COLLEGE OF ENGINEERING CHERTHALA

LAB RECORD

20MCA132 – OBJECT ORIENTED PROGRAMMING LAB

CERTIFICATE

This is certified to be bonafide works of Mr./Ms. …………………………..…………………., In the class ………………...., Reg. No. ……………………., of College of Engineering Cherthala, during the academic year 2022-23.

Teacher In Charge **External Examiner**

Internal Examiner

INDEX

12	Create a class Person with data members , Name, Gender,Address,Age, and constructors. Initialize another class employee that inherits the class person. Create another class teacher and inherit the class employee. Use array of objects to display details of N teachers.	31	27/04/2023	
13	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	35	27/04/2023	
14	Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.	39	01/06/2023	
15	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.	41	01/06/2023	
16	Prepare bill with the given format using calculate method from interface.	45	15/06/2023	
17	Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given number	49	15/06/2023	
18	Write a user defined exception class to authenticate the user’s name and password.	53	22/06/2023	
19	Define 2 classes; one for generating a multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)	55	22/06/2023	

20	Program to draw Circle, Rectangle, Line in Applet.	59	29/06/2023	
21	Program to handle all mouse events and window events.	61	29/06/2023	
22	Write a program to write to a file, then read from the file and display the contents on the console.	63	13/07/2023	
23	Write a program to copy one file to another.	65	13/07/2023	

PROGRAM NO :1

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

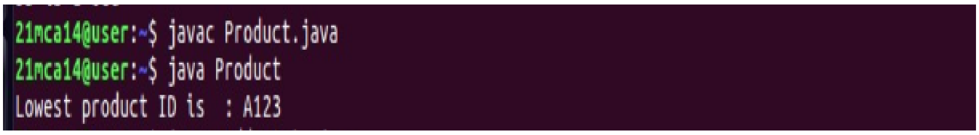
ALGORITHM :

```
Step 1  : Define a class Product with data members pcode,pname,price.
Step 2  : Read the details (product code,name,price,..etc) of three products from the user.
Step 3  : Create three objects for Product class and initialize it with the details read from
          the user.
Step 4  : Compare the prices of three products.
Step 5  : Print the product which has the lowest price.
```

SOURCE CODE :

```
import java.util.Scanner;
public class Product {

        String pcode, pname; int price;
        Product(String pcode_get, String pname_get, int price_get) {
                pcode = pcode_get; pname = pname_get; this.price = price_get;
        }
        void compare(Product p1,Product p2){
                if (price <= p1.price && price <= p2.price)
                        System.out.println("Lowest product ID is : " + pcode);
                if (p2.price <= price && p2.price <= price)
                        System.out.println("Lowest product ID is : " + p2.pcode);
                if (p1.price <= p2.price && p1.price <= price)
                        System.out.println("Lowest product ID is : " + p1.pcode);
        }}
class Product_details{

        public static void main(String[] args) {
                Product p_1 = new Product("A123", "TV", 2);
                Product p_2 = new Product("B123", "RADIO", 21);
                Product p_3 = new Product("C123", "DVD", 3);
                p_1.compare(p_2,p_3);}
}
```

OUTPUT :



```
21mca14@user:~$ javac Product.java
21mca14@user:~$ java Product
Lowest product ID is  : A123
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :2

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

ALGORITHM :

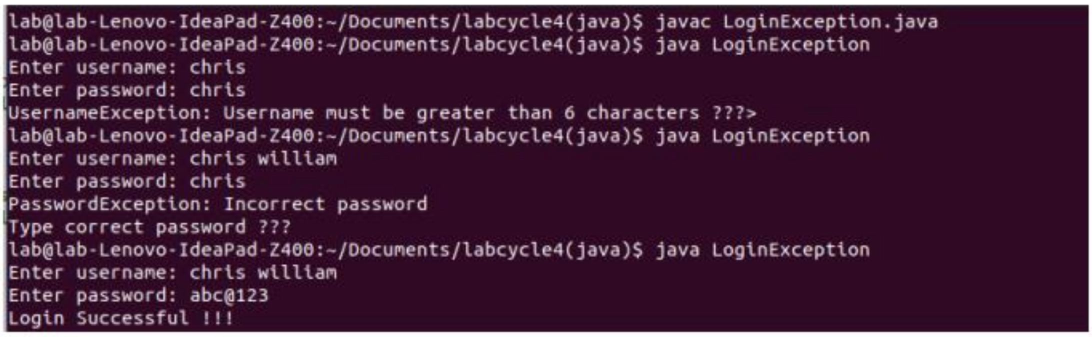
```
Step 1 : Start
Step 2 : Read username and password
Step 3 : Find the length of username
Step 4 : In try block define cases where exceptions may arise
        Step 4.1 : if username length is less than 6
                call UsernameException which inherits class Exception
                by passing message "Username must be greater than 6 characters".
        Step 4.2 : else if password not equal to predifined string
                call  PasswordException which inherits class Exception
                by passing message "Incorrect password Type correct password"
        Step 4.3 : else print  "Login Successful !!"
Step 5 : if there is any of the above Userdefined Exception catch the exception and print
        the corresponding message.
Step 6 : Stop
```

SOURCE CODE :

```
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 LoginException{
        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("abc@123")){
                                        throw new PasswordException("Incorrect password\nType
                                        correct password ???");}
                                else
                                        System.out.println("Login Successful !!!");
                        }
                        catch (UsernameException e){
                                System.out.println(e);
                        }
                        catch (PasswordException p){
                                System.out.println(p);
                        }
                }
        }
```

OUTPUT :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac LoginException.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LoginException
Enter username: chris
Enter password: chris
UsernameException: Username must be greater than 6 characters ???>
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LoginException
Enter username: chris william
Enter password: chris
PasswordException: Incorrect password
Type correct password ???
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LoginException
Enter username: chris william
Enter password: abc@123
Login Successful !!!
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :3

AIM : Write a program to perform complex number addition.

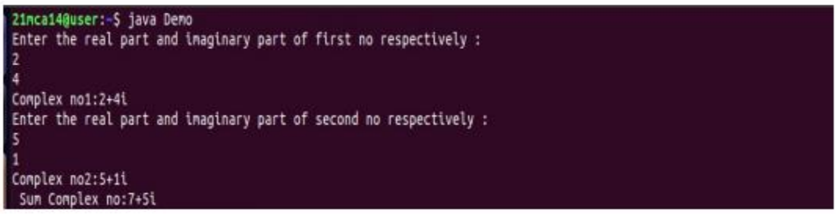
ALGORITHM :

Step 1 : Read the two complex numbers
Step 2 : Perform addition of real parts as well as imaginary parts of two numbers. Store the result.
Step 3 : Print the result.

SOURCE CODE :

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
class Complex{
        int real,img;
        Complex(int treal,int timg){
                real=treal;
                img=timg;
        }
        Complex add_complexno(Complex c1,Complex c2){
                Complex temp = new Complex();
                temp.real=c1.real+c2.real;
                temp.img=c1.img+c2.img;
                return temp;
        }
}
public class Demo{
        public static void main(String args[]) throws IOException{
                BufferedReader r =new BufferedReader(new
                        InputStreamReader(System.in));
                System.out.println("Enter the real part and imaginary part of first no
                respectively :  ");
                int a1=Integer.parseInt(r.readLine());
                int b1=Integer.parseInt(r.readLine());
                Complex c1 = new Complex(a1,b1);
                System.out.println("Complex no 1 : " +c1.real +  "+" + c1.img + "i");
                System.out.println("Enter the real part and imaginary part of second no
                respectively :  ");
                int a2=Integer.parseInt(r.readLine());
                int b2=Integer.parseInt(r.readLine());
                Complex c2 = new Complex(a2,b2);
                System.out.println("Complex no 2 : "+ c2.real + "+" + c2.img + "i");
                Complex c3 = new Complex();
                c3 = c3.add_complexno(c1,c2);
                System.out.println(" sum complex no: " + c3.real + "+" + c3.img + "i");
        }
}
```

OUTPUT :



```
21mca14@user:~$ java Demo
Enter the real part and imaginary part of first no respectively :
2
4
Complex no1:2+4i
Enter the real part and imaginary part of second no respectively :
5
1
Complex no2:5+1i
 Sum Complex no:7+5i
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :4

AIM : Read a matrix from the console and check whether it is symmetric/not

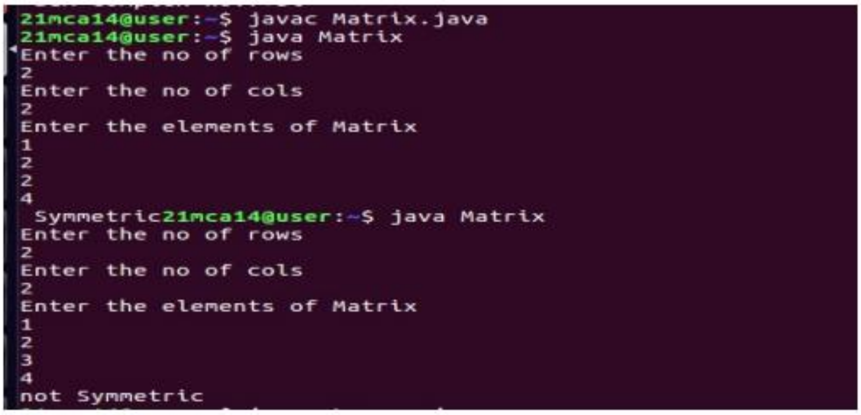
ALGORITHM :

Step 1 : Read the order of the matrix.
Step 2 : If the number of rows and columns are not same, then print matrix is not symmetric and exit.
Step 3 : Else, read the matrix.
Step 4 : Find the transpose of the matrix by interchanging rows and columns.
Step 5 : Check if the original matrix is same as its transpose. If it is same, then print matrix is symmetric. Else, print matrix is not symmetric.

SOURCE CODE :

```
import java.util.Scanner;
class Matrix{
        public static void main(String args[]){
                int i,j,rows,cols,f=0;
                Scanner n=new Scanner(System.in);
                System.out.println("Enter the no of rows: ");
                rows=n.nextInt();
                System.out.println("Enter the no of cols: ");
                cols=n.nextInt();
                if(rows!=cols)
                        System.out.print(" Not symmetric");
                else{
                        int num[][]= new int[rows][cols];
                        System.out.println("Enter the elements of Matrix: ");
                        for(i=0;i<rows;i++){
                                for(j=0;j<cols;j++){
                                        num[i][j]=n.nextInt();
                                }
                        }
                        for(i=0;i<rows;i++){
                                for(j=0;j<cols;j++){
                                        if(num[i][j]!=num[j][i]){
                                                f=1;
                                                break;
                                        }
                                }
                        }
                        if(f==1)
                                System.out.println("Not Symmetric");
                        else
                                    System.out.print(" Symmetric");
                }
        }
}
```

OUTPUT :



```
21mca14@user:~$ javac Matrix.java
21mca14@user:~$ java Matrix
Enter the no of rows
2
Enter the no of cols
2
Enter the elements of Matrix
1
2
2
4
 Symmetric21mca14@user:~$ java Matrix
Enter the no of rows
2
Enter the no of cols
2
Enter the elements of Matrix
1
2
3
4
not Symmetric
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :5

AIM : Create CPU with attribute price. Create an 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.

ALGORITHM :

Step 1 : Define a class CPU with data member price. Define an inner class called Processor with data members to hold the specifications of processor such as no.of cores, manufacturer. Also define a static class RAM(use static keyword) with member variables to hold the information about the memory like memory size, manufacturer etc.

Step 2 : Initialize member variables of classes CPU,RAM and Processor.

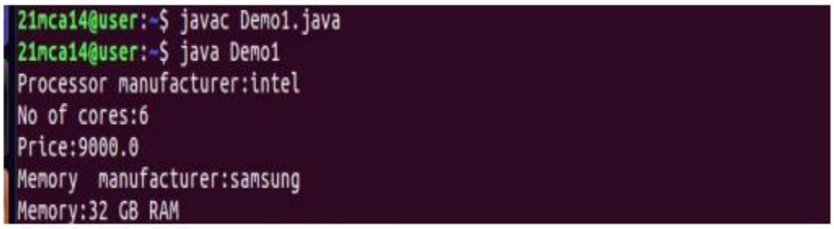
Step 3 : Create an object for CPU class and invoke a method to instantiate an object for inner class Processor then invoke another method to display the specification of the Processor.

