5,l6;

TextField t1,t2,t3,t4,t5,t6;

Button b;

public void init(){

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

t1 = new TextField();

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

t2 = new TextField();

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

t3 = new TextField();

l4 = new Label("MARK 4:");

t4 = new TextField();

l5 = new Label("MARK 5:");

t5 = new TextField();

l6 = new Label("PERCENTAGE:");

t6 = new TextField();

b = new Button("SEE STATUS");

setLayout(null);

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

t1.setBounds(520,50,100,20);

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

t2.setBounds(520,80,100,20);

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

t3.setBounds(520,110,100,20);

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

t4.setBounds(520,140,100,20);

l5.setBounds(450,170,70,20);

t5.setBounds(520,170,100,20);

l6.setBounds(450,200,100,20);

t6.setBounds(550,200,100,20);

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

add(l1);

add(l2);

add(l3);

add(l4);

add(l5);

add(l6);

add(t1);

add(t2);

add(t3);

add(t4);

add(t5);

add(t6);

add(b);

b.addActionListener(this);

}

public void actionPerformed(ActionEvent e){

float m1, m2,m3, m4,m5,percent;

m1= Float.parseFloat(t1.getText());

m2= Float.parseFloat(t2.getText());

m3= Float.parseFloat(t3.getText());

m4= Float.parseFloat(t4.getText());

m5= Float.parseFloat(t5.getText());

percent=((m1+m2+m3+m4+m5)\*100)/500;

t6.setText(String.valueOf(percent));

repaint();

}

public void paint(Graphics g){

float p;

p= Float.parseFloat(t6.getText());

if(p> 50.0) {

g.setColor(Color.YELLOW);

g.fillOval(0,0,100,100);

g.setColor(Color.black);

g.fillOval(25,25,10,10);

g.fillOval(65,25,10,10);

g.setColor(Color.black);

g.fillArc (25,35,50,50,0,-180);

}

else {

g.setColor(Color.YELLOW);

g.fillOval(0,0,100,100);

g.setColor(Color.black);

g.fillOval(25,25,10,10);

g.fillOval(75,25,10,10);

g.setColor(Color.black);

g.drawArc(25,35,50,50,0,180);

}

}

}

**#main.html**

<html>

<head></head>

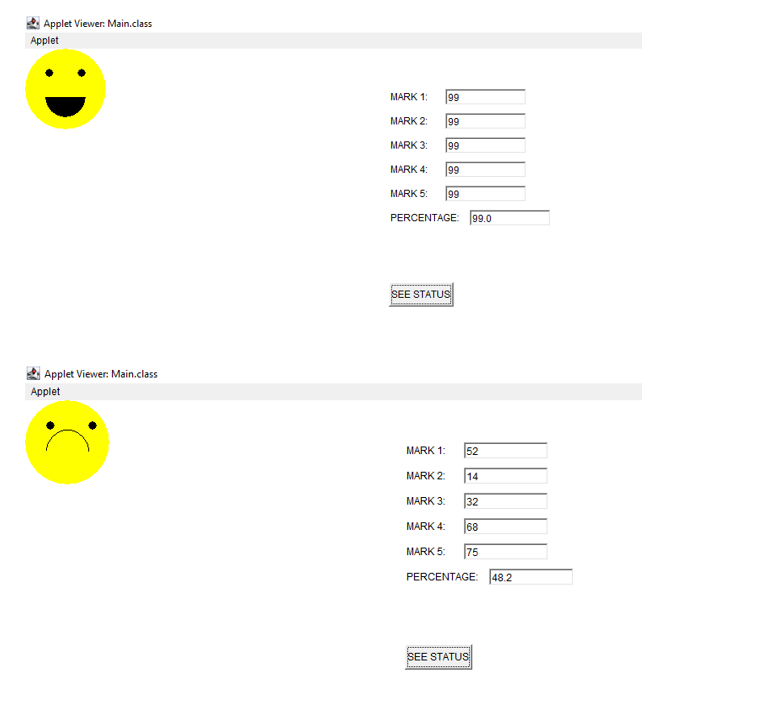
<title>APPLET</title>

<body>

<applet code="Main.class" height="300" width="300"></applet>

</body>

</html>



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

**HouseApplet.java**

import java.awt.\*;

import java.awt.event.\*;

import java.applet;

public class HouseApplet extends Applet {

private Color doorColor;

public void init() {

// Set the initial color of the door

doorColor = Color.BLUE;

// Add mouse click listener

addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

// Change the color of the door to red on mouse click

doorColor = Color.RED;

repaint();

}

});

}

public void paint(Graphics g) {

// Set the background color

setBackground(Color.WHITE);

