DEPARTMENT OF COMPUTER APPLICATIONS

TKM COLLEGE OF ENGINEERING

KOLLAM-691005

20MCA132 -OBJECT ORIENTED PROGRAMMING LAB

PRACTICAL RECORD BOOK

Second Semester MCA

2022

SUBMITTED BY

NAME: SIVAMURUGAN M

REGISTER NUMBER: TKM21MCA-2033

DEPARTMENT OF COMPUTER APPLICATIONS

TKM COLLEGE OF ENGINEERING

KOLLAM-691005

Certificate

This is a bonafide record of the work done by **SIVAMURUGAN M** (TKM21MCA-2033) in the Second Semester in OBJECT ORIENTED PROGRAMMING LAB Course(20MCA132) towards the partial fulfilment of the degree of Master of Computer Applications during the academic year 2022.

Staff Member in-charge Examiner

…… ……

INDEX

COURSE OUTCOME 1

PROGRAM 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.

ALGORITHM:

Step1-Create a class “product”
Step2-Initialize product code, name, price.
Step3-Write function to display product details
Step4-Write function to find the lowest price among products

PROGRAM CODE:

product.java	import java.util.*;
	public class product
	{
	int pcode;
	String pname;
	int price;
	private Scanner r;
	public void get()
	{
	r = new Scanner(System.in);
	System.out.print("Enter the Product id: ");
	this.pcode= r.nextInt();
	System.out.print("Enter the Product name: ");
	this.pname= r.next();
	System.out.print("Enter the Product Price: ");
	this.price= r.nextInt();
	}
	public static void main(String[] args)
	{
	product item1 = new product();
	item1.get();
	System.out.println("Product Code: "+item1.pcode);

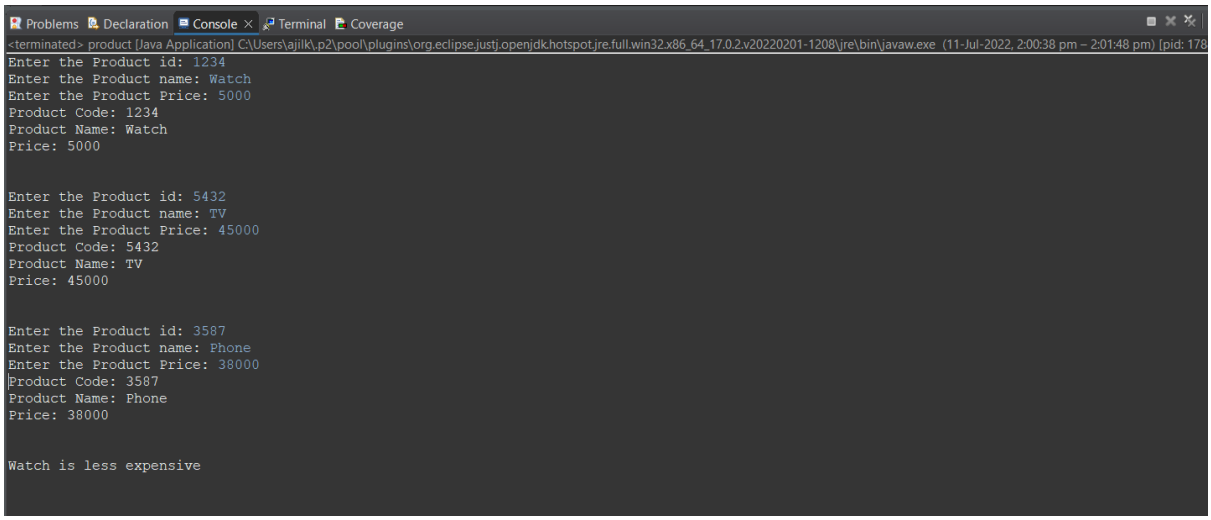
```
System.out.println("Product Name: "+item1.pname );
System.out.println("Price: "+item1.price);
System.out.println("\n");
product item2 = new product();
item2.get();
System.out.println("Product Code: "+item2.pcode);
System.out.println("Product Name: "+item2.pname );
System.out.println("Price: "+item2.price);
System.out.println("\n");
product item3 = new product();
item3.get();
System.out.println("Product Code: "+item3.pcode);
System.out.println("Product Name: "+item3.pname );
System.out.println("Price: "+item3.price);
System.out.println("\n");
if(item1.price<item2.price)
{
    if(item1.price<item3.price)
    {
        System.out.println(item1.pname+" is less expensive");
    }
    else
    {
        System.out.println(item3.pname+" is less expensive");
    }
}
else
{
    if(item2.price<item3.price)
    {
        System.out.println(item2.pname+" is less expensive");
    }
    else
```

```
        {
                System.out.println(item3.pname+" is less expensive");
        }
    }
  }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 2

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

ALGORITHM:
Step 1: Create a class matrix
Step 2: Initialize the variables
Step 3: Input the rows and columns from user
Step 4: Input elements in the 2 matrices
Step 5: Find the sum of 2 matrices

PROGRAM CODE:

matrix.java

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

    private static Scanner read;

    public static void main(String[] args)
    {
        int[][] a=new int[10][10];
        int[][] b=new int[10][10];
        int i,j;
        read = new Scanner(System.in);
        System.out.println("Enter the number of rows:");
        int r = read.nextInt();
        System.out.println("Enter the number of columns:");
        int c = read.nextInt();
        System.out.println("Enter the elements in 1st matrix\n");
        for(i=0;i<r;i++)
        {
            for(j=0;j<c;j++)
            {
                a[i][j]=read.nextInt();
            }
        }
```

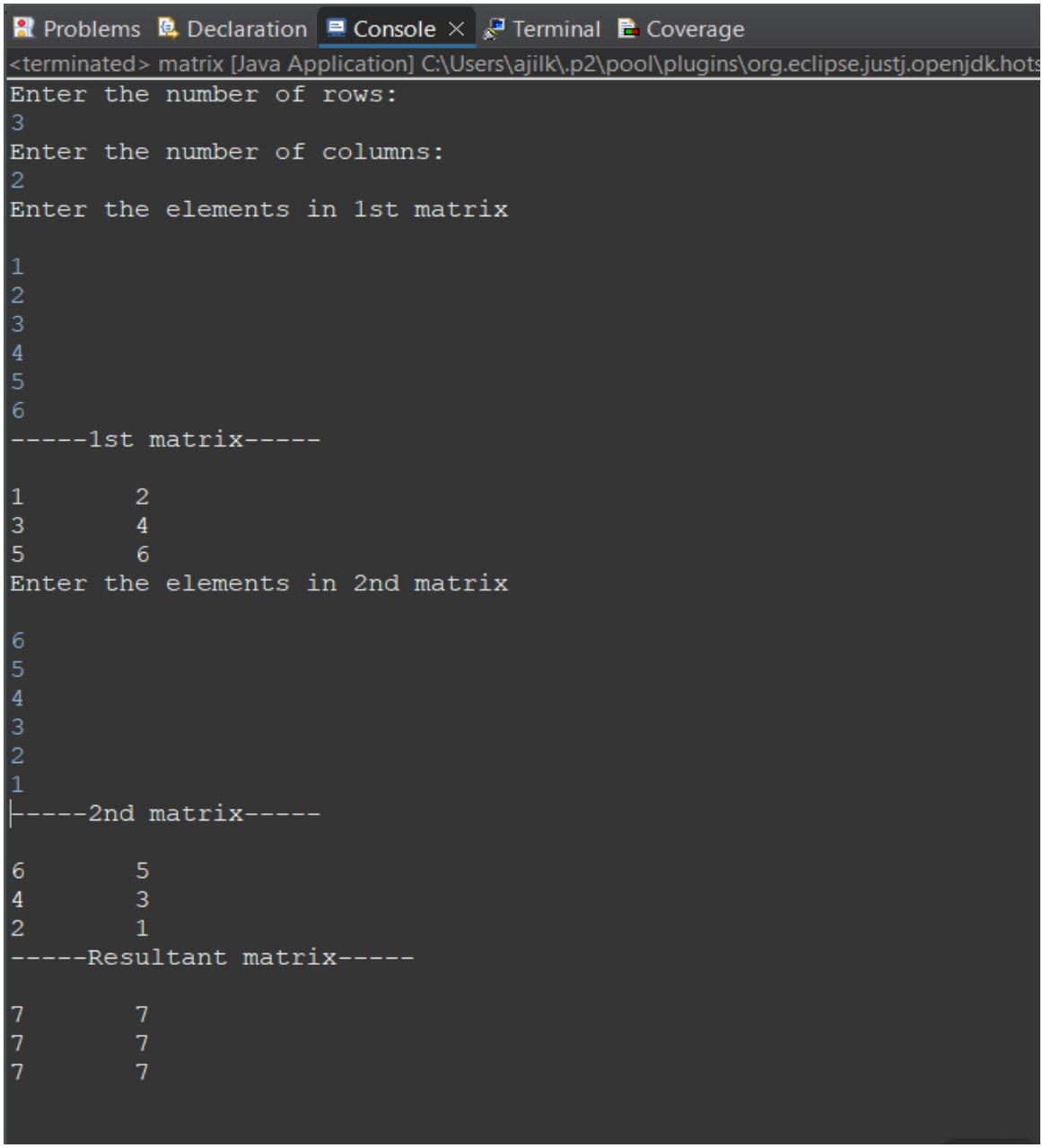
```
System.out.println("-----1st matrix-----\n");
for(i=0;i<r;i++)
{
        for(j=0;j<c;j++)
        {
                System.out.print(a[i][j]+"\t");
        }
        System.out.print("\n");
}
System.out.println("Enter the elements in 2nd matrix\n");
for(i=0;i<r;i++)
{
        for(j=0;j<c;j++)
        {
                b[i][j]=read.nextInt();
        }
}
System.out.println("-----2nd matrix-----\n");
for(i=0;i<r;i++)
{
        for(j=0;j<c;j++)
        {
                System.out.print(b[i][j]+"\t");
        }
        System.out.print("\n");
}
System.out.println("-----Resultant matrix-----\n");
for(i=0;i<r;i++)
{
        for(j=0;j<c;j++)
        {
                System.out.print(((a[i][j])+(b[i][j]))+"\t");
        }
```

```
                    System.out.print("\n");
                }
            }

}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 3

AIM:
Add complex numbers.

ALGORITHM:
Step 1: Create a class complexNo
Step 2: Initialize the variables and objects
Step 3: Input the real part and imaginary part value from user
Step 4: Add the complex numbers
Step 5: Display the result

PROGRAM CODE:

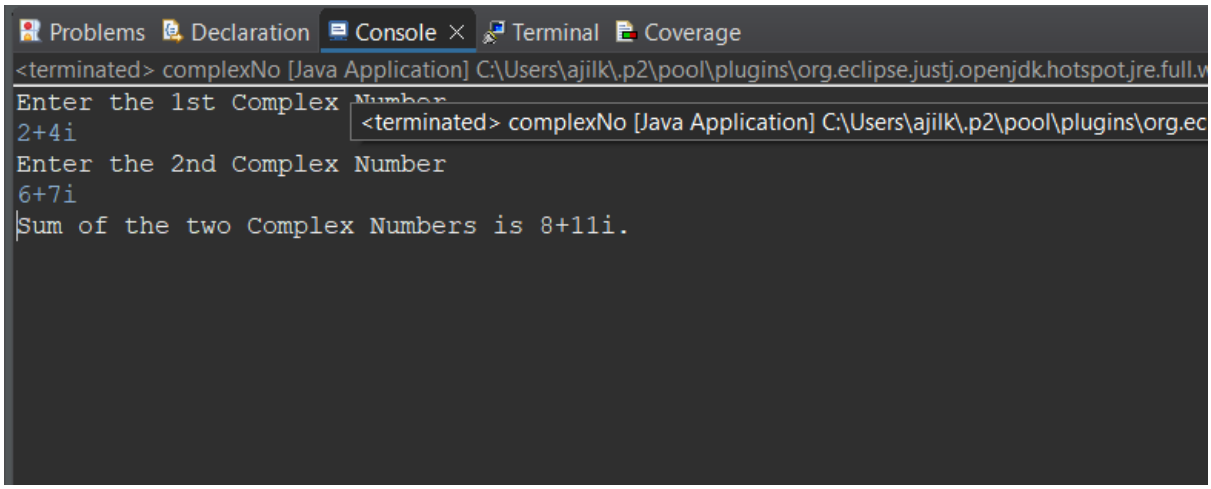
complexNo.java