Step 4 : Create an object for static inner class RAM of Class CPU and invoke a method to display the specification of the class RAM

SOURCE CODE :

```
class CPU{
        double price=9000;
        void display_1(){
                Processor obj_2=new Processor();
                obj_2.display_2();
        }
        class Processor{
                int no_of_cores=6;
                String pross_manufact="intel";
                void display_2(){
                        System.out.println("Processor manufacturer:" + pross_manufact);
                        System.out.println("No of cores:" + no_of_cores);
                        System.out.println("Price:" + price);
                }
        }
        static class RAM{
                String memory="32 GB RAM";
                String mmry_manufact="samsung";
                void display(){
                        System.out.println("Memory manufacturer:" + mmry_manufact);
                        System.out.println("Memory:" + memory);
                }
        }
}
class Demo1{
        public static void main(String args[]){
                CPU obj1=new CPU();
                obj1.display_1();
                CPU.RAM obj=new CPU.RAM();
                obj.display();
        }
}
```

OUTPUT :



```
21mca14@user:~$ javac Demo1.java
21mca14@user:~$ java Demo1
Processor manufacturer:intel
No of cores:6
Price:9000.0
Memory  manufacturer:samsung
Memory:32 GB RAM
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :6

AIM : Program to sort strings.

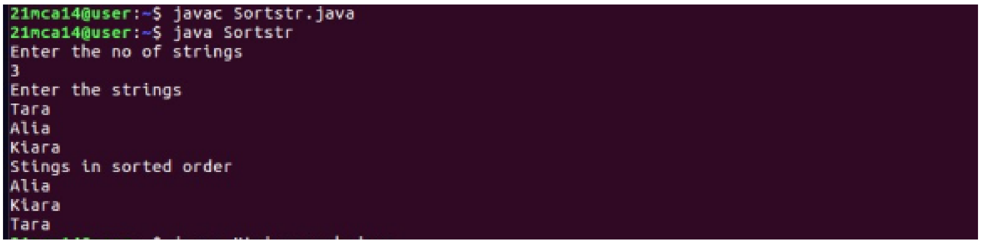
ALGORITHM :

```
Step 1  : Read the total number of strings.
Step 2  : Declare a string array to hold the strings.
Step 3  : Read the strings and store it into the array.
Step 4  : Use nested loops to sort the strings in the array in alphabetical order.
Step 5  : Use outer loop to hold the elements.
Step 6  : Use inner loop to compare with the remaining elements. ( Use compareTo() )
Step 7  : Swap the array elements.
Step 8  : Print the sorted array.
```

SOURCE CODE :

```
import java.util.Scanner;

class Sortstr{
        public static void main(String args[]){
                int i,j,n;
                String temp;
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter the no of strings");
                n=sc.nextInt();
                String str[]= new String[n];
                Scanner s=new Scanner(System.in);
                System.out.println("Enter the strings");
                for(i=0;i<n;i++)
                        str[i]=s.nextLine();
                for(i=0;i<n;i++){
                        for(j=i+1;j<n;j++){
                                if(str[i].compareTo(str[j])>0){
                                        temp=str[i];
                                        str[i]=str[j];
                                        str[j]=temp;
                                }
                        }
                }
                System.out.println("Stings in sorted order");
                for(i=0;i<n;i++)
                        System.out.println(str[i]);

        }
}
```

OUTPUT :



```
21mca14@user:~$ javac Sortstr.java
21mca14@user:~$ java Sortstr
Enter the no of strings
3
Enter the strings
Tara
Alia
Kiara
Stings in sorted order
Alia
Kiara
Tara
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :7

AIM : Search an element in an array.

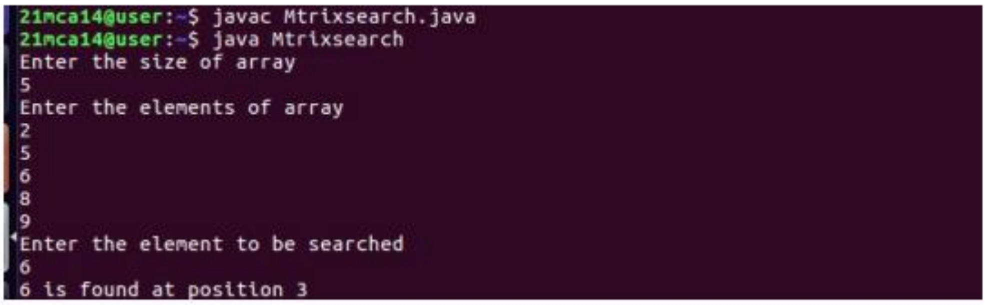
ALGORITHM :

Step 1 : Read the total number of elements in the array.
Step 2 : Declare an array to store the elements.
Step 3 : Read the elements into the array.
Step 4 : Read the element to be searched.
Step 5 : Traverse through the array and check if search value matches the element.
if it matches then print “value is found in respective position” else print “not found”.

SOURCE CODE :

```
import java.util.Scanner;
class Mtrixsearch{
        public static void main(String args[]){
                int i,size,f=0;
                Scanner n=new Scanner(System.in);
                System.out.println("Enter the size of array");
                size=n.nextInt();
                int arr[]= new int[size];
                System.out.println("Enter the elements of array");
                for(i=0;i<size;i++)
                        arr[i]=n.nextInt();
                System.out.println("Enter the element to be searched");
                int a=n.nextInt();
                for(i=0;i<size;i++){
                        if(arr[i]==a){
                                f=1;
                                break;
                        }
                }
                if(f==1)
                        System.out.println(a+" is found at position "+(i+1));
                else
                        System.out.println("not found");
        }
}
```

OUTPUT :



```
21mca14@user:~$ javac Mtrixsearch.java
21mca14@user:~$ java Mtrixsearch
Enter the size of array
5
Enter the elements of array
2
5
6
8
9
Enter the element to be searched
6
6 is found at position 3
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :8

AIM : Perform string manipulation (using Built-in methods of String Class and StringBuffer Class).

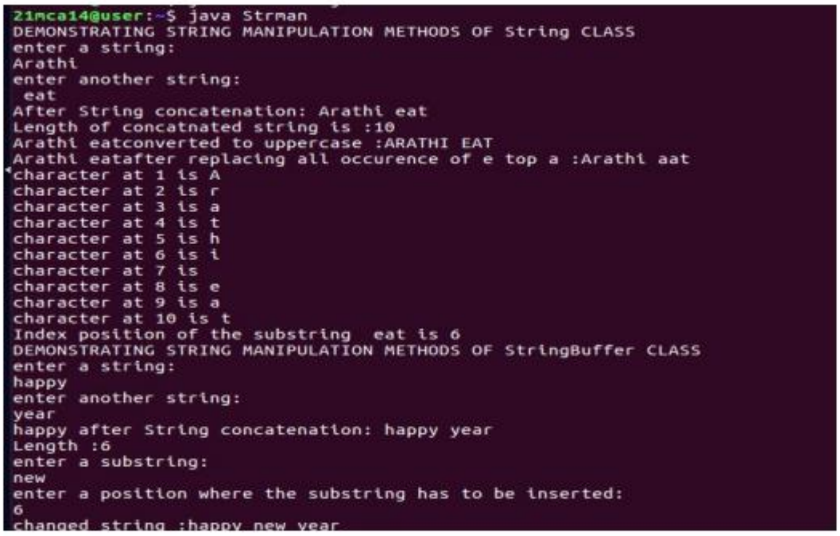
ALGORITHM :

```
Step 1  : Read two Strings str1 and str2.
Step 2  : Use various String manipulation built-in methods of String class
          a) concatenate 2 strings using concat() [str1.concat(str2)]
          b) find length of string using length()  [str1.length()]
          c) convert string to uppercase using toUpperCase() [str1.toUpperCase()]
          d) replace a character by other using replace() [str1.replace('e','a')]
          e) convert a string to character array using tocharArray()
          f) find the index position of a substring using indexOf()
Step 3  :Use various String manipulation built-in methods of StringBuffer class
Step 4  :  Create an object for  StringBuffer class and intialize it with a string
          a) concatenate  strings using append() [str.append(temp1)]
          b) find length of string using length()  [str1.length()]
          c)  read the position to insert a substring using insert() [str.insert(pos,temp2)]
```

SOURCE CODE :

```
import java.util.*;

class Strman{
        public static void main(String args[]){
                Scanner sc =new Scanner(System.in);
                System.out.println("DEMONSTRATING STRING MANIPULATION
                METHODS OF String CLASS");
                System.out.println("enter a string: ");
                String str1=sc.nextLine();
                System.out.println("enter another string: ");
                String str2=sc.nextLine();
                String str3=str1.concat(str2);
                System.out.println("After String concatenation: " + str3);
                System.out.println("Length of concatnated string is :"+ str3.length());
                System.out.println(str3+"converted to uppercase :"+ str3.toUpperCase());
                System.out.println(str3+"after replacing all occurence of e top a :"+
                str3.replace('e','a'));
                char ch[]=str3.toCharArray();
                for(int i=0;i<ch.length;i++)
                        System.out.println("character at "+(i+1)+" is "+ch[i]);
                System.out.println("Index position of the substring "+str2+" is "
                +str3.indexOf(str2));
                System.out.println("DEMONSTRATING STRING MANIPULATION
                METHODS OF StringBuffer CLASS");
                System.out.println("enter a string: ");
                String temp=sc.nextLine();
                StringBuffer str=new StringBuffer(temp);
                System.out.println("enter another string: ");
                String temp1=sc.nextLine();
                str.append(temp1);
                System.out.println(temp+"after String concatenation: " + str);
                System.out.println("Length :"+ str1.length());
                System.out.println("enter a substring: ");
                String temp2=sc.nextLine();
                System.out.println("enter a position where the substring has to be
                inserted: ");
                int pos =sc.nextInt();
                str.insert(pos,temp2);
                System.out.println("changed string :"+str);
        }
}
```

OUTPUT :



```
21mca14@user:~$ java Strman
DEMONSTRATING STRING MANIPULATION METHODS OF String CLASS
enter a string:
Arathi
enter another string:
 eat
After String concatenation: Arathi eat
Length of concatnated string is :10
Arathi eatconverted to uppercase :ARATHI EAT
Arathi eatafter replacing all occurence of e top a :Arathi aat
character at 1 is A
character at 2 is r
character at 3 is a
character at 4 is t
character at 5 is h
character at 6 is i
character at 7 is
character at 8 is e
character at 9 is a
character at 10 is t
Index position of the substring  eat is 6
DEMONSTRATING STRING MANIPULATION METHODS OF StringBuffer CLASS
enter a string:
happy
enter another string:
year
happy after String concatenation: happy year
Length :6
enter a substring:
new
enter a position where the substring has to be inserted:
6
changed string :happy new year
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :9

AIM : Program to create a class for Employee having attributes eNo, eName,Salary. Read n employ information and search for an employee given eNo using the concept of array of objects.