// Draw the house

g.setColor(Color.GRAY);

g.fillRect(100, 100, 200, 200); // House body

g.setColor(Color.RED);

g.fillRect(155, 200, 90, 100); // Door

g.setColor(Color.BLACK);

g.drawRect(155, 200, 90, 100); // Door outline

g.setColor(doorColor); // Set the color of the door

g.fillOval(180, 250, 10, 10); // Doorknob

g.setColor(Color.LIGHT\_GRAY);

g.fillRect(120, 150, 60, 60); // Window 1

g.fillRect(220, 150, 60, 60); // Window 2

g.setColor(Color.BLACK);

g.drawRect(120, 150, 60, 60); // Window 1 outline

g.drawRect(220, 150, 60, 60); // Window 2 outline

g.drawLine(190, 100, 100, 200); // Roof left line

g.drawLine(190, 100, 280, 200); // Roof right line

}

}

**Index.html**

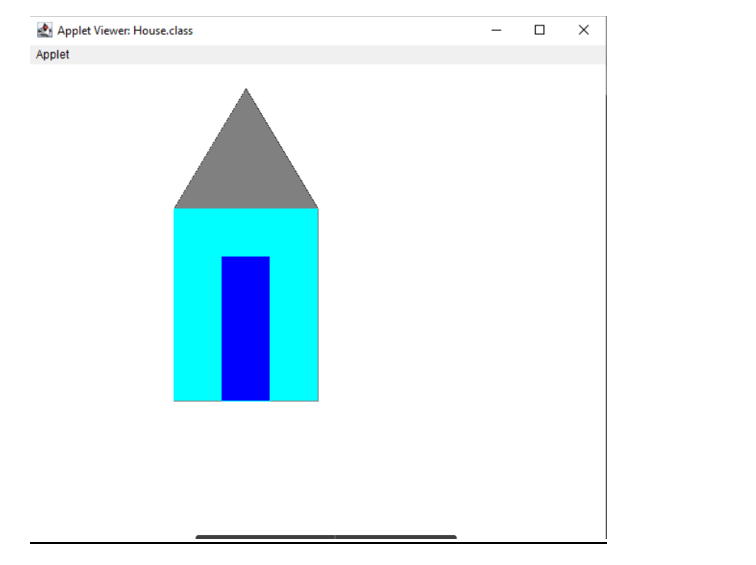
<html>

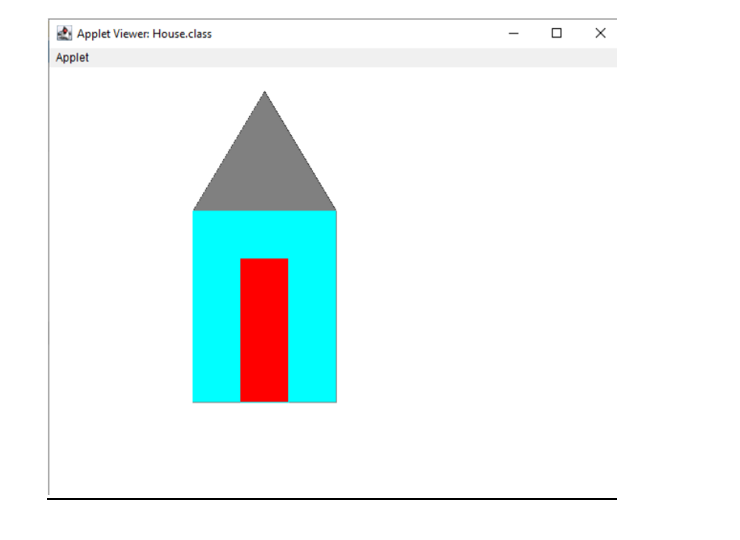
<body>

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

</body></html>

**OUTPUT**

****

****

**5. Implement a simple calculator using AWT components.**

import java.awt.\*;

import java.awt.event.\*;

class Cal extends Frame implements ActionListener{

TextField inp;

Panel p;

String bs[] ={ "1","2","3","+",

"4","5","6","-",

"7","8","9","\*",

"C","0","/","=",

};

Button b[] = new Button[16];

int n1, n2, result;

String opr;

public Cal(){

inp = new TextField(20);

p = new Panel();

add(inp, "North");

add(p, "Center");

p.setLayout(new GridLayout(4,4));

for(int i=0; i<16;i++){

b[i] = new Button(bs[i]);

b[i].addActionListener(this);

p.add(b[i]);

}

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent we){

System.exit(0);

}

});

}

public void actionPerformed(ActionEvent ae){

String str = ae.getActionCommand();

if(str.equals("+")){

opr = "+";

n1 = Integer.parseInt(inp.getText());

inp.setText("");

}

else if(str.equals("-")){

opr = "-";

n1 = Integer.parseInt(inp.getText());

inp.setText("");