```
import java.util.Scanner;
public class complexNo
{
        private static Scanner read;

        public static void main(String[] args)
        {
                read = new Scanner(System.in);
                System.out.println("Enter the 1st Complex Number");
                String c1=read.nextLine();
                String[] n1=c1.split("\\+");
                int r1 = Integer.parseInt(n1[0]);
                int  i1=Integer.parseInt(n1[1].substring(0,(n1[1].length()-
1)));
                System.out.println("Enter the 2nd Complex Number");
                String c2=read.nextLine();
                String[] n2=c2.split("\\+");
                int r2 = Integer.parseInt(n2[0]);
                int  i2=Integer.parseInt(n2[1].substring(0,(n2[1].length()-
1)));
                System.out.println("Sum of the two Complex Numbers
is "+(r1+r2)+"+"+(i1+i2)+"i.");
        }}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 4

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

ALGORITHM:
Step 1: Create a class symmetricMatrix
Step 2: Initialize the variables
Step 3: Input the rows and columns from user
Step 4: Input elements in the 2 matrices
Step 5: Check if the elements in the given matrix and transposed matrix is same or not i.e., symmetric, or not.

PROGRAM CODE:

symmetricMatrix.java

```
import java.util.*;
public class symmetricMatrix
{
	private static Scanner read;
	public static void main(String[] args)
	{
		read = new Scanner(System.in);
		int[][] a= new int [10][10];
		int r,c,i,j;
		System.out.println("Enter the number of rows:");
		r= read.nextInt();
		System.out.println("Enter the number of columns:");
		c= read.nextInt();
		if(r==c)
		{
			System.out.println("Enter the elements in the
matrix:");
			for(i=0;i<r;i++)
			{
				for(j=0;j<c;j++)
				{
					a[i][j]=read.nextInt();
```

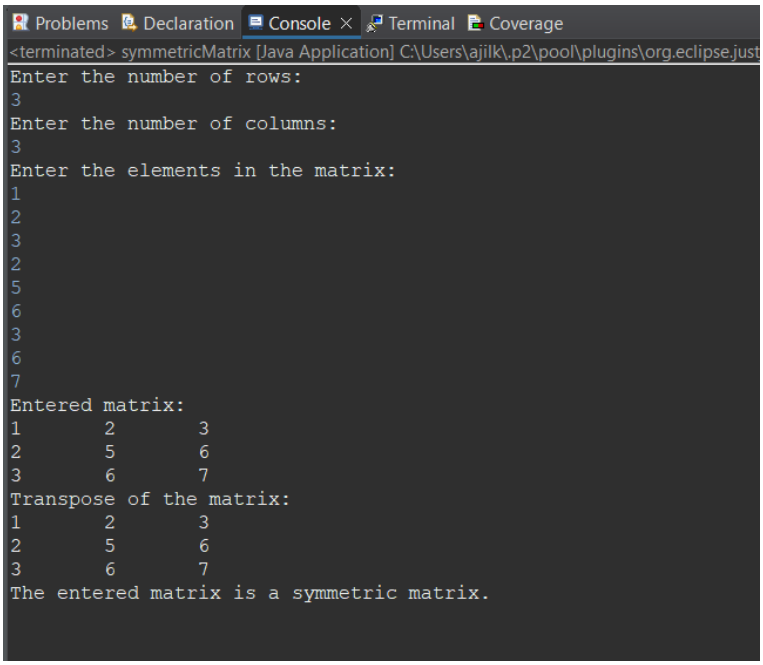
```
                }
        }
        System.out.println("Entered matrix:");
        for(i=0;i<r;i++)
        {
                for(j=0;j<c;j++)
                {
                        System.out.print(a[i][j]+"\t");
                }
                System.out.print("\n");
        }
        System.out.println("Transpose of the matrix:");
        for(i=0;i<r;i++)
        {
                for(j=0;j<c;j++)
                {
                        System.out.print(a[j][i]+"\t");
                }
                System.out.print("\n");
        }
        int flag=0;
        for(i=0;i<r;i++)
        {
                for(j=0;j<c;j++)
                {
                        if(a[i][j]!=a[j][i])
                        {
                                flag=1;
                        }
                }
        }
        if(flag==0)
```

```
                    {
                        System.out.println("The entered matrix is a symmetric matrix.");
                    }
                    else
                    {
                        System.out.println("The entered matrix is not a symmetric matrix.");
                    }
                }
                else
                {
                    System.out.println("Only square matrix can be used.");
                }
            }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM 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..

ALGORITHM:

Step 1: Create a class CPU with attribute price
Step 2: Create inner class processor with number of cores, manufacture attributes
Step 3: Create a static nested class RAM with memory and company attributes
Step 4: Ask to enter the values
Step 5: Display the details

PROGRAM CODE:

computer.java

```
import java.util.Scanner;
class cpu
{
	int price;
	cpu(int cost)
	{
		price=cost;;
	}
	void display()
	{
		System.out.println("Price- "+price);
	}
	class processor
	{
		int core;
		String manufacturer;
		processor(int x,String s)
		{
			core=x;
			manufacturer=s;
		}
		void display()
```

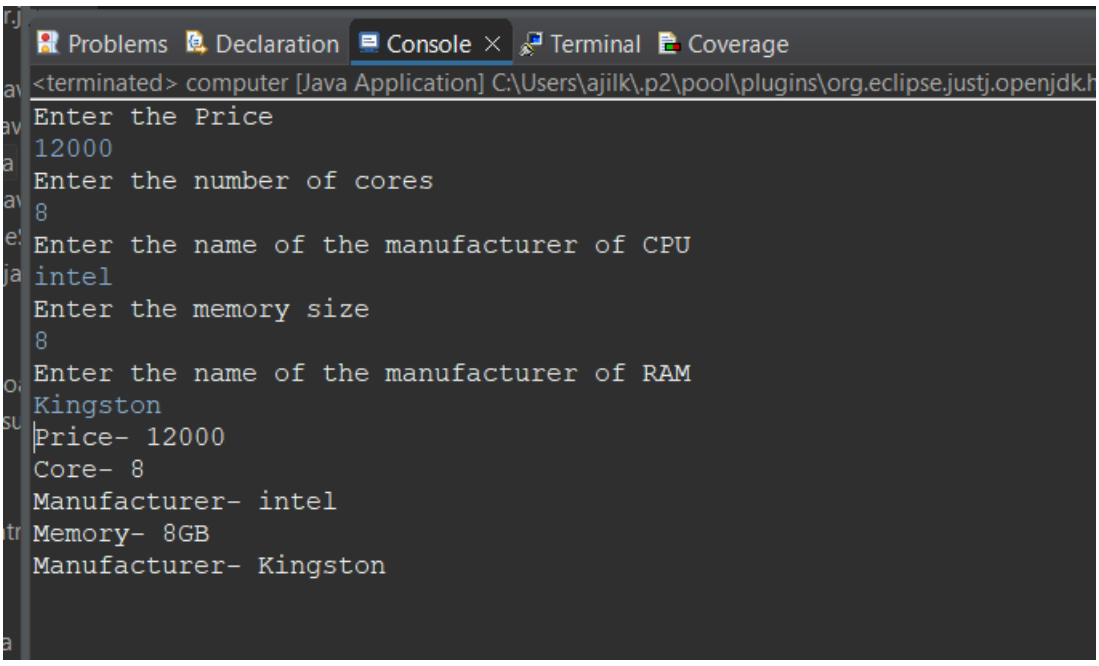
```
                {
                        System.out.println("Core-
"+core+"\n"+"Manufacturer- "+manufacturer);
                }
        }
        static class ram
        {
                int memory;
                String manufacturer;
                ram(int x,String s)
                {
                        memory=x;
                        manufacturer=s;
                }
                void display()
                {
                        System.out.println("Memory-
"+memory+"GB\n"+"Manufacturer- "+manufacturer);
                }
        }
}
public class computer
{

        public static void main(String[] args)
        {
                Scanner read = new Scanner(System.in);
                System.out.println("Enter the Price ");
                int pPrice=read.nextInt();
                cpu c=new cpu(pPrice);
                System.out.println("Enter the number of cores ");
                int pCore=read.nextInt();
                System.out.println("Enter the name of the manufacturer
```

```
of CPU");
                String pMan=read.next();
                System.out.println("Enter the memory size ");
                int rMemory=read.nextInt();
                System.out.println("Enter the name of the manufacturer
of RAM");
                String rMan=read.next();
                cpu.processor p= c.new processor(pCore,pMan);
                cpu.ram r=new cpu.ram(rMemory, rMan);
                c.display();
                p.display();
                r.display();
                System.out.println();

        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



COURSE OUTCOME 2

PROGRAM NO: 6

AIM:
Program to Sort strings

ALGORITHM:
Step 1: Create a class sortString
Step 2: Input the number strings from user
Step 3: Enter the string
Step 4: Sort the given strings

PROGRAM CODE:

sortString.java

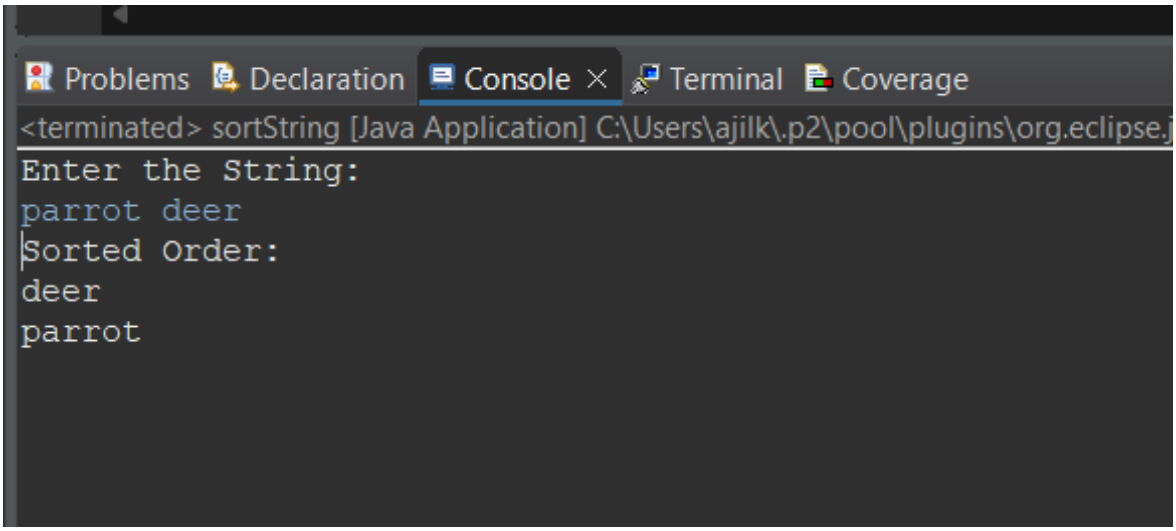
```
import java.util.*;
public class sortString
{
	public static void main(String[] args)
	{
		String s, str;
		String array[] = new String[10];
		Scanner r = new Scanner(System.in);
		System.out.println("Enter the String:");
		s=r.nextLine();
		array = s.split(" ");
		for(int i=0;i<array.length;i++)
		{
			for(int j=i+1;j<array.length;j++)
			{
				if(array[i].compareTo(array[j])>0)
				{
					str = array[i];
					array[i] = array[j];
					array[j] = str;
				}
			}
```

```
                    }
                    System.out.println("Sorted Order: ");
                    for (int i=0;i<array.length;i++)
                    {
                            System.out.println(array[i]);
                    }
                    r.close();
            }
}   }
 }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 7

AIM:
Search an element in an array.

ALGORITHM:
Step 1: Create a class searchArray
Step 2: Enter array of elements
Step 3: Search for an element
Step 4: Find the element with its index number

PROGRAM CODE:

searchArray.java

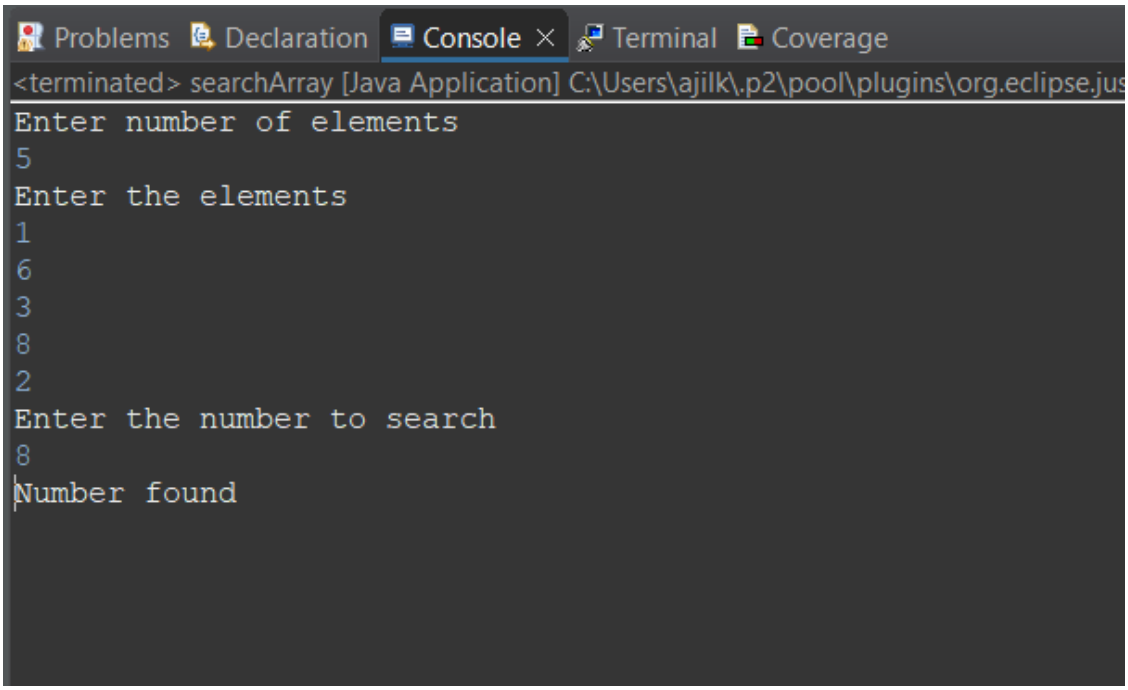
```
package CO2;
import java.util.*;
public class searchArray
{
	public static void main(String [] args)
	{
		int a[]=new int[10];
		int flag=0,n,i,key;
		Scanner r=new Scanner(System.in);
		System.out.println("Enter number of elements");
		n=r.nextInt();
		System.out.println("Enter the elements");
		for (i=0;i<n;i++)
		{
			a[i]=r.nextInt();
		}
		System.out.println("Enter the number to search");
		key=r.nextInt();
		for (i=0;i<n;i++)
		{
			if(a[i]==key)
			{
				System.out.println("Number found");
				flag=1;
```

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

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 8

AIM:
Perform string manipulations.

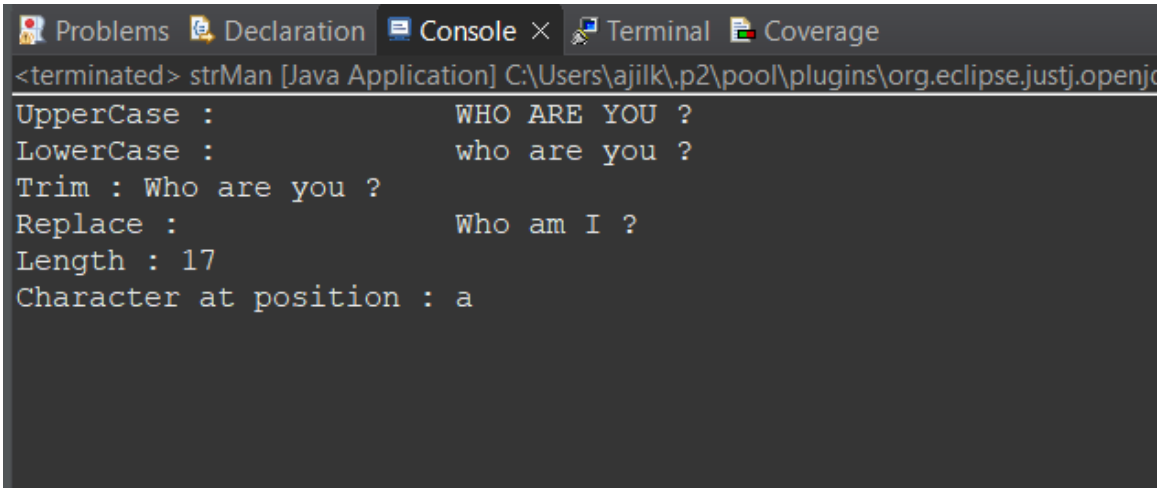
ALGORITHM:
Step 1: Create a class strMan
Step 2: Enter the strings
Step 3: Perform string operations
Step 4: Display the corresponding result.

PROGRAM CODE:

strMan.java	package CO2;
	public class strMan
	{
	public static void main(String [] args)
	{
	String s=" Who are you ? ";
	System.out.println("UpperCase : "+s.toUpperCase());
	System.out.println("LowerCase : "+s.toLowerCase());
	System.out.println("Trim : "+s.trim());
	System.out.println("Replace : "+s.replace("are you","am I"));
	System.out.println("Length : "+s.length());
	System.out.println("Character at position : "+s.charAt(6));
	}
	}

RESULT:
The above program is successfully executed and obtained the output.

OUTPUT:


Problems
Declaration
Console
Terminal
Coverage
<terminated> strMan [Java Application] C:\Users\ajilk\.p2\pool\plugins\org.eclipse.justj.openj
UpperCase : WHO ARE YOU ?
LowerCase : who are you ?
Trim : Who are you ?
Replace : Who am I ?
Length : 17
Character at position : a

PROGRAM 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.

ALGORITHM:

Step 1: Create a class named employee
Step 2: Create attributes eNo, eName, eSalary
Step 3: Input n employee details
Step 4: Search and find employee using the eNo.

PROGRAM CODE:

employee.java

```
import java.util.*;
class employees
{
        int empNo;
        String empName;
        Double empSalary;
        employees(int a,String b,Double c)
        {
                empNo=a;
                empName=b;
                empSalary=c;
        }
}
public class employee
{
        public static void main(String[] args)
        {
                int i=0,flag=0,id,key;
                String name;
                Double salary;
                employees obj[]=new employees[10];
                System.out.println("Enter the number of Employees");
```

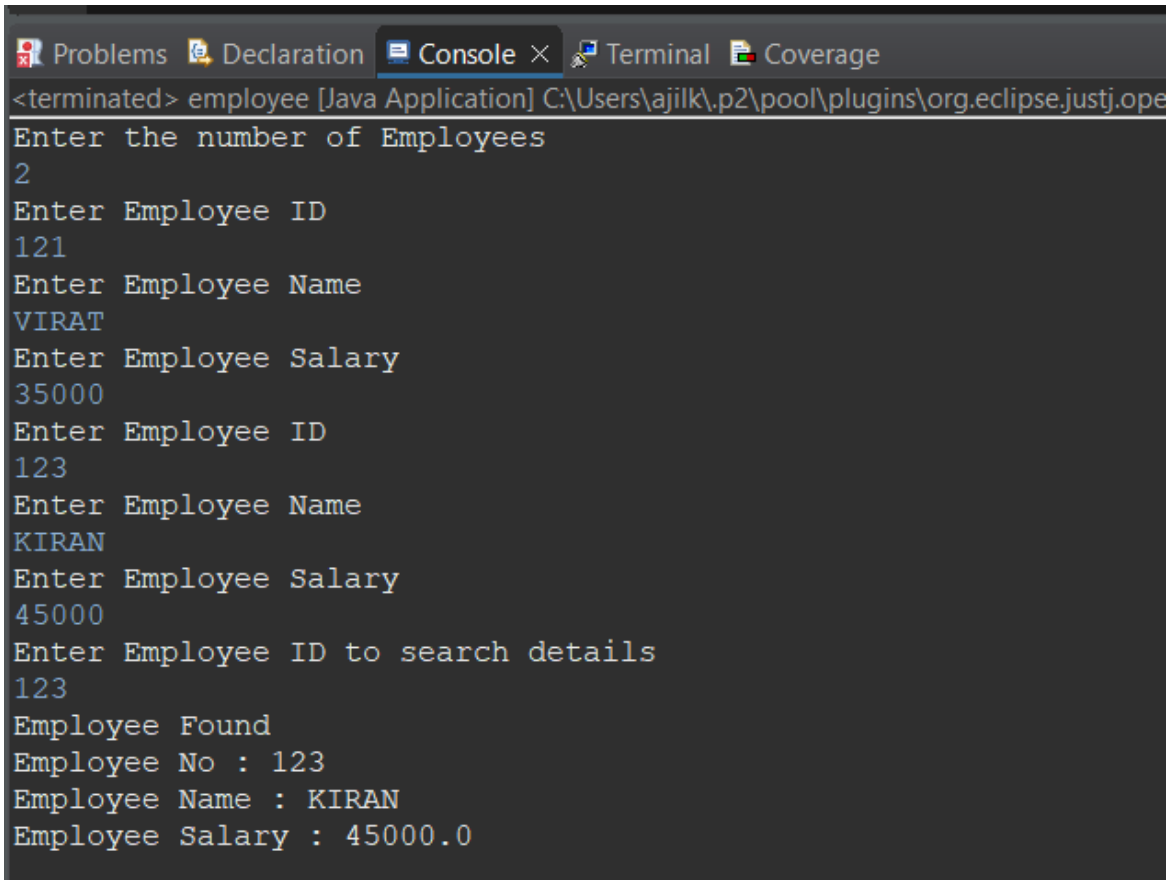
```
Scanner r=new Scanner(System.in);
int n=r.nextInt();
while(i<n)
{
        System.out.println("Enter Employee ID");
        id=r.nextInt();
        System.out.println("Enter Employee Name");
        r.nextLine();
        name=r.nextLine();
        System.out.println("Enter Employee Salary");
        salary=r.nextDouble();
        obj[i]=new employees(id,name,salary);
        i++;
}
System.out.println("Enter Employee ID to search details");
key=r.nextInt();
i=0;
while(i<n)
{
        if(obj[i].empNo==key)
        {
                System.out.println("Employee Found");
                System.out.println("Employee No : "+obj[i].empNo+"\nEmployee Name : "+obj[i].empName+"\nEmployee Salary : "+obj[i].empSalary);
                flag=1;
        }
        i++;
}
if(flag==0)
{
        System.out.println("Employee Not Found");
```

```
                }
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



COURSE OUTCOME 3

PROGRAM NO: 10

AIM:
Area of different shapes using overloaded functions.