ALGORITHM :

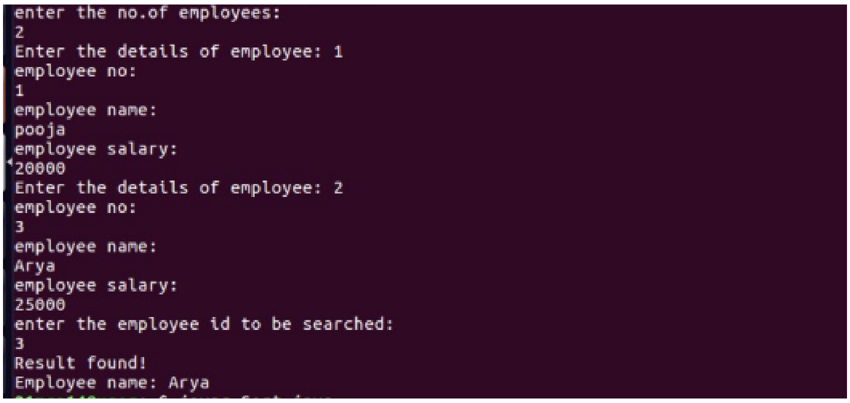
```
Step 1  : Define a class Employee with instance variables eNo, eName, Salary.
Step 2  : Define a parameterized  constructor to initialize theinstance variables.
Step 3  : Read the count of total number of employees.
Step 4  : Define an array of object , “obj”.
Step 5  : Read the details of each employee and store it in obj
Step 6  : Read the eNo of the employee to be searched.
Step 7  : Use loop to traverse through the array.
Step 8  :  Check whether the eNo to be searched is present in the array “obj”.
Step 9  : If yes, then print “Found”. Else, print “Not found”
```

SOURCE CODE :

```
import java.util.*;

class Employee{
        int e_no;
        String e_name;
        float e_salary;
        Employee(int id,String name,float sal){
                e_no=id;
                e_name=name;
                e_salary=sal;
        }
}
class Main{
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                Scanner s = new Scanner(System.in);
                int id,i,flag=0;
                String name;
                float sal;
                System.out.println("enter the no.of employees: ");
                int n=sc.nextInt();
                Employee[] obj=new Employee[n];
                for(i=0;i<n;i++){
                        System.out.println("Enter the details of employee: " + (i+1));
                        System.out.println("employee no: ");
```

```
                    id=sc.nextInt();
                    System.out.println("employee name: ");
                    name=s.nextLine();
                    System.out.println("employee salary: ");
                    sal=sc.nextFloat();
                    obj[i]=new Employee(id,name,sal);
            }
            System.out.println("enter the employee id to be searched: ");
            int temp=sc.nextInt();
            for(i=0;i<n;i++){
                    if(obj[i].e_no==temp){
                            System.out.println("Result found!");
                            flag=1;
                            break;
                    }
            }
            if(flag==1)
                    System.out.println("Employee name: "+ obj[i].e_name);
            else
                    System.out.println("not found" );
    }
}
```

OUTPUT :



```
enter the no.of employees:
2
Enter the details of employee: 1
employee no:
1
employee name:
pooja
employee salary:
20000
Enter the details of employee: 2
employee no:
3
employee name:
Arya
employee salary:
25000
enter the employee id to be searched:
3
Result found!
Employee name: Arya
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :10

AIM : Using the concept of method overloading find the area of different shapes rectangle, circle and square.

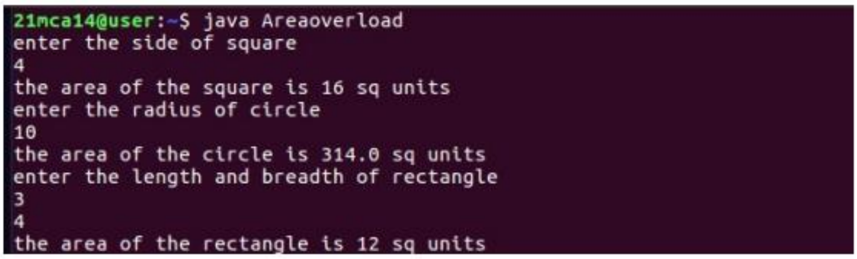
ALGORITHM :

```
Step 1 : Define a class with three methods of same name but with different arguments
         inorder to calculate the area of three different shapes such as rectangle, circle
         and square.
Step 2 : Read the particulars of the shapes and store it
Step 3 : Create three objects and invoke the methods  defined in the step 1 by passing
         the stored values as parameters .
Step 4 : Display the calculated area of the three shapes.
```

SOURCE CODE :

```
import java.util.*;
class Area1{
        void area(int x){
                System.out.println("the area of the square is "+x*x+" sq units");
        }
        void area(int x, int y){
                System.out.println("the area of the rectangle is "+x*y+" sq units");
        }
        void area(double x){
                double z = 3.14 * x * x;
                System.out.println("the area of the circle is "+z+" sq units");
        }
}
class Areaoverload{
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                Area1 obj = new Area1();
                System.out.println("enter the side of square");
                int side = sc.nextInt();
                obj.area(side);
                System.out.println("enter the radius of circle");
                double r = sc.nextDouble();
                obj.area(r);
                System.out.println("enter the length and breadth of rectangle");
                int l = sc.nextInt();
                int b = sc.nextInt();
                obj.area(l,b);




        }
}
```

OUTPUT :



```
21mca14@user:~$ java Areaoverload
enter the side of square
4
the area of the square is 16 sq units
enter the radius of circle
10
the area of the circle is 314.0 sq units
enter the length and breadth of rectangle
3
4
the area of the rectangle is 12 sq units
```

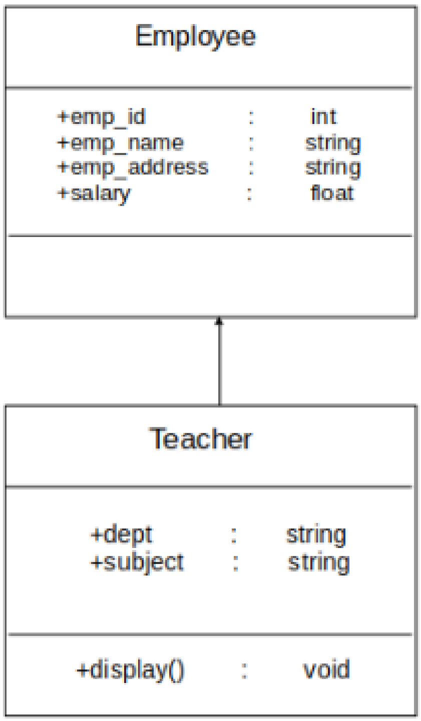
RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :11

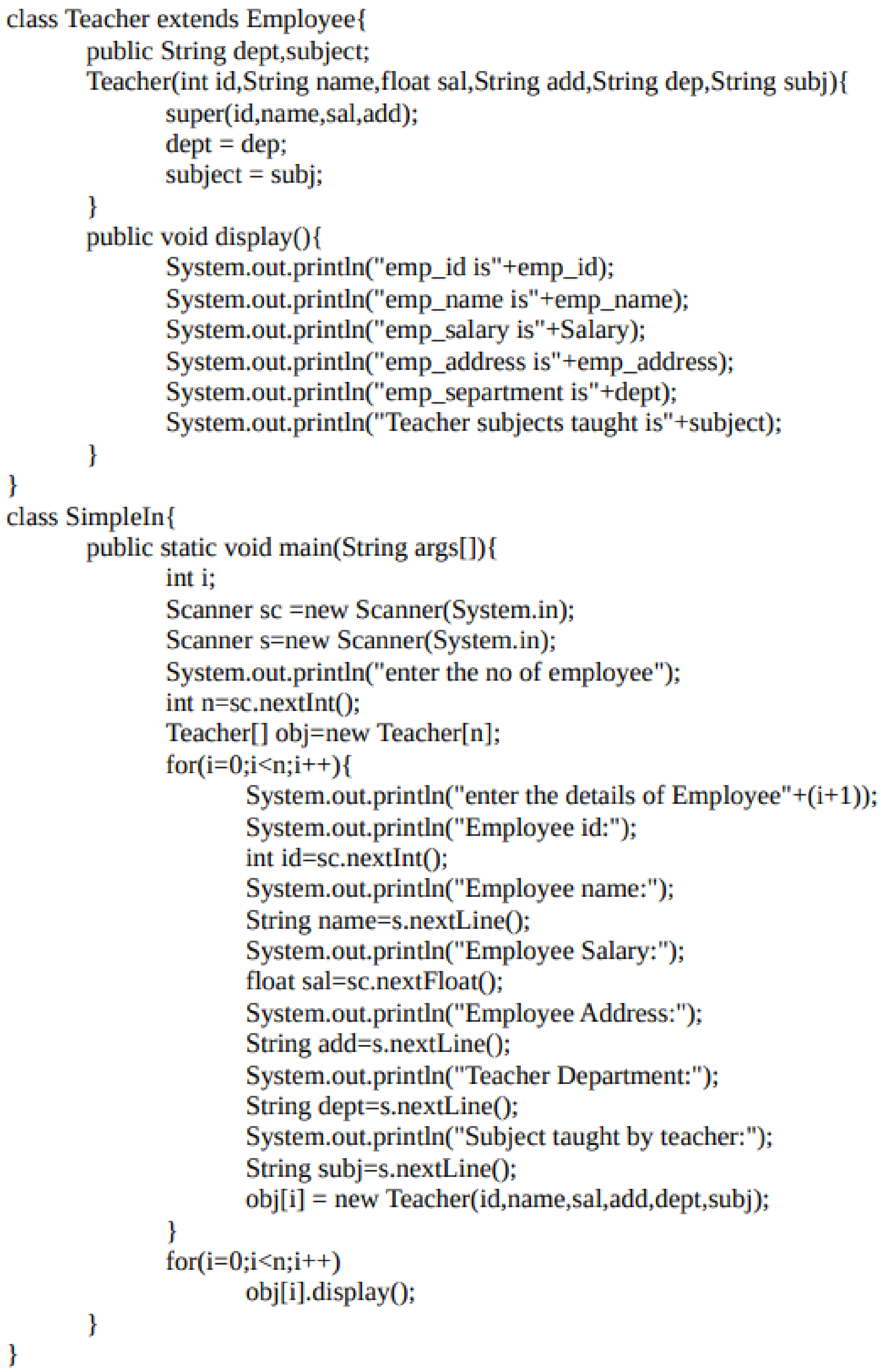
AIM : Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

CLASS DIAGRAM :



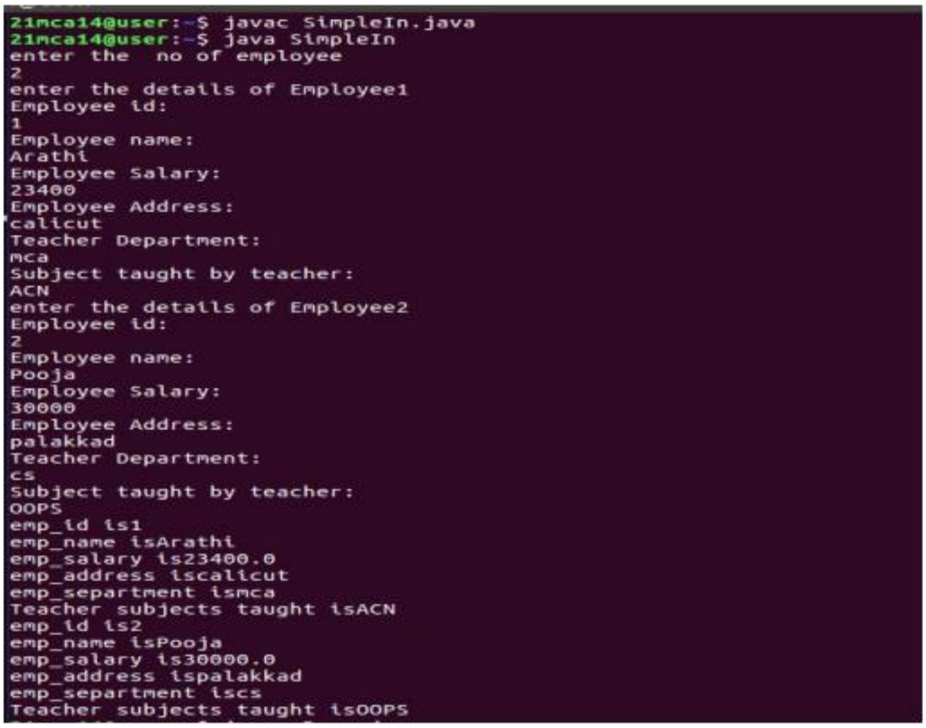
SOURCE CODE :

```
import java.util.*;
class Employee{
        public int emp_id;
        public String emp_name,emp_address;
        public float salary;
        Employee(int id,String name,float sal,String add){
                emp_id=id;
                emp_name=name;
                salary=sal;
                emp_address=add;
         }
}
```



```
class Teacher extends Employee{
        public String dept,subject;
        Teacher(int id,String name,float sal,String add,String dep,String subj){
                super(id,name,sal,add);
                dept = dep;
                subject = subj;
        }
        public void display(){
                System.out.println("emp_id is"+emp_id);
                System.out.println("emp_name is"+emp_name);
                System.out.println("emp_salary is"+Salary);
                System.out.println("emp_address is"+emp_address);
                System.out.println("emp_separtment is"+dept);
                System.out.println("Teacher subjects taught is"+subject);
        }
}
class SimpleIn{
        public static void main(String args[]){
                int i;
                Scanner sc =new Scanner(System.in);
                Scanner s=new Scanner(System.in);
                System.out.println("enter the no of employee");
                int n=sc.nextInt();
                Teacher[] obj=new Teacher[n];
                for(i=0;i<n;i++){
                        System.out.println("enter the details of Employee"+(i+1));
                        System.out.println("Employee id:");
                        int id=sc.nextInt();
                        System.out.println("Employee name:");
                        String name=s.nextLine();
                        System.out.println("Employee Salary:");
                        float sal=sc.nextFloat();
                        System.out.println("Employee Address:");
                        String add=s.nextLine();
                        System.out.println("Teacher Department:");
                        String dept=s.nextLine();
                        System.out.println("Subject taught by teacher:");
                        String subj=s.nextLine();
                        obj[i] = new Teacher(id,name,sal,add,dept,subj);
                }
                for(i=0;i<n;i++)
                        obj[i].display();
        }
}
```

OUTPUT :