}

else if(str.equals("\*")){

opr = "\*";

n1 = Integer.parseInt(inp.getText());

inp.setText("");

}

else if(str.equals("/")){

opr = "/";

n1 = Integer.parseInt(inp.getText());

inp.setText("");

}

else if(str.equals("=")){

n2 = Integer.parseInt(inp.getText());

switch(opr){

case "+":

result = n1 + n2;

break;

case "-":

result = n1 - n2;

break;

case "\*":

result = n1 \* n2;

break;

case "/":

result = n1 / n2;

break;

}

inp.setText(String.valueOf(result));

}

else if(str.equals("C")){

inp.setText("");

n1=n2=result=0;

}

else{

inp.setText(inp.getText() + str);

}

}

public static void main(String[] args) {

Cal c = new Cal();

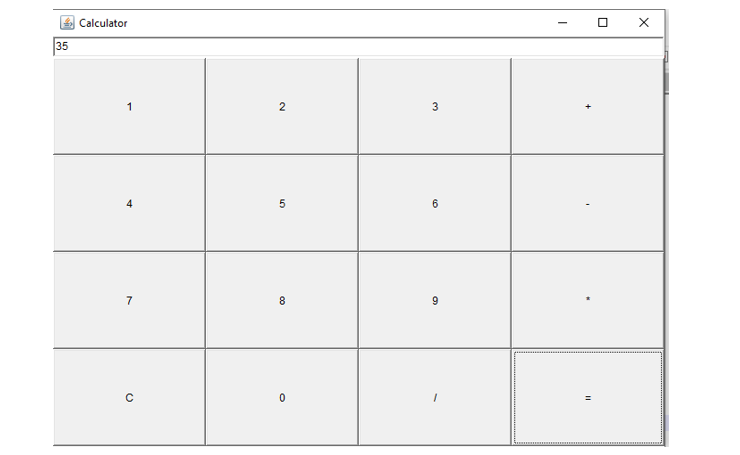
c.setTitle("Calculator");

c.setSize(700,500);

c.setVisible(true);

}

}



**6. 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.**

import java.applet.\*;

import java.awt.\*;

import java.awt.Graphics.\*;

import java.awt.event.\*;

public class Choiceq extends Applet implements ItemListener{

int select;

Choice ch = new Choice();

public void init(){

Label l1 = new Label("Select Choice");

l1.setBounds(750,50,200,50);

add(l1);

//Choice ch = new Choice();

ch.addItem("Select Your Choice");

ch.addItem("Rectangle");

ch.addItem("Square");

ch.addItem("Circle");

ch.addItem("Triangle");

add(ch);

ch.addItemListener(this);

}

public void itemStateChanged(ItemEvent e){

select = ch.getSelectedIndex();

repaint();

}

public void paint(Graphics g){

g.setColor(Color.blue);

super.paint(g);

if(select ==1){

g.drawRect(280,100,160,40);

}

if(select == 2){

g.drawRect(50,50,100,100);

}

if(select == 3){

g.drawOval(150,150,100,100);

}

if(select == 4){

g.drawLine(120,130,280,130);

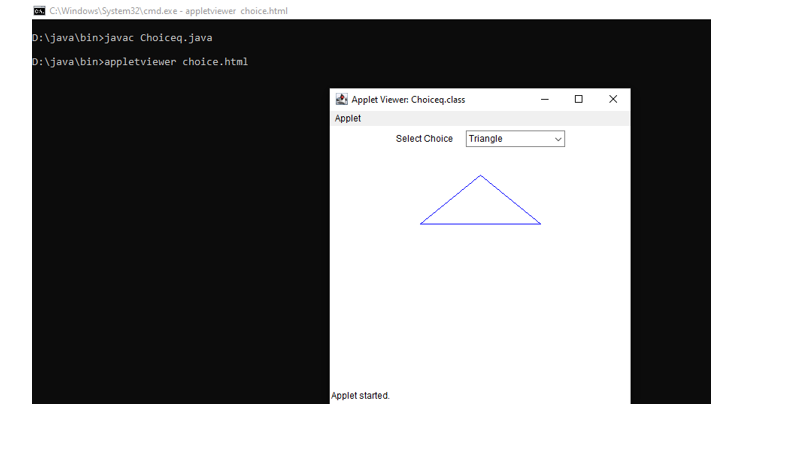
g.drawLine(120,130,200,65);