ALGORITHM:
Step 1: Start
Step 2: Define the main class
Step 3: Define methods with the same method name that performs the area operation for each shape.
Step 4: Display the areas of each shape.

PROGRAM CODE:

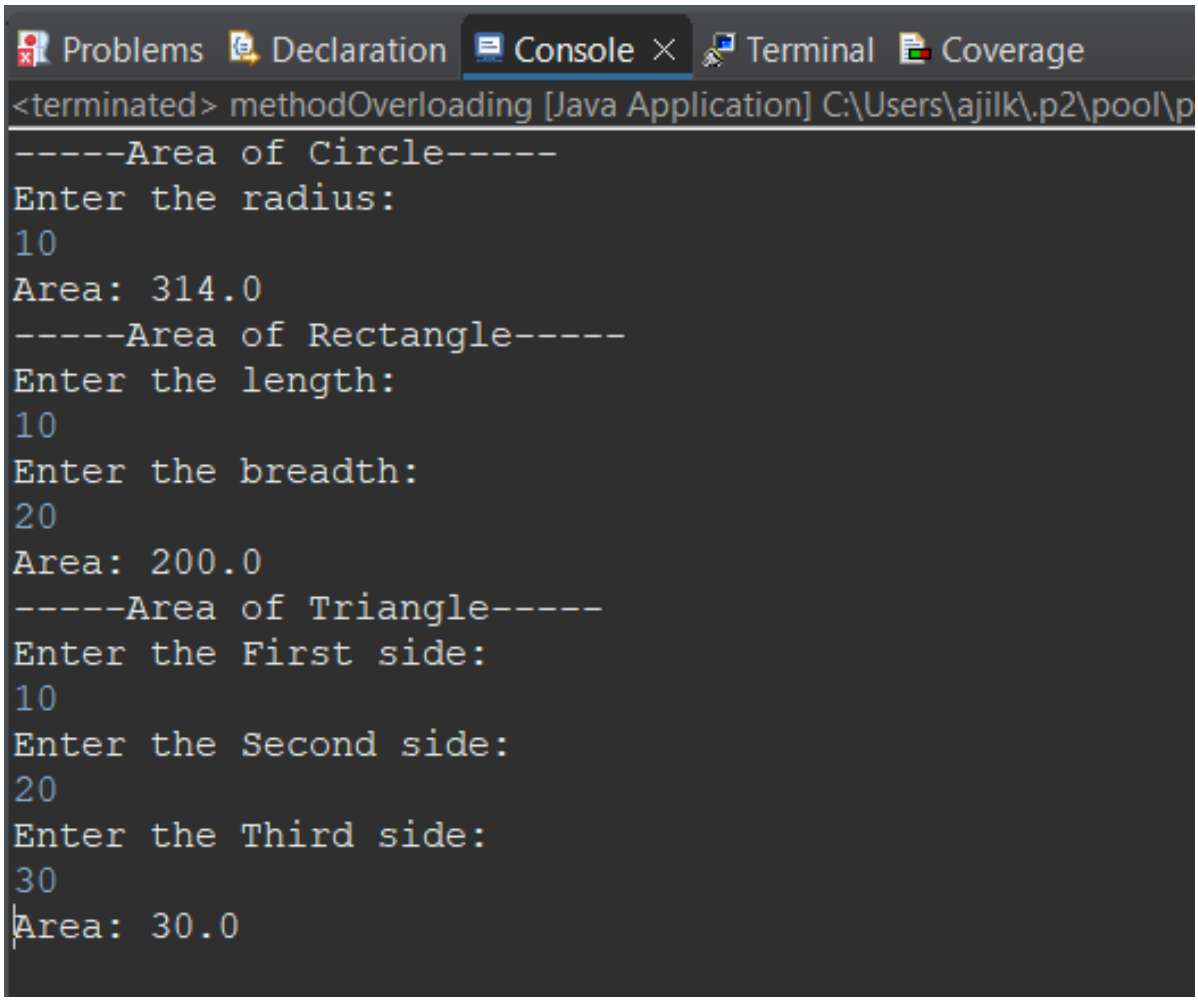
methodOverloading.java	import java.util.*;
	public class methodOverloading
	{
	public static double area(int r)
	{
	double a=(3.14*r*r);
	return a;
	}
	public static double area(int l,int b)
	{
	double a=(l*b);
	return a;
	}
	public static double area(int x,int y,int z)
	{
	double a=((0.5)*(x+y+z));
	return a;
	}
	public static void main(String[] args)
	{
	Scanner read = new Scanner(System.in);
	System.out.println("-----Area of Circle----

```
-");
                    System.out.println("Enter the radius: ");
                    int r1=read.nextInt();
                    System.out.println("Area: "+area(r1));
                    System.out.println("-----Area of
Rectangle-----");
                    System.out.println("Enter the length: ");
                    int l1=read.nextInt();
                    System.out.println("Enter the breadth: ");
                    int b1=read.nextInt();
                    System.out.println("Area: "+area(l1,b1));
                    System.out.println("-----Area of Triangle-
----");
                    System.out.println("Enter the First side:
");
                    int s1=read.nextInt();
                    System.out.println("Enter the Second
side: ");
                    int s2=read.nextInt();
                    System.out.println("Enter the Third side:
");
                    int s3=read.nextInt();
                    System.out.println("Area:
"+area(s1,s2,s3));
                }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:


Problems
Declaration
Console
Terminal
Coverage
<terminated> methodOverloading [Java Application] C:\Users\ajilk\.p2\pool\p
-----Area of Circle-----
Enter the radius:
10
Area: 314.0
-----Area of Rectangle-----
Enter the length:
10
Enter the breadth:
20
Area: 200.0
-----Area of Triangle-----
Enter the First side:
10
Enter the Second side:
20
Enter the Third side:
30
Area: 30.0

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.

ALGORITHM:

Step 1: Start
Step 2: Create class "employee" with the provided data members and define the constructors.
Step 3: Create another class "teacher" that performs inheritance of employee class and define constructors for the same.
Step 4: Create an array of objects in the corresponding class.
Step 5: Display the details for the number of teachers provided.

PROGRAM CODE:

teacher.java

```
import java.util.*;
class employees
{
        int empNo;
        String empName;
        Double empSalary;
        employees(int a,String b,Double c)
        {
                empNo=a;
                empName=b;
                empSalary=c;
        }
}
public class employee
{
        public static void main(String[] args)
        {
                int i=0,flag=0,id,key;
                String name;
```

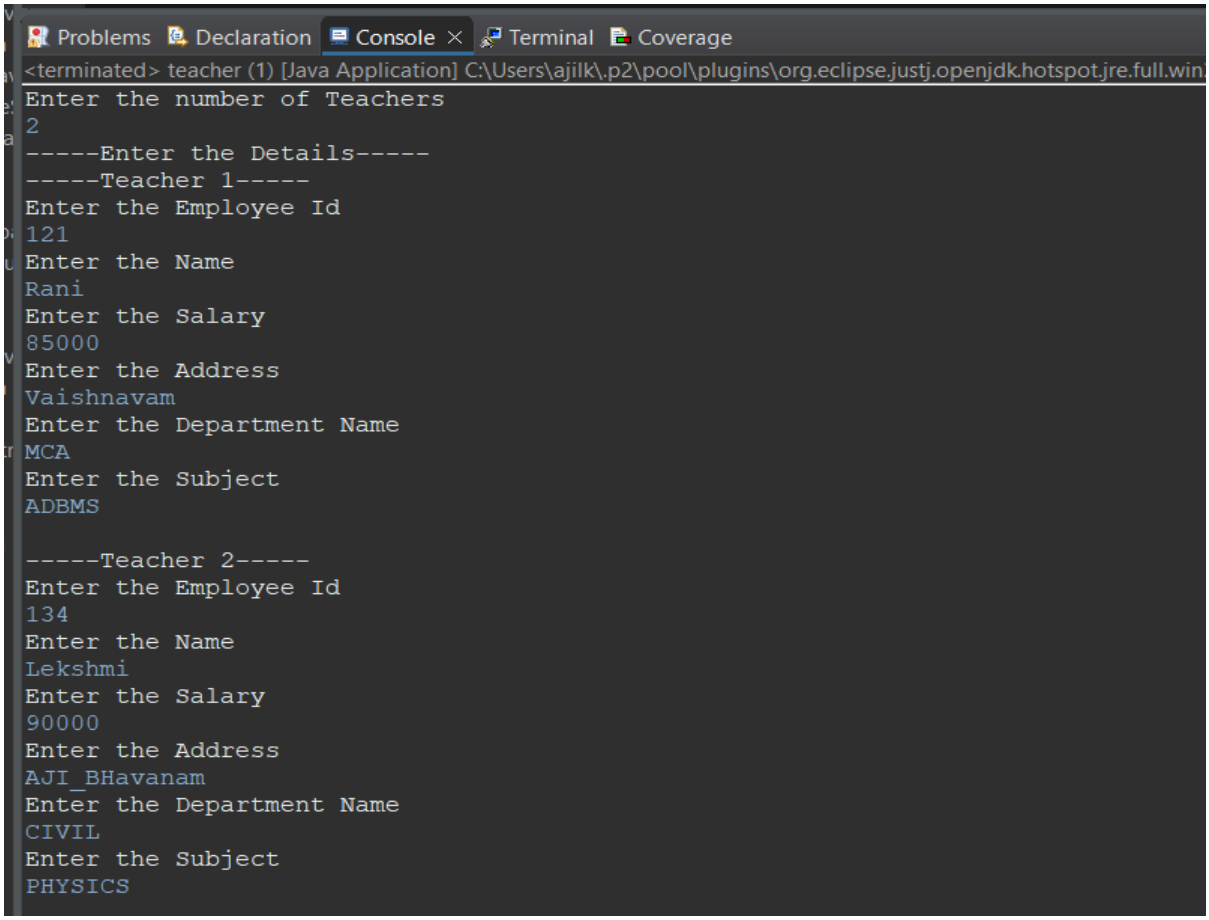
```
Double salary;
employees obj[]=new employees[10];
System.out.println("Enter the number of Employees");
Scanner r=new Scanner(System.in);
int n=r.nextInt();
while(i<n)
{
        System.out.println("Enter Employee ID");
        id=r.nextInt();
        System.out.println("Enter Employee Name");
        r.nextLine();
        name=r.nextLine();
        System.out.println("Enter Employee Salary");
        salary=r.nextDouble();
        obj[i]=new employees(id,name,salary);
        i++;
}
System.out.println("Enter Employee ID to search details");
key=r.nextInt();
i=0;
while(i<n)
{
        if(obj[i].empNo==key)
        {
                System.out.println("Employee Found");
                System.out.println("Employee No : "+obj[i].empNo+"\nEmployee Name : "+obj[i].empName+"\nEmployee Salary : "+obj[i].empSalary);
                flag=1;
        }
        i++;
}
```

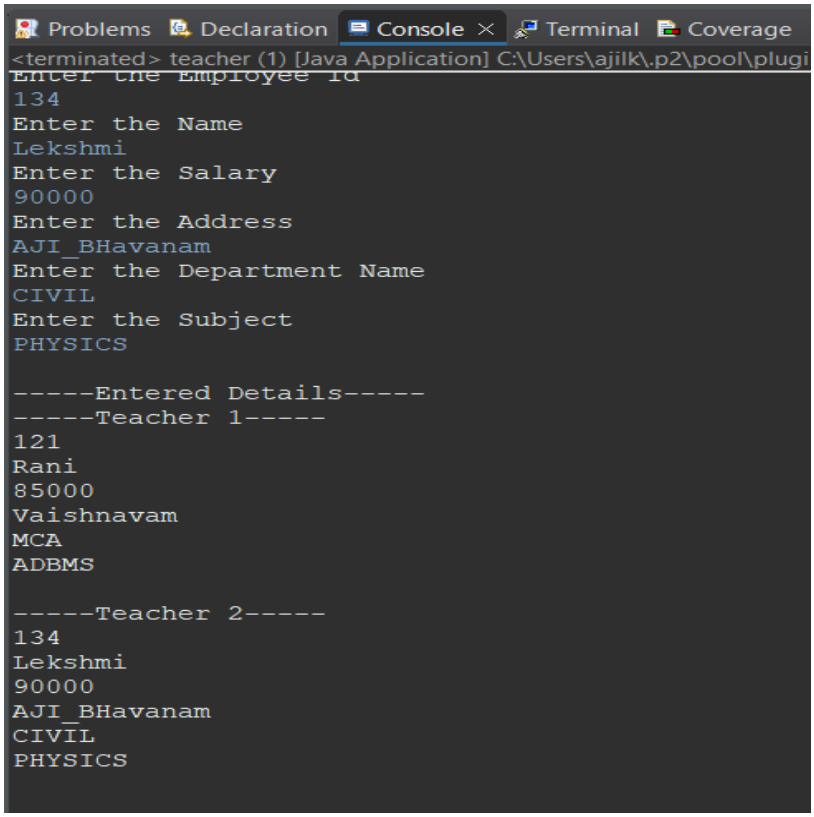
```
                if(flag==0)
                {
                        System.out.println("Employee Not Found");
                }
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:


Problems
Declaration
Console
Terminal
Coverage
<terminated> teacher (1) [Java Application] C:\Users\ajilk\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win
Enter the number of Teachers
2
-----Enter the Details-----
-----Teacher 1-----
Enter the Employee Id
121
Enter the Name
Rani
Enter the Salary
85000
Enter the Address
Vaishnavam
Enter the Department Name
MCA
Enter the Subject
ADBMS
-----Teacher 2-----
Enter the Employee Id
134
Enter the Name
Lekshmi
Enter the Salary
90000
Enter the Address
AJI_BHavanam
Enter the Department Name
CIVIL
Enter the Subject
PHYSICS


Problems
Declaration
Console
Terminal
Coverage
<terminated> teacher (1) [Java Application] C:\Users\ajilk\.p2\pool\plugi
134
Enter the Name
Lekshmi
Enter the Salary
90000
Enter the Address
AJI_BHavanam
Enter the Department Name
CIVIL
Enter the Subject
PHYSICS
-----Entered Details-----
-----Teacher 1-----
121
Rani
85000
Vaishnavam
MCA
ADBMS
-----Teacher 2-----
134
Lekshmi
90000
AJI_BHavanam
CIVIL
PHYSICS

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

ALGORITHM:

Step 1: Start
Step 2: Create class "Person" with the provided data members and define the constructors.
Step 3: Create another class "Teachers" that performs inheritance of employee class and define constructors for the same.
Step 4: Create an array of objects in the corresponding class.
Step 5: Display the details for the number of teachers provided.

PROGRAM CODE:

Teachers.java	package javaprgm;
	import java.util.*;
	class Person
	{
	String name,gender,address;
	int age;
	Person(String n,String g,String add,int a)
	{
	name = n;
	gender = g;
	address = add;
	age = a;
	}
	}
	class Employees extends Person
	{
	int empId,salary;

```
        String company,qualification;
        Employees(String name,String gender,String address,int age,int
id,String c,String q,int s)
        {
                super(name,gender,address,age);
                empId = id;
                company = c;
                qualification = q;
                salary = s;
        }
}
public class Teachers extends Employees
{
        String subject,department;
        int teacherId;
        private static Scanner read;
        Teachers(String name,String gender,String address,int age,int
empId,String company,String qualification,int salary,String sub,String
dept,int tId)
        {

        super(name,gender,address,age,empId,company,qualification,salary)
;
                subject=sub;
                department=dept;
                teacherId=tId;
        }
        void display()
        {
                System.out.println("Name: " + name);
                System.out.println("Gender: " + gender);
                System.out.println("Address: " + address);
                System.out.println("Age: " + age);
```

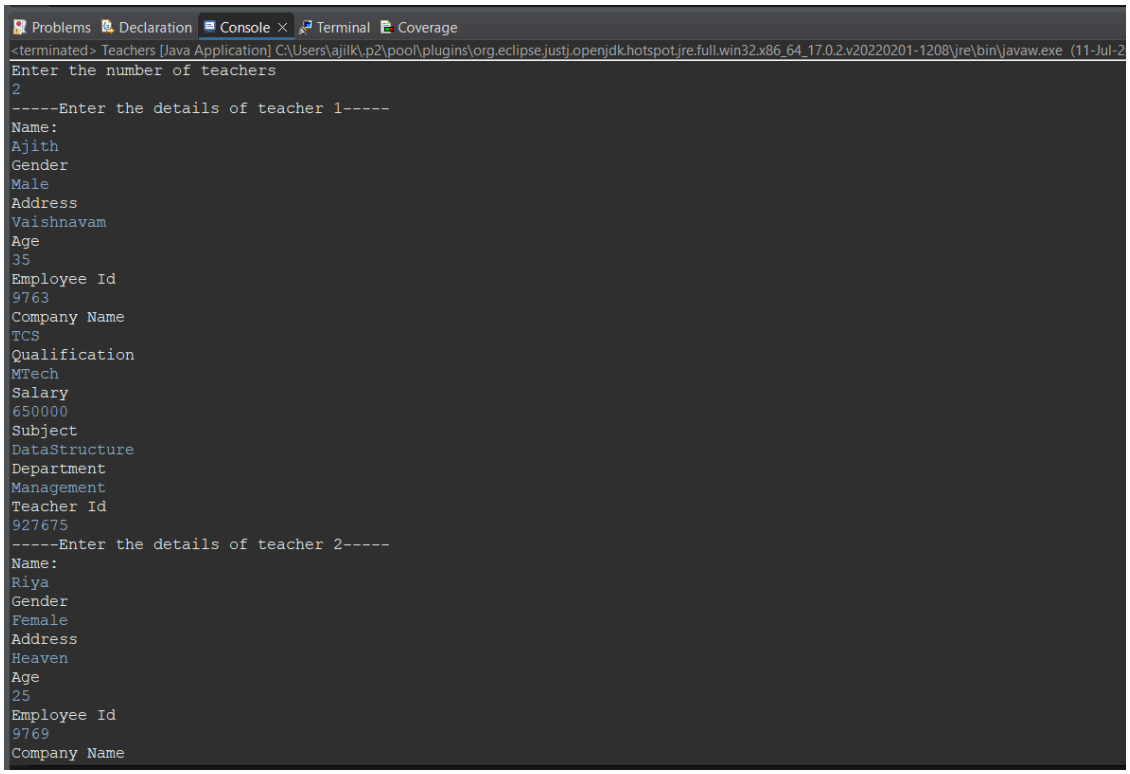
```
                System.out.println("Employee Id: "+ empId);
                System.out.println("Company Name: " + company);
                System.out.println("Qualification: " + qualification);
                System.out.println("Salary: " + salary);
                System.out.println("Subject: " + subject);
                System.out.println("Department: " + department);
                System.out.println("Teacher Id: " + teacherId);
        }
        public static void main(String[] args)
        {
                read = new Scanner(System.in);
                int i,no;
                Teachers obj[] = new Teachers[10];
                System.out.println("Enter the number of teachers");
                no = read.nextInt();
                for(i=0;i<no;i++)
                {
                        System.out.println("-----Enter the details of teacher "+(i+1)+"-----");
                        System.out.println("Name:");
                        String tname = read.next();
                        System.out.println("Gender");
                        String tgen = read.next();
                        System.out.println("Address");
                        String tadd = read.next();
                        System.out.println("Age");
                        int tage = read.nextInt();
                        System.out.println("Employee Id");
                        int tempid = read.nextInt();
                        System.out.println("Company Name");
                        String tcomp = read.next();
                        System.out.println("Qualification");
                        String tq = read.next();
```