```
21mca14@user:~$ javac SimpleIn.java
21mca14@user:~$ java SimpleIn
enter the  no of employee
2
enter the details of Employee1
Employee id:
1
Employee name:
Arathi
Employee Salary:
23400
Employee Address:
calicut
Teacher Department:
mca
Subject taught by teacher:
ACN
enter the details of Employee2
Employee id:
2
Employee name:
Pooja
Employee Salary:
30000
Employee Address:
palakkad
Teacher Department:
cs
Subject taught by teacher:
OOPS
emp_id is1
emp_name isArathi
emp_salary is23400.0
emp_address iscalicut
emp_separtment ismca
Teacher subjects taught isACN
emp_id is2
emp_name isPooja
emp_salary is30000.0
emp_address ispalakkad
emp_separtment iscs
Teacher subjects taught isOOPS
```

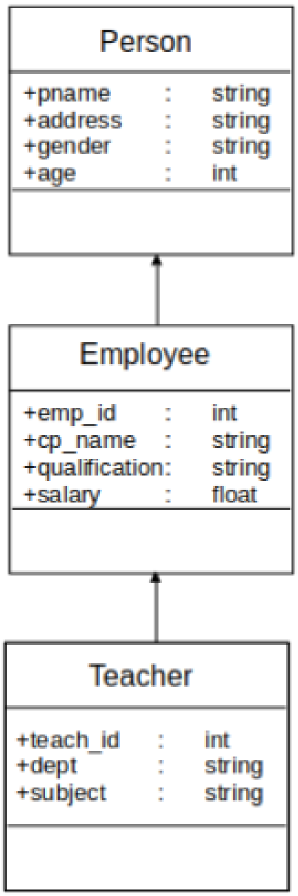
RESULT :

Program is successfully executed and output is verified.

PROGRAM 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, Teacher id and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

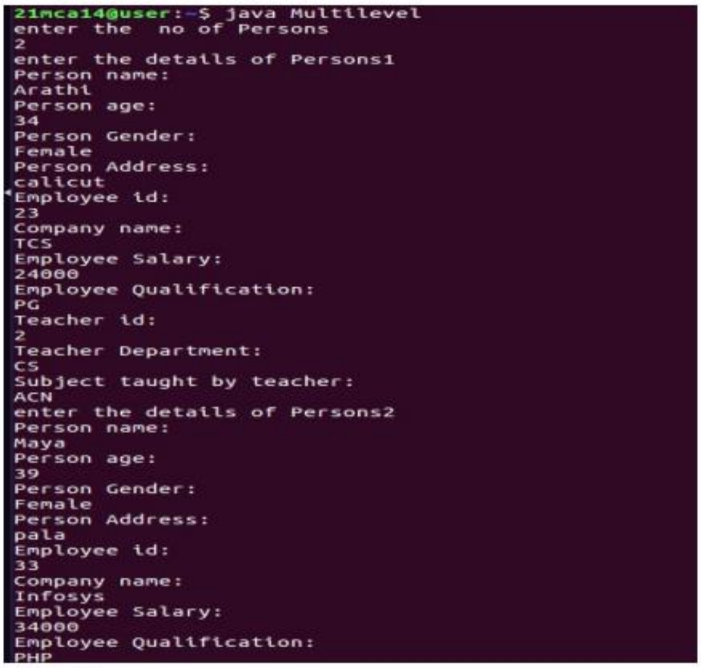
CLASS DIAGRAM:



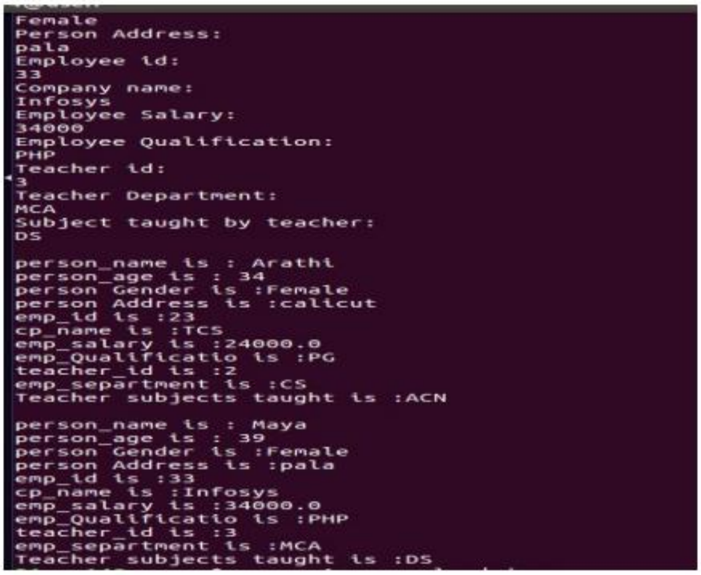
SOURCE CODE :

```
import java.util.*;
        class Person{
                public int age;
                public String pname,gender,address;
        Person(String name,String sex,String add,int age){
            pname=name;
            this.age=age;
            gender=sex;
            address=add;
        }
}
class Employee extends Person{
    public int emp_id;
    public String cp_name,qualification;
    public float salary;
    Employee(int id,String cname,float sal,String qualifctn,String name,String
        sex,String add,int age){
            super(name,sex,add,age);
            emp_id=id;
            cp_name=cname;
            salary=sal;
            qualification=qualifctn;
    }
}
class Teacher extends Employee{
    public int teach_id;
    public String dept,subject;
    Teacher(int id,String cname,float sal,String quali,String name,String sex,String
        add,int age,String dep,String subj,int tid){
            super(id,cname,sal,quali,name,sex,add,age);
            teach_id=tid;
            dept = dep;
            subject = subj;
    }
    public void display(){
        System.out.println();
        System.out.println("person_name is : "+pname);
        System.out.println("person_age is : "+age);
        System.out.println("person Gender is :"+gender);
        System.out.println("person Address is :" +address);
        System.out.println("emp_id is :"+emp_id);
        System.out.println("cp_name is :"+cp_name);
        System.out.println("emp_salary is :"+Salary);
        System.out.println("emp_Qualificatio is :"+qualification);
        System.out.println("teacher_id is :"+teach_id);
        System.out.println("emp_separtment is :"+dept);
        System.out.println("Teacher subjects taught is :"+subject);
    }
}
class Multilevel{
    public static void main(String args[]){
        int i;
        Scanner sc =new Scanner(System.in);
        Scanner s =new Scanner(System.in);
        System.out.println("enter the no of Persons");
        int n=sc.nextInt();
        Teacher[] obj=new Teacher[n];
        for(i=0;i<n;i++){
            System.out.println("enter the details of Persons"+(i+1));
            System.out.println("Person name:");
            String name=s.nextLine();
            System.out.println("Person age:");
            int age=sc.nextInt();
            System.out.println("Person Gender:");
            String sex=s.nextLine();
            System.out.println("Person Address:");
            String add=s.nextLine();
            System.out.println("Employee id:");
            int id=sc.nextInt();
            System.out.println("Company name:");
            String cname=s.nextLine();
            System.out.println("Employee Salary:");
            float sal=sc.nextFloat();
            System.out.println("Employee Qualification:");
            String quali=s.nextLine();
            System.out.println("Teacher id:");
            int tid=sc.nextInt();
            System.out.println("Teacher Department:");
            String dep=s.nextLine();
            System.out.println("Subject taught by teacher:");
            String subj=s.nextLine();
            obj[i] = new Teacher
    (id,cname,sal,quali,name,sex,add,age,dep,subj,tid);
    }
    for(i=0;i<n;i++)
        obj[i].display();
    }
}
```

OUTPUT :



```
21mca14@user:~$ java Multilevel
enter the  no of Persons
2
enter the details of Persons1
Person name:
Arathi
Person age:
34
Person Gender:
Female
Person Address:
calicut
Employee id:
23
Company name:
TCS
Employee Salary:
24000
Employee Qualification:
PG
Teacher id:
2
Teacher Department:
CS
Subject taught by teacher:
ACN
enter the details of Persons2
Person name:
Maya
Person age:
39
Person Gender:
Female
Person Address:
pala
Employee id:
33
Company name:
Infosys
Employee Salary:
34000
Employee Qualification:
PHP
```



```
Female
Person Address:
pala
Employee id:
33
Company name:
Infosys
Employee Salary:
34000
Employee Qualification:
PHP
Teacher id:
3
Teacher Department:
MCA
Subject taught by teacher:
DS

person_name is : Arathi
person_age is : 34
person Gender is :Female
person Address is :calicut
emp_id is :23
cp_name is :TCS
emp_salary is :24000.0
emp_Qualificatio is :PG
teacher_id is :2
emp_separtment is :CS
Teacher subjects taught is :ACN

person_name is : Maya
person_age is : 39
person Gender is :Female
person Address is :pala
emp_id is :33
cp_name is :Infosys
emp_salary is :34000.0
emp_Qualificatio is :PHP
teacher_id is :3
emp_separtment is :MCA
Teacher subjects taught is :DS
```

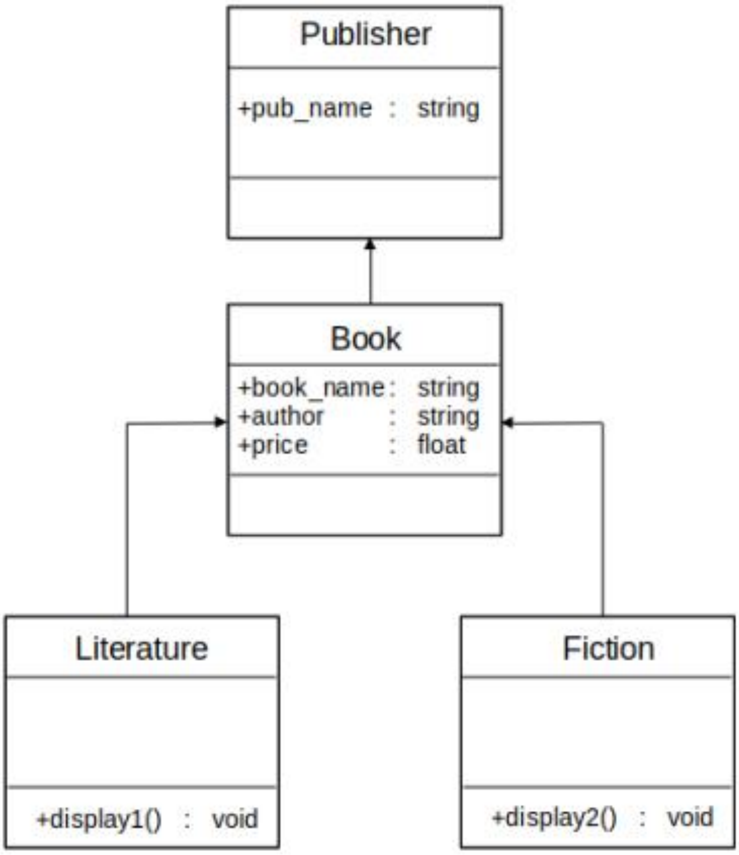
RESULT :

Program is successfully executed and output is verified.

PROGRAM 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.

CLASS DIAGRAM :



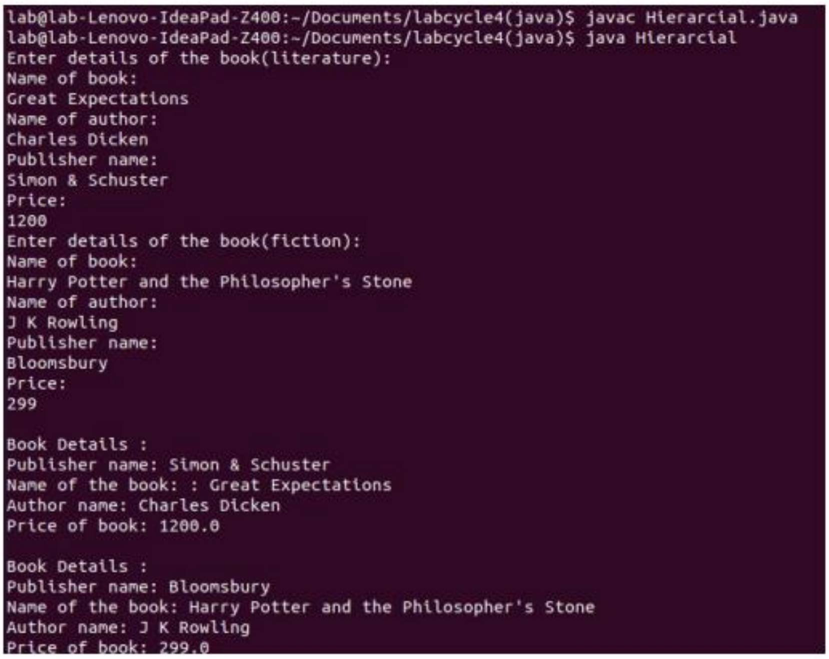
SOURCE CODE :