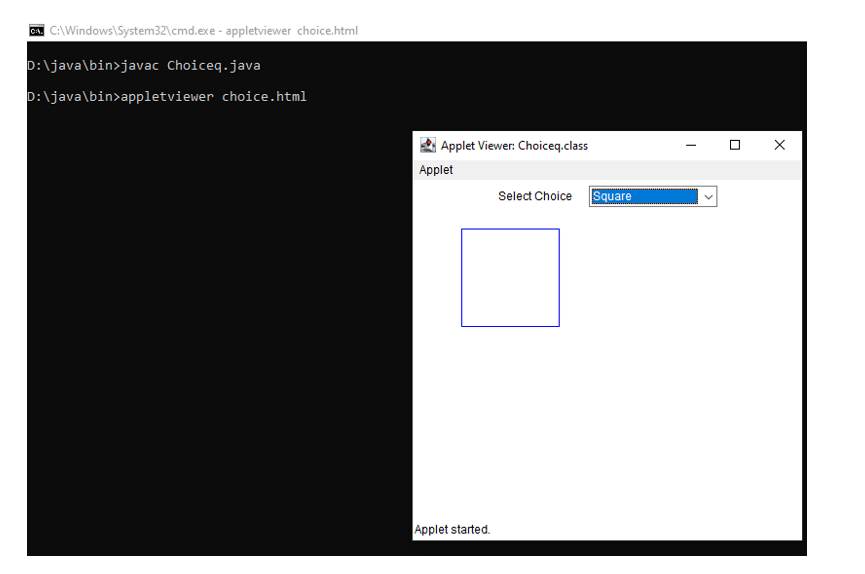
g.drawLine(200,65,280,130);

}

}

}





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

8. Develop a program to handle Key events.

**Course Outcome 6 (CO6):**

1. Program to list the sub directories and files in a given directory and also search for a file name.

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

3. Write a program to copy one file to another.

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

5. Client server communication using Socket – TCP/IP

**6. Client Server communication using DatagramSocket - UDP**

**Client.java**

import java.util.\*;

import java.io.\*;

import java.net.\*;

public class Client{

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

DatagramSocket client=new DatagramSocket();

InetAddress add=InetAddress.getByName("localhost");

Scanner sc=new Scanner(System.in);

System.out.println("Message server:");

String str=sc.next();

byte[] bufBytes=str.getBytes();

DatagramPacket datagramPacket=new DatagramPacket(bufBytes,bufBytes.length,add,4220);

client.send(datagramPacket);

client.close();

}

}

**Server.java**

import java.io.\*;

import java.net.\*;

public class Server{

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

DatagramSocket server=new DatagramSocket(4220);

//InetAddress add=InetAddress.getByName("localhost");

//String str="Ping from Client!!!";

byte[] buf=new byte[256];

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

server.receive(packet);

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

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

server.close();

}

}

import java.io.FileOutputStream;

public class FileOutputStreamExample{

public static void main(String[] args){

try{

FileOutputStream f = new FileOutputStream("D:\\testout.java");

f.write(90);

f.close();

System.out.println("Sucess..");

}catch(Exception e){

System.out.println(e);

}

}

}

1) To write program to list the sub directories and files in a given directory and also search for

a file name-

CODEimport

java.io.\*;

public class co6q1 {

public static void main(String[] args) {

File file = new File("../../../");

String[] fileList = file.list();

for(String str : fileList) {

System.out.println(str);

}

FilenameFilter filter = new FilenameFilter() {

public boolean accept (File dir, String name) {

return name.startsWith("A");

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

String[] children = file.list(filter);

if (children == null) {

System.out.println("Either dir does not exist or is not a directory");

} else {

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

String filename = children[i];

System.out.println(filename);

}}}}

**OUTPUT**

2)

To write a program to write to a file, then read from the file and display the contents on the

console.

Codeimport

java.io.BufferedReader;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class co6q2 {

public static void main(String[] args) {

try {

FileWriter writer = new FileWriter("java\_write.txt",true);

writer.write("new file is created");

writer.close();

FileReader reader = new FileReader("java\_write.txt");

BufferedReader br= new BufferedReader(reader);

String line;

System.out.println("Data read from the file");

while ((line = br.readLine()) != null) {

System.out.println(line);

} reader.close();

} catch (IOException e) {

System.out.println("-----Error-----");

}}

}

Output3)

To write a program to copy one file to another-

Codeimport

java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

public class co6q3 {

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

// TODO Auto-generated method stub

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

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

int i;

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

fileoutput.write(i);

}

System.out.println("copied");

fileinput.close();

fileoutput.close();

}

}

Output4)

To write a program that reads from a file having integers. Copy even numbers and odd

numbers to separate files.

Codeimport

java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

public class co6q4 {

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

// TODO Auto-generated method stub

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

FileOutputStream destination\_odd = new FileOutputStream ("odd.txt");

FileOutputStream destination\_even = new FileOutputStream

("even.txt");

int i;

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

if(i%2==0) {

destination\_even.write(i);

}

else {

destination\_odd.write(i);

}}

System.out.println("copied");

source.close();

destination\_even.close();

destination\_odd.close();

}

}

Output

.