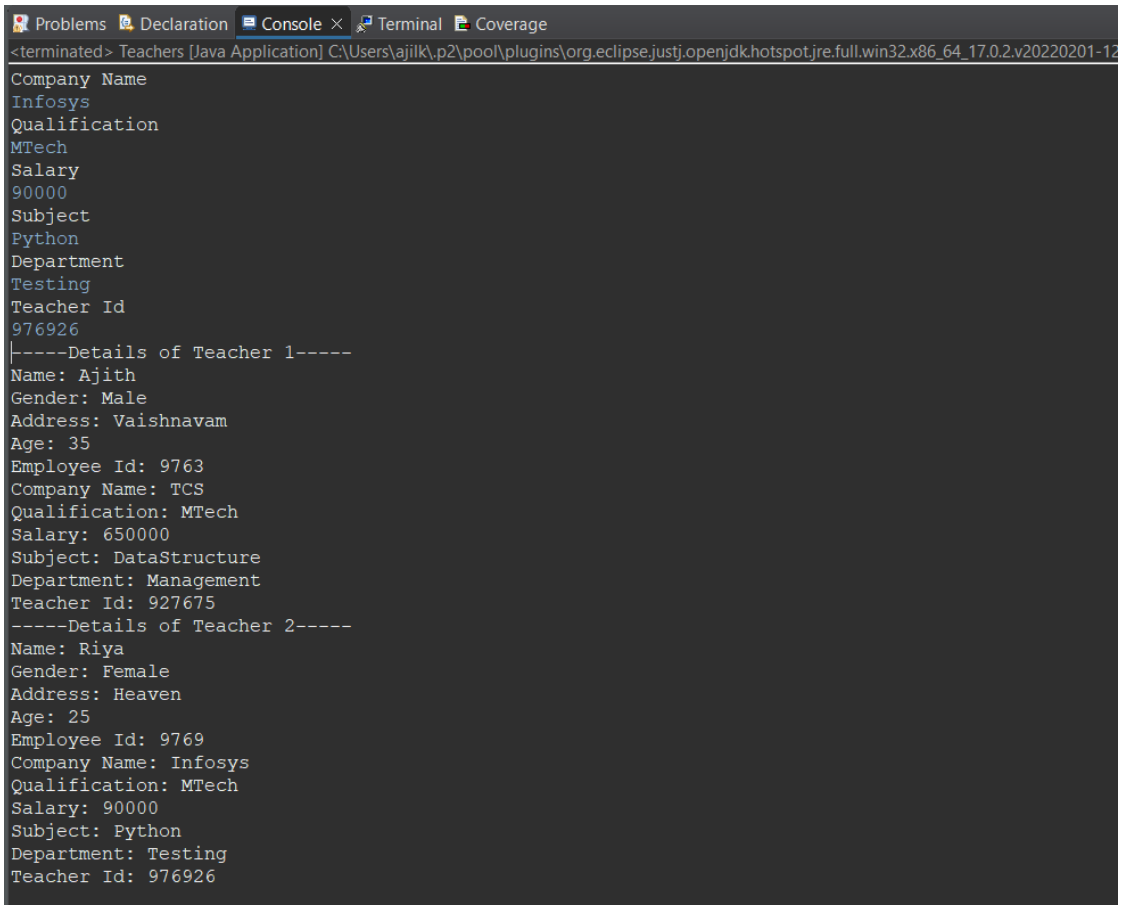
```
                    System.out.println("Salary");
                    int tsalary = read.nextInt();
                    System.out.println("Subject");
                    String tsub = read.next();
                    System.out.println("Department");
                    String tdept = read.next();
                    System.out.println("Teacher Id");
                    int tId= read.nextInt();
                    obj[i]=                                          new
Teachers(tname,tgen,tadd,tage,tempid,tcomp,tq,tsalary,tsub,tdept,tId);
                }
                for(i=0;i<no;i++)
                {
                    System.out.println("-----Details of Teacher "+(i+1)+"-
----");
                    obj[i].display();
                }
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:


Problems Declaration Console Terminal Coverage
<terminated> Teachers [Java Application] C:\Users\ajilk\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (11-Jul-2
Enter the number of teachers
2
-----Enter the details of teacher 1-----
Name:
Ajith
Gender
Male
Address
Vaishnavam
Age
35
Employee Id
9763
Company Name
TCS
Qualification
MTech
Salary
650000
Subject
DataStructure
Department
Management
Teacher Id
927675
-----Enter the details of teacher 2-----
Name:
Riya
Gender
Female
Address
Heaven
Age
25
Employee Id
9769
Company Name


Problems Declaration Console Terminal Coverage
<terminated> Teachers [Java Application] C:\Users\ajilk\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-12
Company Name
Infosys
Qualification
MTech
Salary
90000
Subject
Python
Department
Testing
Teacher Id
976926
-----Details of Teacher 1-----
Name: Ajith
Gender: Male
Address: Vaishnavam
Age: 35
Employee Id: 9763
Company Name: TCS
Qualification: MTech
Salary: 650000
Subject: DataStructure
Department: Management
Teacher Id: 927675
-----Details of Teacher 2-----
Name: Riya
Gender: Female
Address: Heaven
Age: 25
Employee Id: 9769
Company Name: Infosys
Qualification: MTech
Salary: 90000
Subject: Python
Department: Testing
Teacher Id: 976926

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.

ALGORITHM:

Step 1: Start
Step 2: Create class “publisher” and initialize its data members.
Step 3: Create classes book, literature, fiction. Each class inherit from their subsequent previous class and have its own data members.
Step 4: Create an array of objects in the corresponding class.
Step 5: Display the details of books provided.

PROGRAM CODE:

Book.java	package OOPS;
	import java.util.*;
	class Books
	{
	int bookId;
	String bookName;
	Scanner sc=new Scanner(System.in);
	Books()
	{
	System.out.println("Enter the bookid:");
	bookId=sc.nextInt();
	System.out.println("Enter the name of book:");
	bookName=sc.next();
	
	}
	
	}
	class Publisher extends Books
	{
	String publisherName;
	String edition;

```
        Scanner sc=new Scanner(System.in);
        Publisher()
        {
                System.out.println("Enter the Publisher Name:");
                publisherName=sc.nextLine();
                System.out.println("Enter the edition:");
                edition=sc.next();
        }
}
class Fiction extends Publisher
{
        String author;
        String genere;
        int price;
        Scanner sc=new Scanner(System.in);
        Fiction()
        {
                super();
                System.out.println("Enter the author:");
                author=sc.nextLine();
                System.out.println("Enter the genere:");
                genere=sc.nextLine();
                System.out.println("Enter the price:");
                price=sc.nextInt();
        }
        void display()
        {
                System.out.println("-----THE FICTION BOOK
DETAILS-----");
                System.out.println("The bookid is:"+bookId);
                System.out.println("The book name is:"+bookName);
                System.out.println("The publisher name
is:"+publisherName);
```

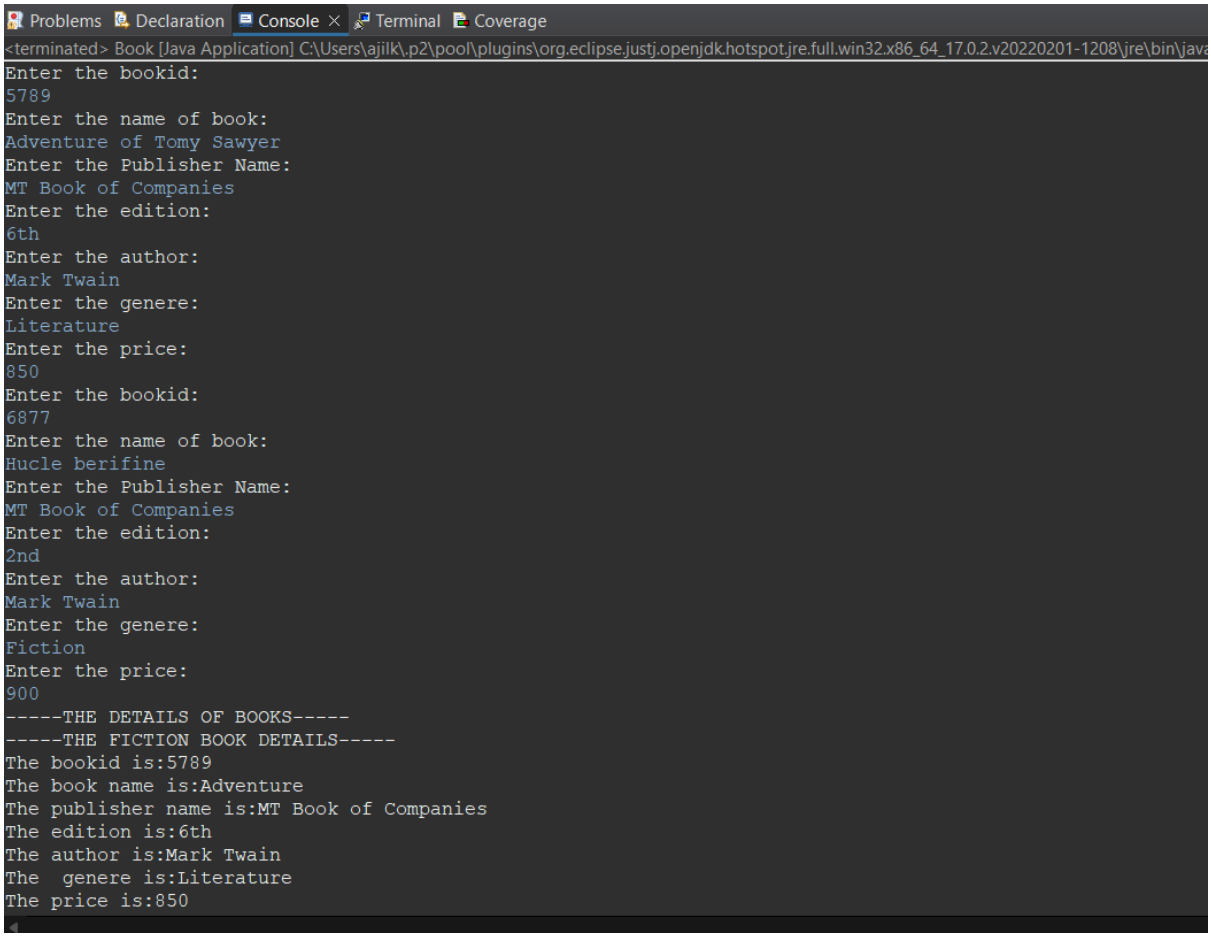
```
                System.out.println("The edition is:"+edition);
                System.out.println("The author is:"+author);
                System.out.println("The  genere is:"+genere);
                System.out.println("The price is:"+price);
        }
}
class Literature extends Publisher
{
        String autho;
        String gener;
        int pric;
        Scanner sc=new Scanner(System.in);
        Literature()
        {
                super();
                System.out.println("Enter the author:");
                autho=sc.nextLine();
                System.out.println("Enter the genere:");
                gener=sc.nextLine();
                System.out.println("Enter the price:");
                pric=sc.nextInt();
        }
        void display()
        {
                System.out.println("-----THE LITERATURE BOOK
DETAILS-----");
                System.out.println("The bookid is:"+bookId);
                System.out.println("The book name is:"+bookName);
                System.out.println("The publisher name
is:"+publisherName);
                System.out.println("The edition is:"+edition);
                System.out.println("The author is:"+autho);
                System.out.println("The  genere is:"+gener);
```

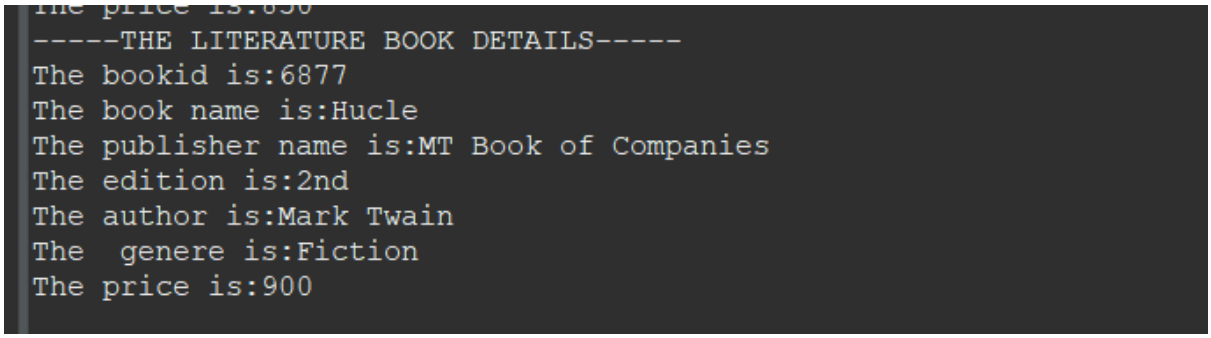
```
                System.out.println("The price is:"+pric);
        }
}
class Book {
        public static void main(String[] args)
        {
                Fiction fict=new Fiction();
                Literature liter=new Literature();
                System.out.println("-----THE DETAILS OF BOOKS-----
");
                fict.display();
                liter.display();
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:


Problems
Declaration
Console
Terminal
Coverage
<terminated> Book [Java Application] C:\Users\ajilk\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\java
Enter the bookid:
5789
Enter the name of book:
Adventure of Tomy Sawyer
Enter the Publisher Name:
MT Book of Companies
Enter the edition:
6th
Enter the author:
Mark Twain
Enter the genere:
Literature
Enter the price:
850
Enter the bookid:
6877
Enter the name of book:
Hucle berifine
Enter the Publisher Name:
MT Book of Companies
Enter the edition:
2nd
Enter the author:
Mark Twain
Enter the genere:
Fiction
Enter the price:
900
-----THE DETAILS OF BOOKS-----
-----THE FICTION BOOK DETAILS-----
The bookid is:5789
The book name is:Adventure
The publisher name is:MT Book of Companies
The edition is:6th
The author is:Mark Twain
The genere is:Literature
The price is:850


-----THE LITERATURE BOOK DETAILS-----
The bookid is:6877
The book name is:Hucle
The publisher name is:MT Book of Companies
The edition is:2nd
The author is:Mark Twain
The genere is:Fiction
The price is:900

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: Start
Step 2: Create an interface.
Step 3: Create classes Student and Sports that implements the interface results.
Step 4: Display the academic and sports score of the student.