```
import java.util.Scanner;
class Publisher{
        public String pub_name;
        Publisher(String name){
                pub_name=name;
        }
}
class Book extends Publisher{
        public String book_name,author;
```

```
    public float price;
    Book(String pname,String aname,String bname,float price){
        super(pname);
        book_name=bname;
        author=aname;
        this.price=price;
    }
}
class Literature extends Book{
    Literature(String bname,String aname,String pname,float p1){
        super(pname,aname,bname,p1);
    }
    public void display1(){
        System.out.println("Publisher name: " + pub_name);
        System.out.println("Name of the book: : " + book_name);
        System.out.println("Author name: " + author);
        System.out.println("Price of book: " + price);
    }
}
class Fiction extends Book{
    Fiction(String bname,String aname,String pname,float p2){
        super(pname,aname,bname,p2);
    }
    public void display2(){
        System.out.println("Publisher name: " + pub_name);
        System.out.println("Name of the book: " + book_name);
        System.out.println("Author name: " + author);
        System.out.println("Price of book: " + price);
    }
}
class hierarcial{
public static void main(String args[]){
    Scanner sc=new Scanner(System.in);
    Scanner s=new Scanner(System.in);
    System.out.println("Enter details of the book(literature): ");
    System.out.println("Name of book: ");
    String b1name=sc.nextLine();
    System.out.println("Name of author: ");
    String a1name=sc.nextLine();
    System.out.println("Publisher name: ");
    String p1name=sc.nextLine();
    System.out.println("Price: ");
    float p1=s.nextFloat();
    System.out.println("Enter details of the book(fiction): ");
    System.out.println("Name of book: ");
    String b2name=sc.nextLine();
```

```
         System.out.println("Name of author: ");
        String a2name=sc.nextLine();
        System.out.println("Publisher name: ");
        String p2name=sc.nextLine();
        System.out.println("Price: ");
        float p2=s.nextFloat();
        System.out.println(" ");
        System.out.println("Book Details : ");
        Literature obj1=new Literature(b1name,a1name,p1name,p1);
        obj1.display1();
        System.out.println(" ");
        System.out.println("Book Details : ");
        Fiction obj2=new Fiction(b2name,a2name,p2name,p2);
        obj2.display2();
        }
}
```

OUTPUT :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Hierarcial.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Hierarcial
Enter details of the book(literature):
Name of book:
Great Expectations
Name of author:
Charles Dicken
Publisher name:
Simon & Schuster
Price:
1200
Enter details of the book(fiction):
Name of book:
Harry Potter and the Philosopher's Stone
Name of author:
J K Rowling
Publisher name:
Bloomsbury
Price:
299

Book Details :
Publisher name: Simon & Schuster
Name of the book: : Great Expectations
Author name: Charles Dicken
Price of book: 1200.0

Book Details :
Publisher name: Bloomsbury
Name of the book: Harry Potter and the Philosopher's Stone
Author name: J K Rowling
Price of book: 299.0
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM 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.

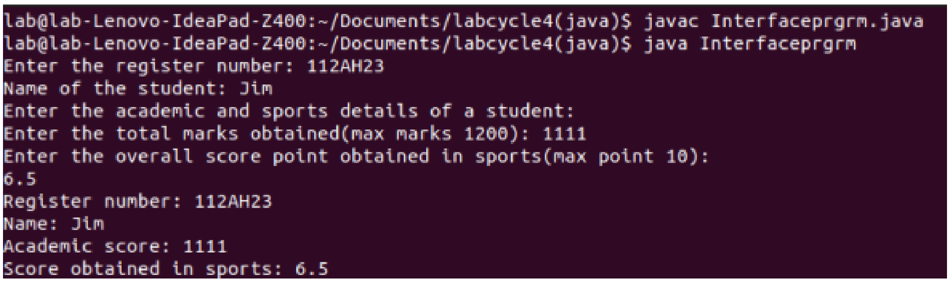
ALGORITHM :

Step 1 : Define two interfaces Student and Sports. Define the function prototype of method which sets the value of the instance variables of class which implemets it.
Step 2 : Define a class Result which implements the interfaces Student and Sports.
Step 3 : Define the methods specified in the interfcaes Student and Sports in class Result inorder to store the academic and sports details of a student. Also define method to display the academic and sports details.
Step 4 : Read the details(name,academic and sports) of a student.
Step 5 : Invoke the methods(inherit methods) of class Result to initialize the variables.
Step 6 : Display the details.

SOURCE CODE :

```
import java.util.*;
interface Student{
        void set_value1(String sname,int s,String regno);
}
interface Sports{
        void set_value2(float p);
}
class Result implements Student, Sports{
        public String name,reg_no;
        public int total;
        public float score_point;
        public void set_value1(String sname,int s,String regno){
                name=sname;
                total=s;
                reg_no=regno;
        }
        public void set_value2(float pt){
                score_point=pt;
        }
        public void display(){
                System.out.println("Register number: "+ reg_no);
                System.out.println("Name: " + name);
                System.out.println("Academic score: " + total);
                System.out.println("Score obtained in sports: " + score_point);
        }
}
class Interfaceprgrm{
        public static void main(String args[]){
                Scanner sc=new Scanner(System.in);
                Scanner s=new Scanner(System.in);
                int total;float pt;
                System.out.print("Enter the register number: ");
                String reg_no=sc.nextLine();
                System.out.print("Name of the student: ");
                String name=sc.nextLine();
                System.out.println("Enter the academic and sports details of a student: ");
                System.out.print("Enter the total marks obtained(max marks 1200): ");
                total=s.nextInt();
                System.out.println("Enter the overall score point obtained in sports(max
point 10): ");
                pt=s.nextFloat();
                Result obj=new Result();
                obj.set_value1(name,total,reg_no);
                obj.set_value2(pt);
                obj.display();
        }
}
```

<u>OUTPUT</u> :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Interfaceprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Interfaceprgrm
Enter the register number: 112AH23
Name of the student: Jim
Enter the academic and sports details of a student:
Enter the total marks obtained(max marks 1200): 1111
Enter the overall score point obtained in sports(max point 10):
6.5
Register number: 112AH23
Name: Jim
Academic score: 1111
Score obtained in sports: 6.5
```

<u>RESULT</u> :

Program is successfully executed and output is verified.

PROGRAM 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.

ALGORITHM :

```
Step 1  : Create an interface to define the prototypes of functions, area() and  perimeter()
          which are used to find  the area and perimeter of circle and rectangle.
Step 2 : Define two classess, Circle having instance variable radius and Rectangle
          having instance variables length and breadth. Implement the interface with the
          classess. That is, the classess must define the prototypes declared in the
          interface.
         [Use the formulas,
               a) Circle
                     1. area=3.14*radius*radius
                     2. perimeter=2*3.14*radius
               b) Rectangle
                     1. area=length*breadth
                     2. perimeter=2*(length+breadth)]
Step 3 : Define a menu to display the options (circle and rectangle).
Step 4 : Read the choice of the user.
Step 5 : If the choice is to find the area and perimeter of circle then,
          a) Read the radius and store it.
          b) Invoke the methods of class Circle by instantiating an object of the class.
          c) Display the calculated result.
Step 6  : If the choice is to find the area and perimeter of rectangle then,
          a) Read length and breadth and store it
          b) Invoke the methods of class Circle by instantiating an object of the class.
          c) Display the result.
```

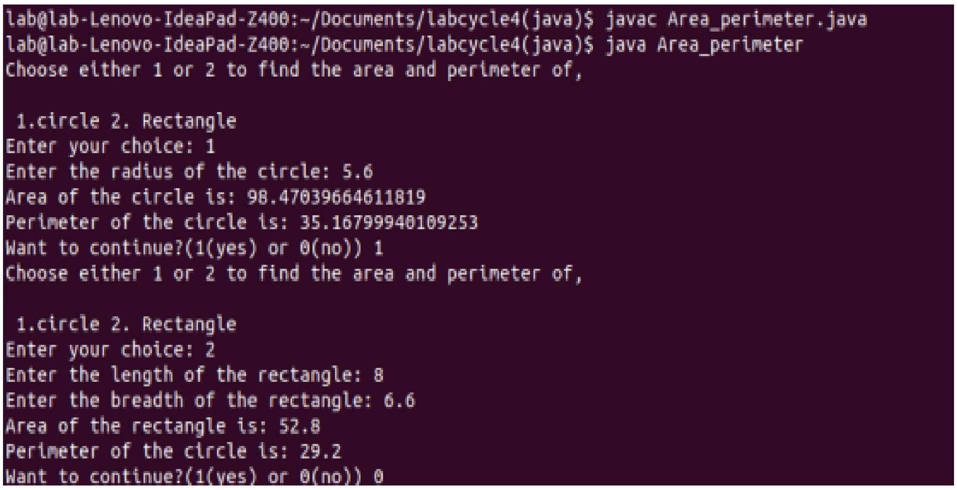
SOURCE CODE :

```
import java.util.*;
interface interface1{
        void area();
        void perimeter();
}
class Circle implements interface1{
        public float radius;
        Circle(float rad){
                radius=rad;
        }
        public void area(){
```

```
        System.out.println("Area of the circle is: " + (3.14*radius*radius));
    }
    public void perimeter(){
        System.out.println("Perimeter of the circle is: " + (2*3.14*radius));
    }
}
class Rectangle implements interface1{
    public float len,bdth;
    Rectangle(Float l, Float b){
        len=l;
        bdth=b;
    }
    public void area(){
        System.out.println("Area of the rectangle is: " + (len*bdth));
    }
    public void perimeter(){
        System.out.println("Perimeter of the circle is: " + (2*(len+bdth)));
    }
}
class Area_perimeter{
    public static void main(String[]args){
        int ch=1;
        while(ch == 1){
            Scanner sc=new Scanner(System.in);
            System.out.println("Choose either 1 or 2 to find the area and
            perimeter of,");
            System.out.println("\n 1.circle 2. Rectangle ");
            System.out.print("Enter your choice: ");
            int choice=sc.nextInt();
            switch(choice){
                case 1: System.out.print("Enter the radius of the circle: ");
                    float r=sc.nextFloat();
                    Circle obj1=new Circle(r);
                    obj1.area();
                    obj1.perimeter();
                    break;
                case 2: System.out.print("Enter the length of the rectangle:
                        ");
                    float l=sc.nextFloat();
                    System.out.print("Enter the breadth of the
                        rectangle: ");
                    float b=sc.nextFloat();
                    Rectangle obj2=new Rectangle(l,b);
                    obj2.area();
                    obj2.perimeter();
                    break;
```

```
                        }
                        System.out.print("Want to continue?(1(yes) or 0(no)) ");
                        ch=sc.nextInt();
                }
        }

}
```

OUTPUT :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Area_perimeter.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Area_perimeter
Choose either 1 or 2 to find the area and perimeter of,

 1.circle 2. Rectangle
Enter your choice: 1
Enter the radius of the circle: 5.6
Area of the circle is: 98.47039664611819
Perimeter of the circle is: 35.16799940109253
Want to continue?(1(yes) or 0(no)) 1
Choose either 1 or 2 to find the area and perimeter of,

 1.circle 2. Rectangle
Enter your choice: 2
Enter the length of the rectangle: 8
Enter the breadth of the rectangle: 6.6
Area of the rectangle is: 52.8
Perimeter of the circle is: 29.2
Want to continue?(1(yes) or 0(no)) 0
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM 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
```

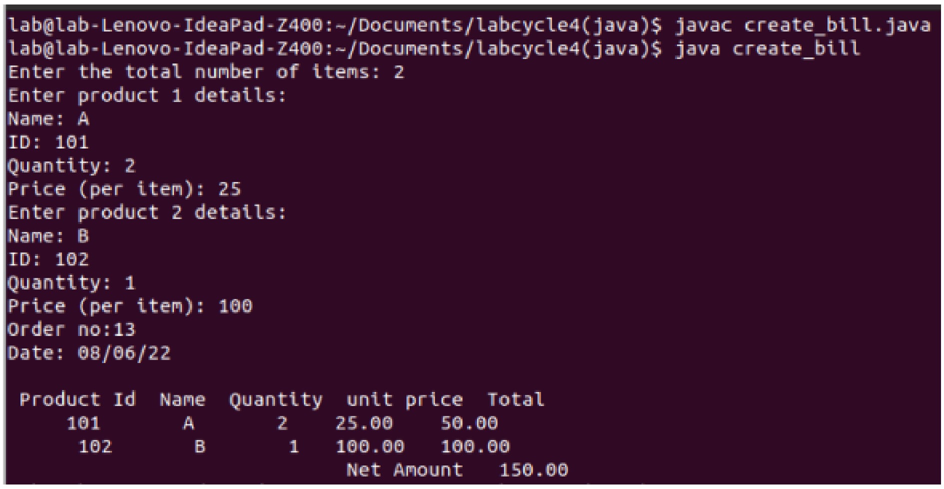
ALGORITHM :

Step 1 : Create an interface which contains a prototype of function method().
Step 2 : Define a class Product which implemets the interface. Also,
  a) define the instance variables to store the values product ID, product name, quantity, unit price and total price.
  b) define the function method () inorder to display the details of products in a bill format. [Use format() of String class to format the output.].
Step 3 : Define an array of object to store the details of all the products.
Step 4 : Invoke the methods defined in the class Product to generate the bill.

SOURCE CODE :

```
import java.util.*;
import java.text.SimpleDateFormat;

interface bill{
        void method();
}
class Product implements bill{
        String p_name;
        int p_id,p_qty,order_no;
        float p_price,total_price;
        Product(){
        }
        Product(String name,int id,int qty,float price,float total){
                p_name=name;
                p_id=id;
                p_qty=qty;
                p_price=price;
                total_price=total;
        }
        void display(){
                Date date = new Date();
                SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yy");
                String str = formatter.format(date);
                Random rand = new Random();
                int upperbound = 25;
                int int_random = rand.nextInt(upperbound);
                System.out.println("Order no:"+ int_random);
                System.out.println("Date: " + str);
                System.out.println("\n Product Id  Name  Quantity  unit price  Total");
        }
        public void method(){
                System.out.format("%8d  %7s  %7d  %8.2f  %8.2f\n  ",  p_id,  p_name,
                p_qty, p_price, total_price);
        }
}
class create_bill{
        public static void main(String args[]){
                Scanner sc=new Scanner(System.in);
                Scanner s=new Scanner(System.in);
                float net_amt=0,total;
                System.out.print("Enter the total number of items: ");
                int n=sc.nextInt();
                Product [] obj=new Product[n];
                for(int i=0;i<n;i++){
                        System.out.println("Enter product" + " " +(i+1) + " " + "details:");
                        System.out.print("Name: ");
                        String name = s.nextLine();
                        System.out.print("ID: ");
                        int id= sc.nextInt();
                        System.out.print("Quantity: ");
                        int qty = sc.nextInt();
                        System.out.print("Price (per item): ");
                        float price = sc.nextFloat();
                        total = price * qty;
                        net_amt+=total;
                        obj[i]=new Product(name,id,qty,price,total);
                }
                Product prdct=new Product();
                prdct.display();
                for(int i=0;i<n;i++)
                        obj[i].method();
                String temp="Net Amount";
                System.out.format("%38s %8.2f ",temp,net_amt);
                System.out.println();
        }
}
```

OUTPUT :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac create_bill.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java create_bill
Enter the total number of items: 2
Enter product 1 details:
Name: A
ID: 101
Quantity: 2
Price (per item): 25
Enter product 2 details:
Name: B
ID: 102
Quantity: 1
Price (per item): 100
Order no:13
Date: 08/06/22

 Product Id  Name  Quantity  unit price  Total
     101       A       2    25.00    50.00
      102       B       1   100.00   100.00
                             Net Amount   150.00
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM 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.

ALGORITHM :

Step 1 : Start
Step 2 : create a package Arithmetic
Step 3 : create 4 source files with interfaces and classes to implement Arithematic operations Addition,Subtraction,Multiplication,Division
Step 4 : write package name at the top of every source file
Step 5 : Create a main program that imports the package Arithmetic with classes Addition,Subtraction,Multiplication,Division.
Step 6 : read the numbers to do Arithematic operation (Addition, Subtraction, Multiplication, Division) and create an object to invoke the methods defined in the classes of package inorder to do the operations
Step 7 : Stop

SOURCE CODE :

Prgrm6.java

```
import java.util.Scanner;
import Arithmetic.Addition;
import Arithmetic.Subtraction;
import Arithmetic.Multiplication;
import Arithmetic.Division;
public class Prgrm6{
        public static void main(String args[]){
                Scanner sc=new Scanner(System.in);
                int i=1,ch;
                System.out.println("_____Program to demonstrate the basic arithmetic
operations_____");
                System.out.println("Enter two numbers:  ");
                float n1=sc.nextFloat();
                float n2=sc.nextFloat();
                while(i!=0){
                        System.out.println("1.   Addition   \t2.   Subtraction   \t3.
Multiplication \t4. Division");
                        System.out.print("Choose an option: ");
                        ch=sc.nextInt();
                        switch(ch){
                                case 1: Addition obj1=new Addition();
```

```
                    obj1.calculate(n1,n2);
                    break;
                case 2: Subtraction obj2=new Subtraction();
                    obj2.calculate(n1,n2);
                    break;
                case 3: Multiplication obj3=new Multiplication();
                    obj3.calculate(n1,n2);
                    break;
                case 4: Division obj4=new Division();
                    obj4.calculate(n1,n2);
                    break;
            }
            System.out.print("Do you want to continue?(0(no),1(yes)) ");
            i=sc.nextInt();
        }

    }
}
```

Addition.java

```
package Arithmetic;
interface interface1{
    void calculate(float a,float b);
}
public class Addition implements interface1{
    public void calculate(float a,float b){
        System.out.println("Sum of the given numbers is : " + (a+b));
    }
public static void main(String args[]){}
}
```

Subtraction.java

```
package Arithmetic;
interface interface1{
    void calculate(float a,float b);
}
public class Subtraction implements interface1 {
    public void calculate(float a,float b){
        System.out.println("Difference of the given numbers is : " + (a-b));
    }
public static void main(String args[]){}
}
```

Multiplication.java

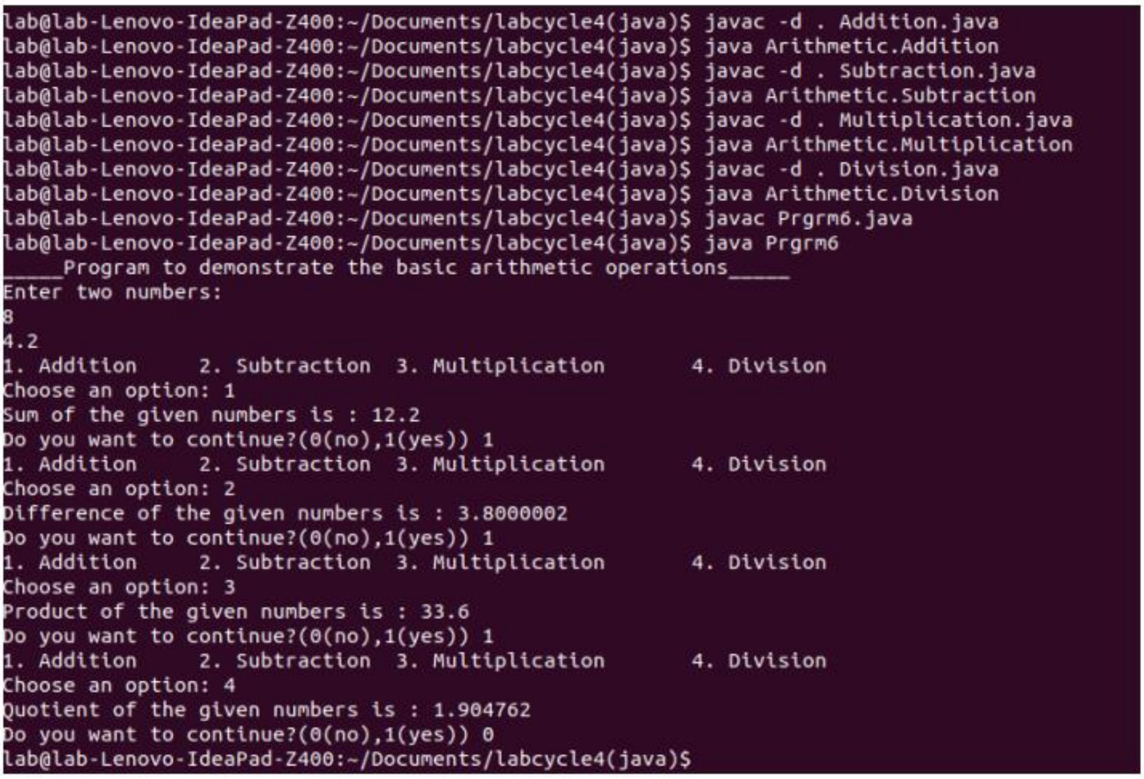
```
package Arithmetic;
interface interface1{
    void calculate(float a,float b);
}
```

```
public class Multiplication implements interface1{
        public void calculate(float a,float b){
                System.out.println("Product of the given numbers is : " + (a*b));
        }
public static void main(String args[]){}
}
```

Division.java

```
package Arithmetic;
interface interface1{
        void calculate(float a,float b);
}
public class Division implements interface1{
        public void calculate(float a,float b){
                System.out.println("Quotient of the given numbers is : " + (a/b));
        }
public static void main(String args[]){}
}
```

OUTPUT :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Addition.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Arithmetic.Addition
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Subtraction.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Arithmetic.Subtraction
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Multiplication.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Arithmetic.Multiplication
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Division.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Arithmetic.Division
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Prgrm6.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Prgrm6
_____Program to demonstrate the basic arithmetic operations_____
Enter two numbers:
8
4.2
1. Addition     2. Subtraction  3. Multiplication       4. Division
Choose an option: 1
Sum of the given numbers is : 12.2
Do you want to continue?(0(no),1(yes)) 1
1. Addition     2. Subtraction  3. Multiplication       4. Division
Choose an option: 2
Difference of the given numbers is : 3.8000002
Do you want to continue?(0(no),1(yes)) 1
1. Addition     2. Subtraction  3. Multiplication       4. Division
Choose an option: 3
Product of the given numbers is : 33.6
Do you want to continue?(0(no),1(yes)) 1
1. Addition     2. Subtraction  3. Multiplication       4. Division
Choose an option: 4
Quotient of the given numbers is : 1.904762
Do you want to continue?(0(no),1(yes)) 0
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :18

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

ALGORITHM :

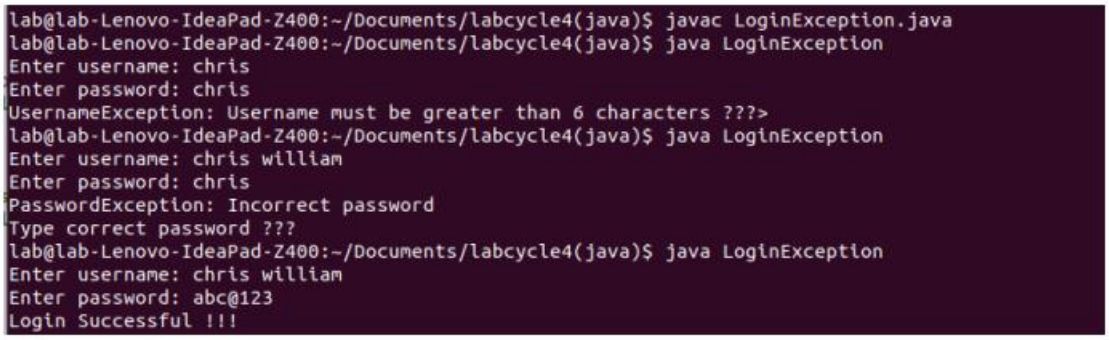
```
Step 1 : Start
Step 2 : Read username and password
Step 3 : Find the length of username
Step 4 : In try block define cases where exceptions may arise
        Step 4.1 : if username length is less than 6
                   call UsernameException which inherits class Exception
                   by passing message “Username must be greater than 6 characters”.
        Step 4.2 : else if password not equal to predifined string
                   call  PasswordException which inherits class Exception
                   by passing message “Incorrect password Type correct password”
        Step 4.3 : else print  “Login Successful !!”
Step 5 : if there is any of the above Userdefined Exception catch the exception and print
        the corresponding message.
Step 6 : Stop
```

SOURCE CODE :