PROGRAM CODE:

Result.java

```
package OOPS;
import java.util.Scanner;
class Sports
{
   String sport;
   double rating;
   Sports(String s, double r)
   {
      sport = s;
      rating = r;
   }

}
class Student extends Sports
{
   String grade;
   double overallPer;
   Student(String s, double r,String g, double p)
   {
      super(s, r);
      grade = g;
      overallPer = p;
   }
```

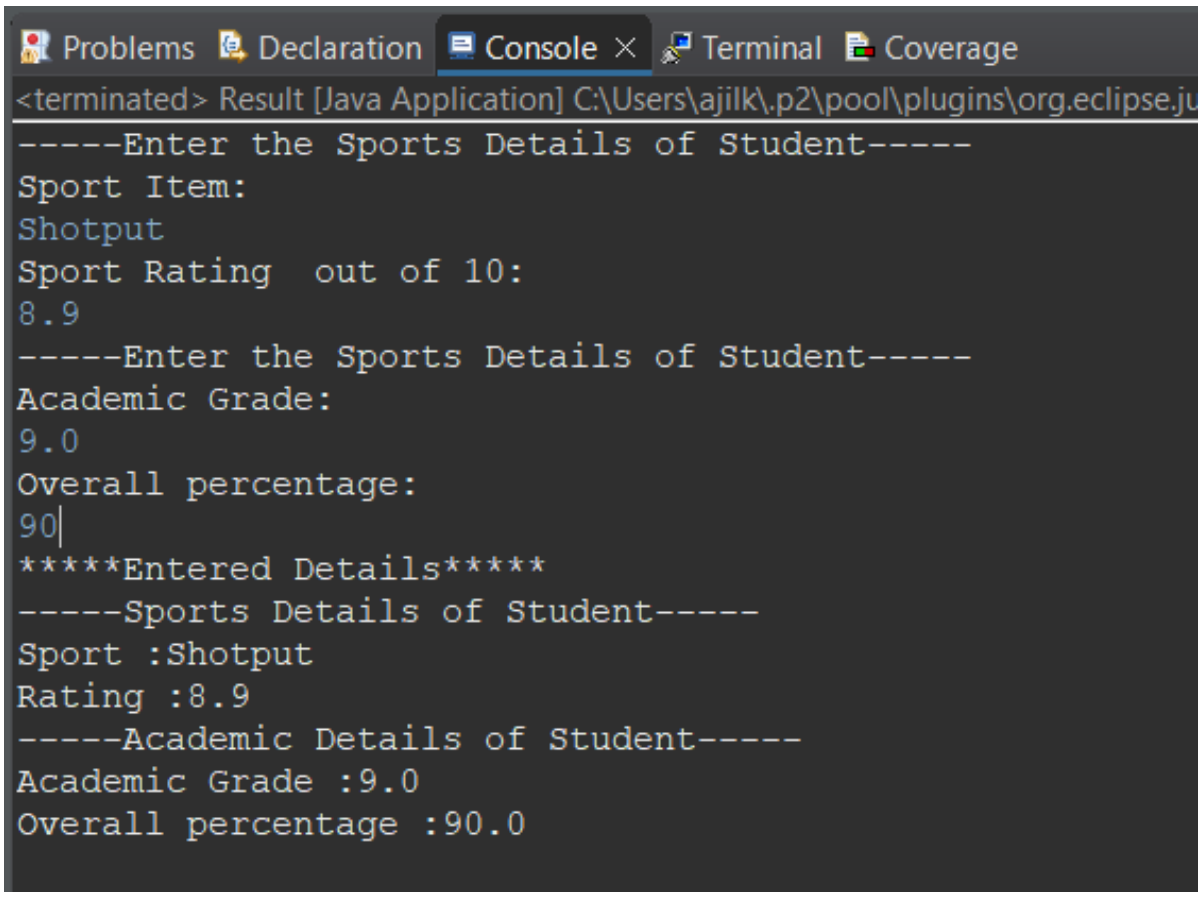
```
}
public class Result extends Student
{
   Result(String s, double r,String g, double p)
   {
      super(s, r, g, p);
   }
   void display()
   {
         System.out.println("*****Entered Details*****");
      System.out.println("-----Sports Details of Student-----");
      System.out.println("Sport :"+sport);
      System.out.println("Rating :"+rating);
      System.out.println("-----Academic Details of Student-----");
      System.out.println("Academic Grade :"+grade);
      System.out.println("Overall percentage :"+overallPer);
   }

   public static void main(String[] args)
   {
      Scanner read =new Scanner(System.in);
      System.out.println("-----Enter the Sports Details of Student-----");
      System.out.println("Sport Item: ");
      String si =read.next();
      System.out.println("Sport Rating  out of 10: ");
      double sr =read.nextDouble();
      System.out.println("-----Enter the Sports Details of Student-----");
      System.out.println("Academic Grade: ");
      String ag =read.next();
      System.out.println("Overall percentage: ");
      double op =read.nextDouble();
      read.close();
      Result obj= new Result(si,sr,ag,op);
```

```
        obj.display();
    }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



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: Start
Step 2: Create an interface Calculation that has the methods to take inputs and compute area and perimeter.
Step 3: Create classes Student and Sports that implements calculation.
Step 4: Display the area and perimeter of circle or rectangle depending upon the choice the user selects.

PROGRAM CODE:

areaPerimeter.java	package OOPS;
	
	import java.util.Scanner;
	
	interface prototype
	{
	double pi=3.14;
	void area();
	void perimeter();
	}
	class circle implements prototype
	{
	Float r;
	double area,per;
	circle()
	{ Scanner sc=new Scanner(System.in);
	System.out.println("enter radius");
	r=sc.nextFloat();
	}
	public void area()
	{

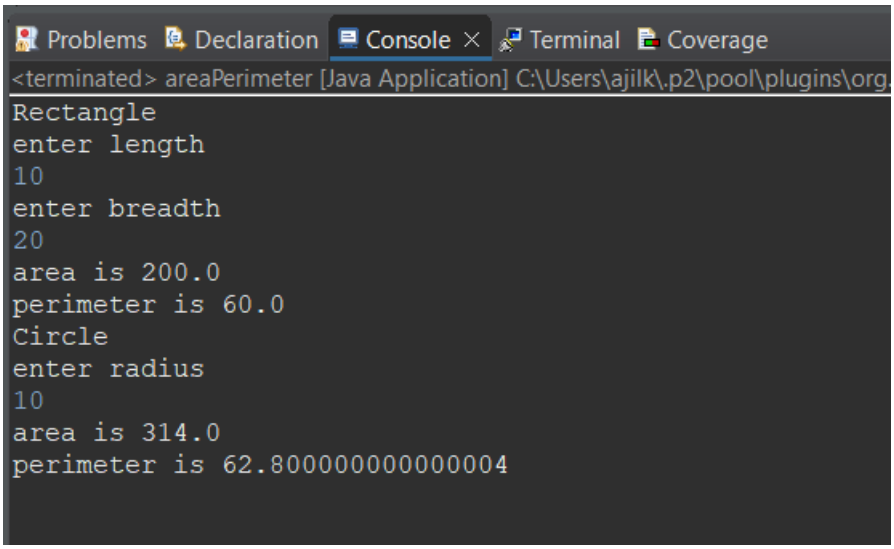
```
                area=pi*r*r;
                System.out.println("area is "+area);
        }
        public void perimeter()
        {
                per=2*pi*r;
                System.out.println("perimeter is "+per);

        }

}
class rectangle implements prototype
{
        float l,b;
        double area,per;
        rectangle()
        {       Scanner sc=new Scanner(System.in);
                System.out.println("enter length");
                l=sc.nextFloat();
                System.out.println("enter breadth");
                b=sc.nextFloat();
        }
        public void area()
        {
                area=l*b;
                System.out.println("area is "+area);
        }
        public void perimeter()
        {
                per=2*(l+b);
                System.out.println("perimeter is "+per);

        }
```

```
}
public class areaPerimeter {


        public static void main(String args[])
        {
                prototype p;
                System.out.println("Rectangle");
                rectangle r=new rectangle();
                p=r;
                p.area();
                p.perimeter();
                System.out.println("Circle");
                circle c=new circle();
                p=c;
                p.area();
                p.perimeter();
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



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: Start
Step 2: Create interface calc that performs the calculation operations.
Step 3: Create class bill that implements the interface calc.
Step 4: Display the net amount by acquiring the data for the specific inputs.

PROGRAM CODE:

calculateBill.java

```
package OOPS;
import java.util.Scanner;
import java.util.Date;
public class calculateBill implements outline
{
        int id,quantity,unit,total,orderid;
        String  name;
        Date d;
        public void addItem()
        {
                System.out.println("\nEnter the item id");
                id=s.nextInt();
                s.nextLine();
                System.out.println("\nEnter the item name");
                name=s.nextLine();
                System.out.println("\nEnter the item quantity");
                quantity=s.nextInt();
                System.out.println("\nEnter the item unit price");
```

```
                unit=s.nextInt();
                s.nextLine();
                total=unit*quantity;
        }
        public void forHeader()
        {
                d=new Date();
                System.out.println("Enter the Order ID");
                orderid=s.nextInt();
                s.nextLine();
        }
        public void showHeader()
        {
                System.out.println("\nOrder ID : "+orderid);
                System.out.println("\nDate :"+d.toString());
        }
        public void prepareBill()
        {
                System.out.format("%10d %10s %10d %10d
%10d",id,name,quantity,unit,total);
        }
        public static void main(String[] args)
        {
                Scanner s=new Scanner(System.in);
                int ch=1;
                int n=5,i=0,net=0;
                calculateBill newbill[]=new calculateBill[n];
                while(ch==1 && i<n )
                {
                        System.out.println("Item "+(i+1));
                        newbill[i]=new calculateBill();
                        if(i==0){
                                newbill[i].forHeader();
```