```
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 LoginException{
        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("abc@123")){
                throw new PasswordException("Incorrect password\nType
                correct password ???");}
            else
                System.out.println("Login Successful !!!");
        }
        catch (UsernameException e){
            System.out.println(e);
        }
        catch (PasswordException p){
            System.out.println(p);
        }
    }
}
```

OUTPUT :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac LoginException.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LoginException
Enter username: chris
Enter password: chris
UsernameException: Username must be greater than 6 characters ???>
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LoginException
Enter username: chris william
Enter password: chris
PasswordException: Incorrect password
Type correct password ???
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LoginException
Enter username: chris william
Enter password: abc@123
Login Successful !!!
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :19

AIM : Define two classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

ALGORITHM :

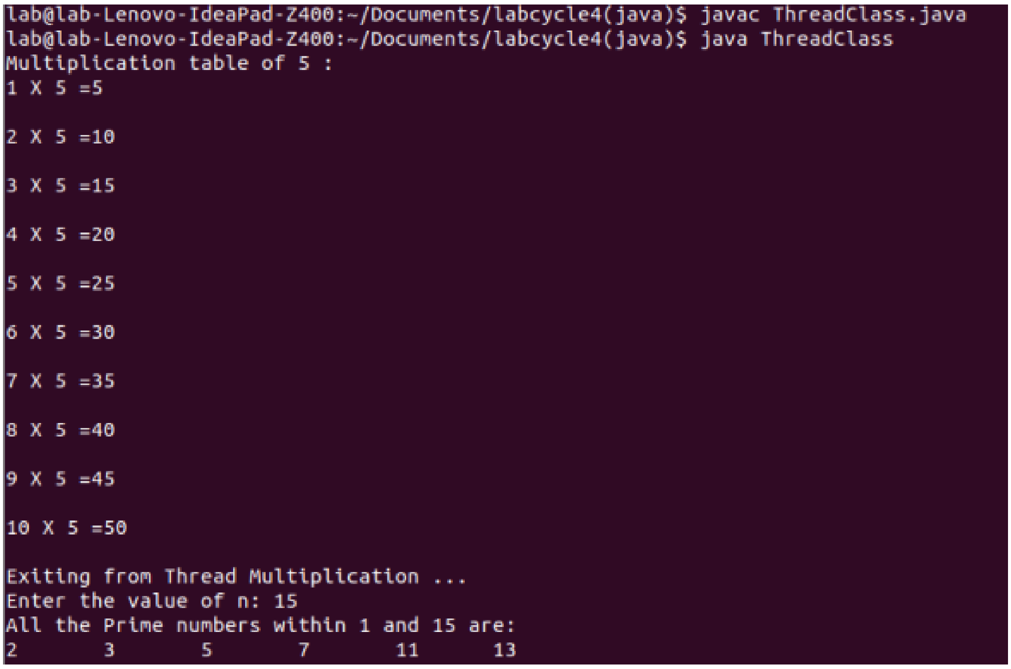
```
Step 1 : Start
Step 2 : create an object a for class multiplication which inherits Thread class
Step 3 : create an object b for class primenumbers which inherits Thread class
Step 4 : call run() in class  multiplication by a.start()
        Step 4.1 : set i=1
        Step 4.2 : if i<=10
                print “i x 5 = i*5”
                i = i+1 go to step 4.2
        Step 4.3 : else exit from multiplication thread
Step 5 : call run() in class  primenumbers by b.start()
        Step 5.1 : Read the limit value n
        Step 5.2 : call prime_N()
                Step 5.2.1 : set x=1
                Step 5.2.2 : if x <= n do step 4.2.3 to 4.2.10
                Step 5.2.3 : check if x=1 or x=0 then i=i+1
                Step 5.2.4 :  set flag=1
                Step 5.2.5 :  set y=2
                Step 5.2.6 : if y <= x/2 do steps 4.2.7 to 4.2.8
                Step 5.2.7 : if x % y = 0 set flag=0 and go to 4.2.9
                Step 5.2.8 : y=y+1 go to step 4.2.6
                Step 5.2.9 : if flag = 1 print x
                Step 5.2.10 : print new line go to step 4.2.2
Step 6 : Stop
```

SOURCE CODE :

```
import java.util.*;
class multiplication extends Thread{
        public void run(){
                System.out.println("Multiplication table of 5 : ");
                for(int i = 1; i <= 10; i++){
                        System.out.println(i + " X 5 =" + i*5 +"\n");}
                System.out.println("Exiting from Thread Multiplication ...");
        }
}
```

```
class primenumbers extends Thread{
        public void run(){
                Scanner sc=new Scanner(System.in);
                System.out.print("Enter the value of n: ");
                int n=sc.nextInt();
                prime_N(n);
        }
        static void prime_N(int N){
                int x,y,flag;
                System.out.println("All the Prime numbers within 1 and " + N + " are:");
                for (x = 1; x <= N; x++){
                if (x == 1 || x == 0)
                                continue;
                flag = 1;
                for (y = 2; y <= x / 2; ++y){
                                if (x % y == 0) {
                                flag = 0;
                                break;
                                }
                }
                if (flag == 1)
                                System.out.print( x + "\t");
                }
                System.out.println();
        }
}
public class ThreadClass{
        public static void main(String args[]){
                multiplication a = new multiplication();
                primenumbers b = new primenumbers();
                a.start();
                b.start();
        }
}
```

OUTPUT :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac ThreadClass.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java ThreadClass
Multiplication table of 5 :
1 X 5 =5

2 X 5 =10

3 X 5 =15

4 X 5 =20

5 X 5 =25

6 X 5 =30

7 X 5 =35

8 X 5 =40

9 X 5 =45

10 X 5 =50

Exiting from Thread Multiplication ...
Enter the value of n: 15
All the Prime numbers within 1 and 15 are:
2       3       5       7       11      13
```

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :20

AIM: program to draw circle, Rectangle line in an applet.

ALGORITHM:

Step 1: Import the necessary Java libraries for AWT (Abstract Window Toolkit) and Applet.

Step 2: Define a class named "CircleSquare" that extends the "Applet" class.

Step 3: Override the "paint" method, which is called by the system to render graphics on the applet's canvas.

Step 4: Inside the "paint" method:

a. Set the drawing color to red using `g.setColor(Color.red);

b. Set the drawing color to blue using `g.setColor(Color.blue);

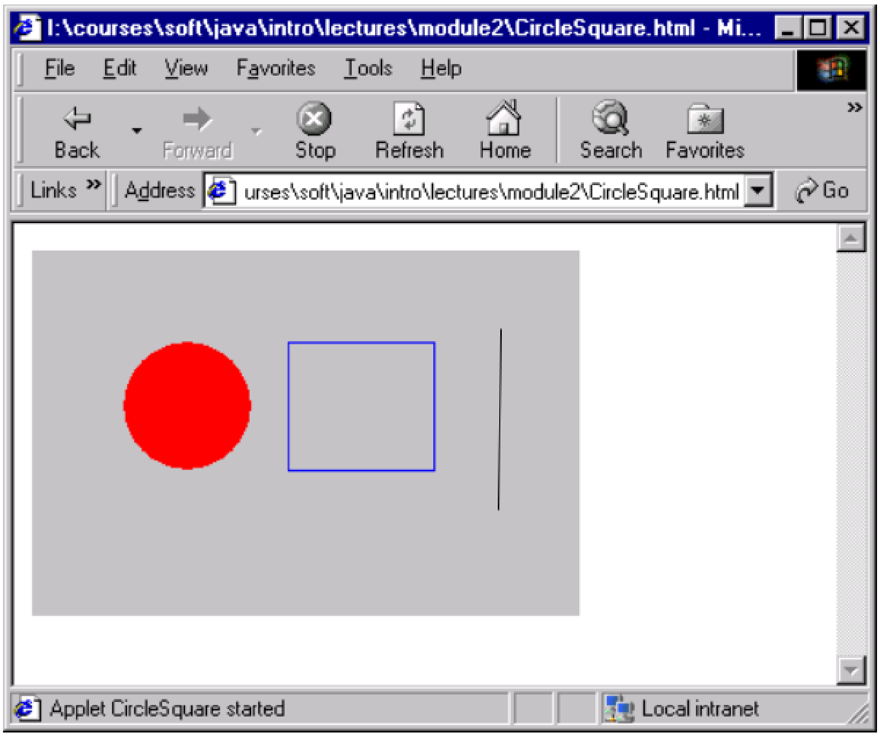
c. Set the drawing color to green using `g.setColor(Color.green);

Step 5: The applet will automatically call the "paint" method and execute the program.

SOURCE CODE.

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

public class CircleSquare extends Applet {

  public void paint (Graphics g)
  {
    // Draw a red, filled circle:
    g.setColor (Color.red);
    g.fillOval (50,50,70,70);

    // Draw an unfilled blue square next to it:
    g.setColor (Color.blue);
    g.drawRect (140,50,80,70);

   // Now we tell g to change the color
    g.setColor(Color.green);
    g.drawLine(0, 0, 100, 100);

  }

}
```

OUTPUT:


I:\courses\soft\java\intro\lectures\module2\CircleSquare.html - Mi...
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites
Links Address urses\soft\java\intro\lectures\module2\CircleSquare.html Go
Applet CircleSquare started
Local intranet

RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :21

AIM: Program to handle all mouse events.

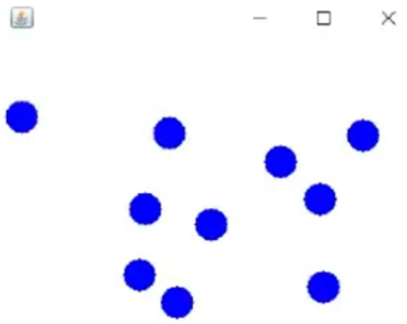
ALGORITHM:

```
        Step 1: Import necessary libraries and create a class.
        Step 2: Create a custom frame class that extends Frame or JFrame.
        Step 3: Implement event handling logic.
        Step 4: Implement your event handling logic.
        Step 5: Add components and run the application.
```

SOURCE CODE:

```
import java.awt.*;
import java.awt.event.*;
public class mousexamp12 extends Frame implements MouseListener
{
mousexamp12()
{
addMouseListener(this);
setSize(400,400);
setLayout(null);
setVisible(true);
}
public void mouseClicked(MouseEvent e)
{
Graphics g=getGraphics();
g.setColor(Color.blue);
g.fillOval(e.getX(),e.getY(),30,30);
}
public void mouseEntered(MouseEvent e)
{
}
public void mouseExited(MouseEvent e)
{
}
public void mousePressed(MouseEvent e)
{
}
public void mouseReleased(MouseEvent e)
{

}
public static void main(String args[])
{
new mousexamp12();
}
}
```

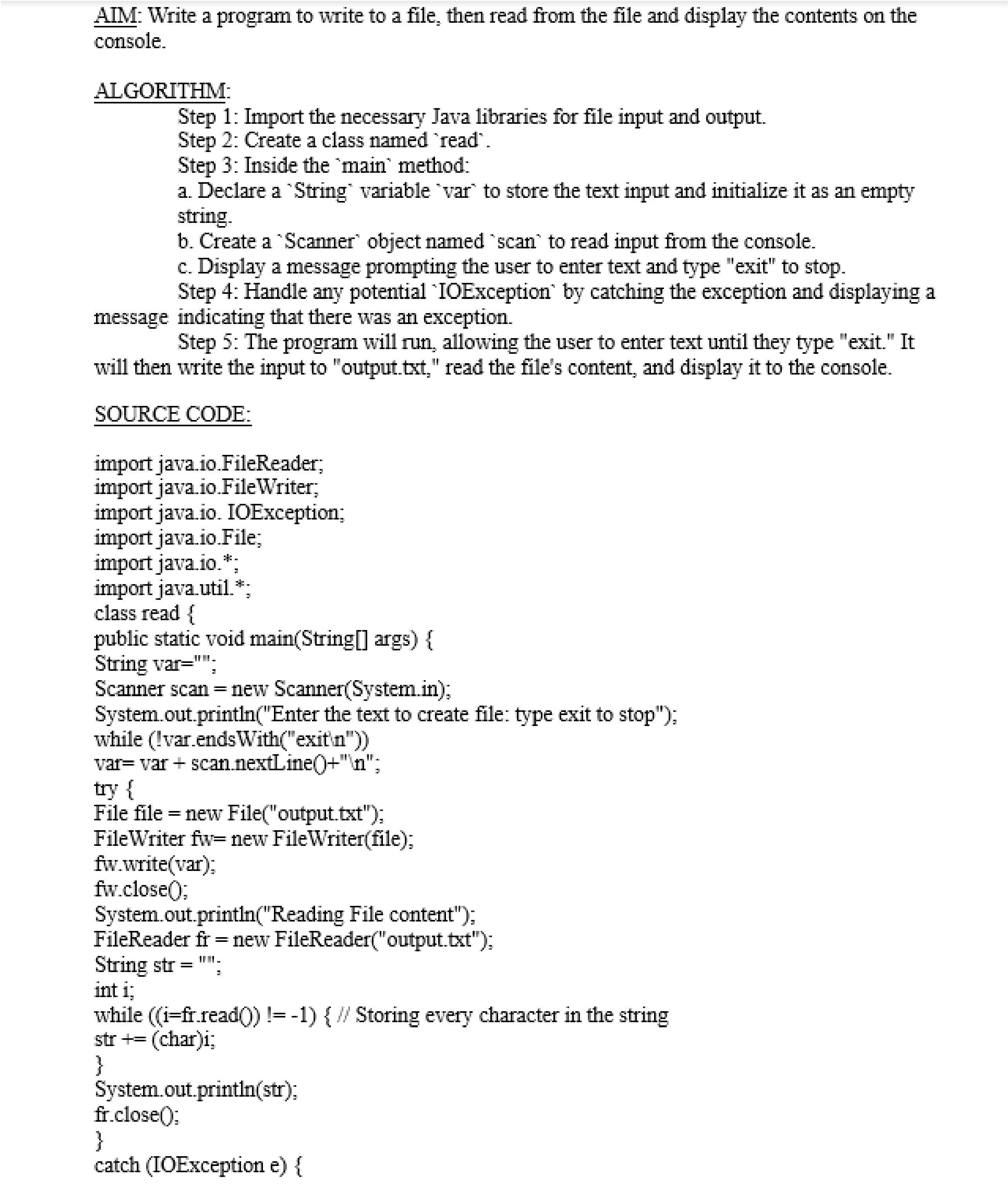
OUTPUT:



RESULT :

Program is successfully executed and output is verified.

PROGRAM NO :22



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

ALGORITHM:

Step 1: Import the necessary Java libraries for file input and output.

Step 2: Create a class named `read`.

Step 3: Inside the `main` method:

a. Declare a `String` variable `var` to store the text input and initialize it as an empty string.

b. Create a `Scanner` object named `scan` to read input from the console.

c. Display a message prompting the user to enter text and type "exit" to stop.

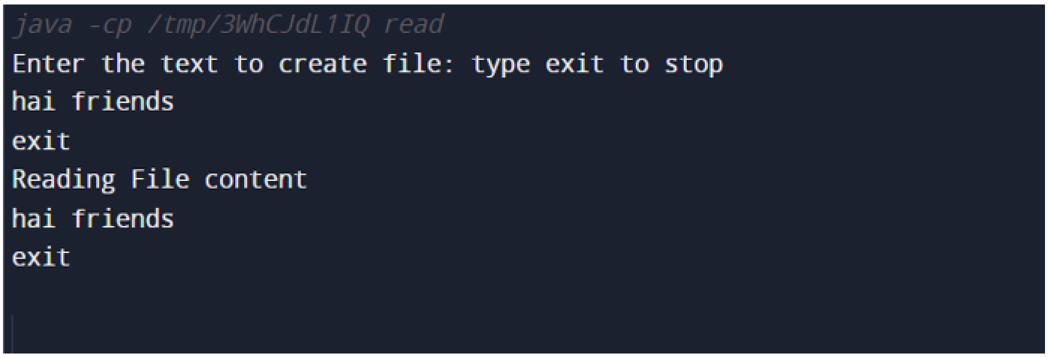
Step 4: Handle any potential `IOException` by catching the exception and displaying a message indicating that there was an exception.

Step 5: The program will run, allowing the user to enter text until they type "exit." It will then write the input to "output.txt," read the file's content, and display it to the console.

SOURCE CODE:

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io. IOException;
import java.io.File;
import java.io.*;
import java.util.*;
class read {
public static void main(String[] args) {
String var="";
Scanner scan = new Scanner(System.in);
System.out.println("Enter the text to create file: type exit to stop");
while (!var.endsWith("exit\n"))
var= var + scan.nextLine()+"\n";
try {
File file = new File("output.txt");
FileWriter fw= new FileWriter(file);
fw.write(var);
fw.close();
System.out.println("Reading File content");
FileReader fr = new FileReader("output.txt");
String str = "";
int i;
while ((i=fr.read()) != -1) { // Storing every character in the string
str += (char)i;
}
System.out.println(str);
fr.close();
}
catch (IOException e) {
    System.out.println("There are some exception");
}
}
}
```

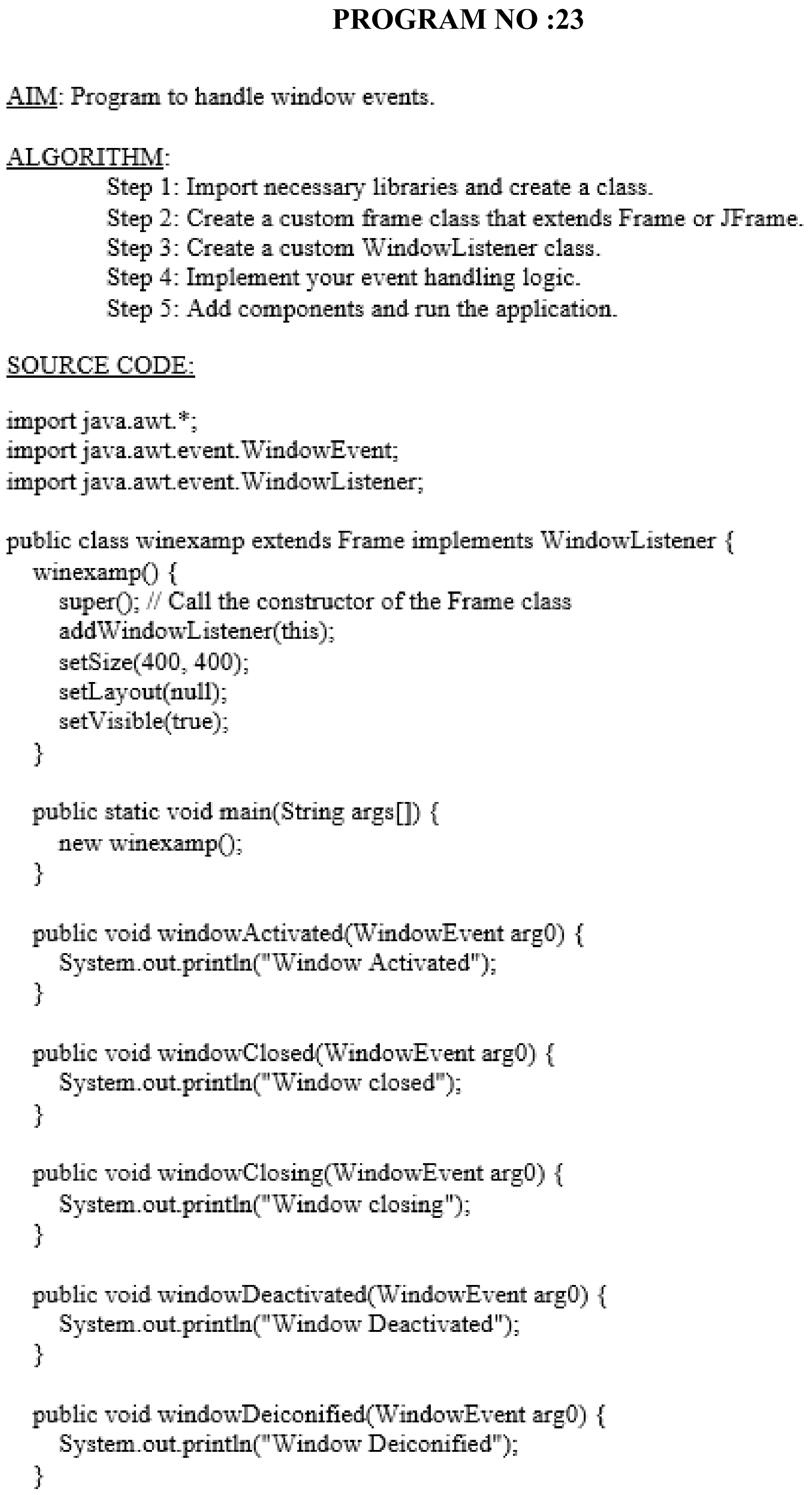
OUTPUT:



```
java -cp /tmp/3WhCJdL1IQ read
Enter the text to create file: type exit to stop
hai friends
exit
Reading File content
hai friends
exit
```

RESULT :

Program is successfully executed and output is verified.



PROGRAM NO :23

AIM: Program to handle window events.

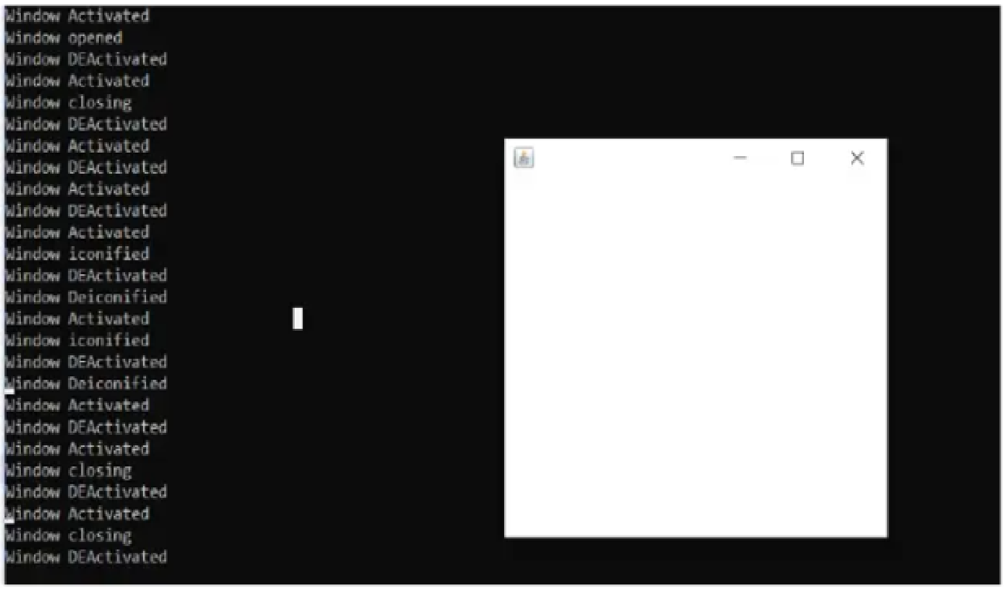
ALGORITHM:

```
        Step 1: Import necessary libraries and create a class.
        Step 2: Create a custom frame class that extends Frame or JFrame.
        Step 3: Create a custom WindowListener class.
        Step 4: Implement your event handling logic.
        Step 5: Add components and run the application.
```

SOURCE CODE:

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

public class winexamp extends Frame implements WindowListener {
    winexamp() {
        super(); // Call the constructor of the Frame class
        addWindowListener(this);
        setSize(400, 400);
        setLayout(null);
        setVisible(true);
    }

    public static void main(String args[]) {
        new winexamp();
    }

    public void windowActivated(WindowEvent arg0) {
        System.out.println("Window Activated");
    }

    public void windowClosed(WindowEvent arg0) {
        System.out.println("Window closed");
    }

    public void windowClosing(WindowEvent arg0) {
        System.out.println("Window closing");
    }

    public void windowDeactivated(WindowEvent arg0) {
        System.out.println("Window Deactivated");
    }

    public void windowDeiconified(WindowEvent arg0) {
        System.out.println("Window Deiconified");
    }
    public void windowIconified(WindowEvent arg0) {
        System.out.println("Window iconified");
    }

    public void windowOpened(WindowEvent arg0) {
        System.out.println("Window opened");
    }
}
```

OUTPUT:



```
Window Activated
Window opened
Window DEActivated
Window Activated
Window closing
Window DEActivated
Window Activated
Window DEActivated
Window Activated
Window DEActivated
Window Activated
Window iconified
Window DEActivated
Window Deiconified
Window Activated
Window iconified
Window DEActivated
Window Deiconified
Window Activated
Window DEActivated
Window Activated
Window closing
Window DEActivated
Window Activated
Window closing
Window DEActivated
```

RESULT :

Program is successfully executed and output is verified.