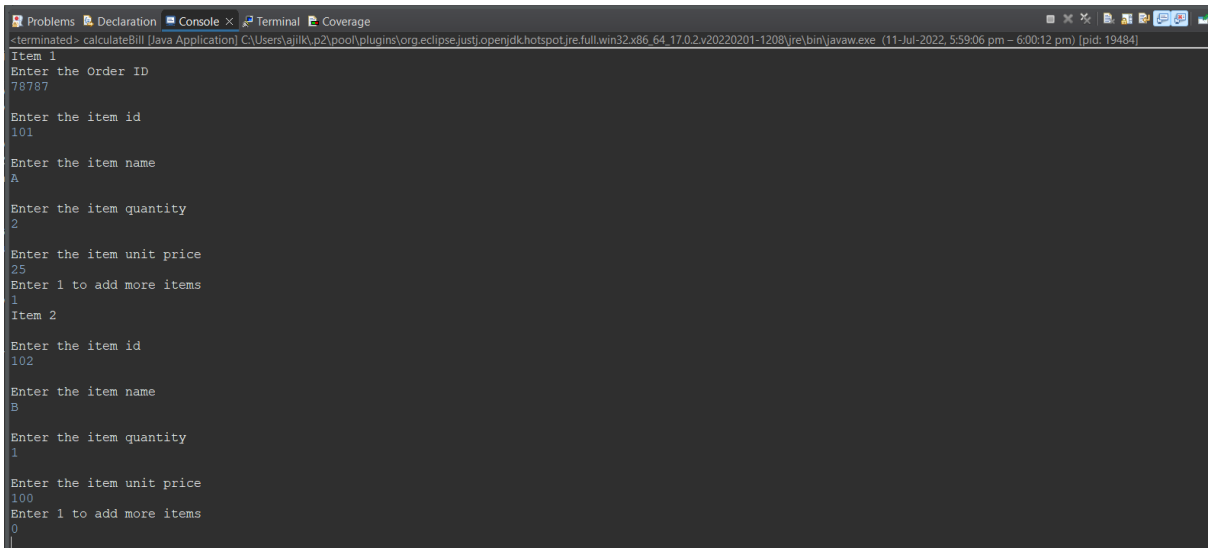
```
                    }
                    newbill[i].addItem();
                    i++;
                    System.out.println("Enter 1 to add more items");
                    ch=s.nextInt();
                }
                newbill[0].showHeader();
                System.out.printf("%10s %10s %10s %10s %10s","PRODUCT ID", "NAME", "QUANTITY", "UNIT PRIZE", "TOTAL");
                System.out.println();
                for(int z=0;z<55;z++)
                {
                    System.out.print("-");
                }
                System.out.println();
                for(int j=0;j<i;j++)
                {
                    newbill[j].prepareBill();
                    System.out.println();
                }
                for(int z=0;z<55;z++)
                {
                    System.out.print("-");
                }
                System.out.println();
                for(int j=0;j<i;j++)
                {
                    net+=newbill[j].total;
                }
                System.out.println("Net Amount :"+net);
        }
}
```

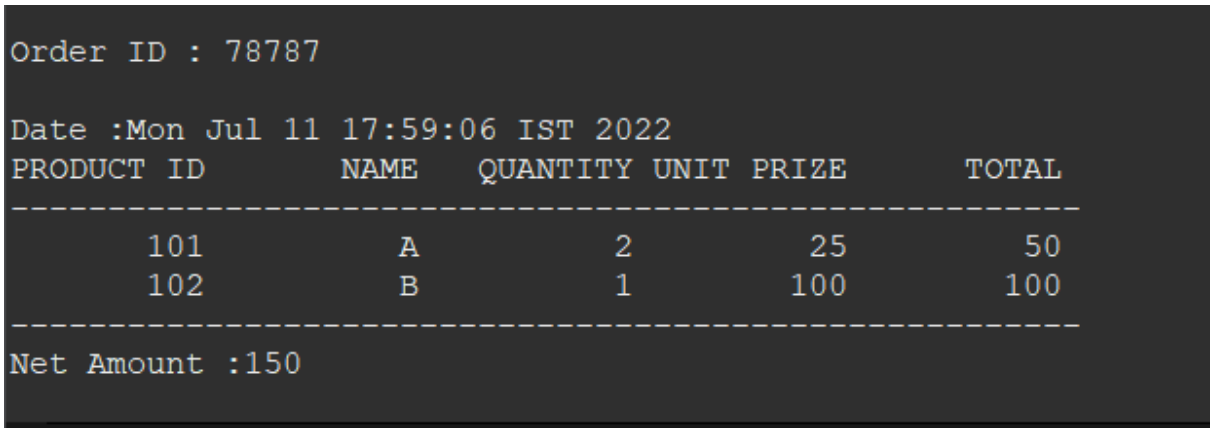
	interface outline
	{
	Scanner s=new Scanner(System.in);
	public void prepareBill();
	void addItem();
	void forHeader();
	void showHeader();
	}

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:





COURSE OUTCOME 4

PROGRAM 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.

ALGORITHM:

Step 1: Start
Step 2: To create a package named graphics, create a folder of the same name in the directory.
Step 3: Inside the graphics folder, create modules for finding the areas of rectangle, circle, triangle, and square.
Step 4: Outside the graphics folder, write a program to access the modules mentioned above and print the output.

PROGRAM CODE:

co4_pg1.java

```
import java.util.*;
import graphics.rectangle;
import graphics.triangle;
import graphics.circle;
import graphics.square;
public class co4_pg1
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		int s,l,b,r;
		rectangle rectangle = new rectangle();
		triangle triangle = new triangle();
		circle circle = new circle();
		square square = new square();
		System.out.println("Enter the length of the rectangle :");
		l=sc.nextInt();
		System.out.println("Enter the breadth of the rectangle
:");
		b=sc.nextInt();
		rectangle.area(l,b);
		System.out.println("Enter the height of the triangle :");
		l=sc.nextInt();
		System.out.println("Enter the base of the triangle :");
		b=sc.nextInt();
		triangle.area(l,b);
		System.out.println("Enter the side of square :");
		s=sc.nextInt();
		square.area(s);
		System.out.println("Enter the radius of circle :");
```

	r=sc.nextInt();
	circle.area(r);
	}
	}

circle.java	package graphics;
	interface cir
	{
	void area(int r);
	}
	public class circle implements cir
	{
	public void area(int r)
	{
	System.out.println("Area of circle :"+3.14*r*r);
	}
	}

rectangle.java	package graphics;
	interface rect
	{
	public void area(int len,int bre);
	}
	public class rectangle implements rect
	{
	public void area(int leng,int bre)
	{
	System.out.println("Area of rectangle :"+leng*bre);
	}
	}

square.java	package graphics;
	interface squ
	{
	void area(int s);
	}
	public class square implements squ
	{
	public void area(int s)
	{
	System.out.println("Area of square :"+s*s);
	}
	}

triangle.java	package graphics;
	interface tri
	{
	void area(int b,int h);
	}
	public class triangle
	{
	public void area(int b,int h)
	{

```
                System.out.println("Area of triangle :"+0.5*b*h);
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Enter the length of the rectangle :
2
Enter the breadth of the rectangle :
3
Area of rectangle :6
Enter the height of the triangle :
4
Enter the base of the triangle :
2
Area of triangle :4.0
Enter the side of square :
4
Area of square :16
Enter the radius of circle :
5
Area of circle :78.5
```

PROGRAM 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.

ALGORITHM:

Step 1: Start

Step 2: To create a package named arithmetic, create a folder of the same name in the directory.

Step 3: Inside arithmetic package, create module to perform addition, subtraction, multiplication, and division of 2 numbers.

Step 4: Outside the folder, write another program that access the above module and print the output.

Step 5: Stop

PROGRAM CODE:

co4_pg2.java

```
import java.util.*;
import arithmetic.calculate;
public class co4_pg2
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                calculate calculate = new calculate();
                int i,j;
                System.out.println("Enter the two numbers");
                i=sc.nextInt();
                j=sc.nextInt();
                calculate.add(i,j);
                calculate.sub(i,j);
                calculate.mul(i,j);
                calculate.div(i,j);
        }
}
```

calculate.java

```
package arithmetic;
interface cal
{
        void add(int n1,int n2);
        void sub(int n1,int n2);
        void mul(int n1,int n2);
        void div(int n1,int n2);
}
```

```
public class calculate implements cal
{
	public void add(int n1,int n2)
	{
		System.out.println("Addition :"+(n1+n2));
	}
	public void sub(int n1,int n2)
	{
		System.out.println("Subtraction :"+(n1-n2));
	}
	public void mul(int n1,int n2)
	{
		System.out.println("Multiplication :"+(n1*n2));
	}
	public void div(int n1,int n2)
	{
		System.out.println("Division :"+(float)(n1/n2));
	}
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Enter the two numbers
2
2
Addition :4
Subtraction :0
Multiplication :4
Division :1.0
```

PROGRAM NO: 19

AIM:

Write a user defined exception class to authenticate the user name and password.

ALGORITHM:

Step 1: Start

Step 2: Create two classes UsernameException and PasswordException that represents userdefined Exception.

Step 2: Inside the main class accept username and password

Step 3: Try block defines error code, if the username doesn't satisfies the required condition throws exception

Step 4: If an error occurs in the try block catch block is executed

Step 5: Finally block executed regardless of the result.

Step 6: Stop

PROGRAM CODE:

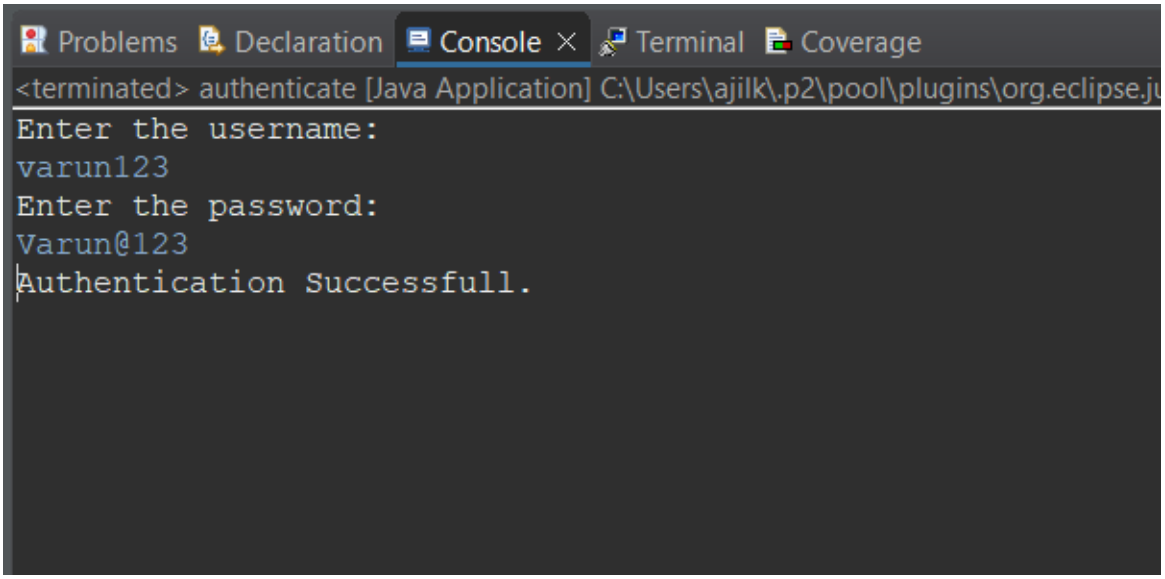
authenticate.java	package javaprgm;
	import java.util.*;
	class AuthenticationException extends Exception
	{
	public AuthenticationException(String str)
	{
	System.out.println(str);
	}
	}
	public class authenticate
	{
	public static void main(String[] args)
	{
	try
	{
	Scanner read = new Scanner(System.in);
	System.out.println("Enter the username: ");
	String uname = read.nextLine();
	System.out.println("Enter the password: ");
	String pass = read.nextLine();
	if(uname.equals("varun123") &&
	pass.equals("Varun@123"))
	{

```
                                        System.out.println("Authentication
Successfull.");
                        }
                        else
                        {
                                throw new
AuthenticationException("Authentication Failed.");
                        }
                }
                catch(AuthenticationException a)
                {
                        System.out.println(a);
                }
        }

}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 20

AIM:

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

ALGORITHM:

Step 1: Start

Step 2: Create a class named NegativeIntegerException that inherits Exception class with a constructor inside which we call the Exception class constructor and pass error meaasage.

Step 3: Inside the main(), Read the limit of array

Step 4: Inside the try block,read the array and check if any element is less than 0

Step 5: If true, throw NegException with appropriate message.

Step 6: Calculate the average of the array and print it

Step 7: Inside the catch exception, Print “Negative Integer”

Step 8: Stop

PROGRAM CODE:

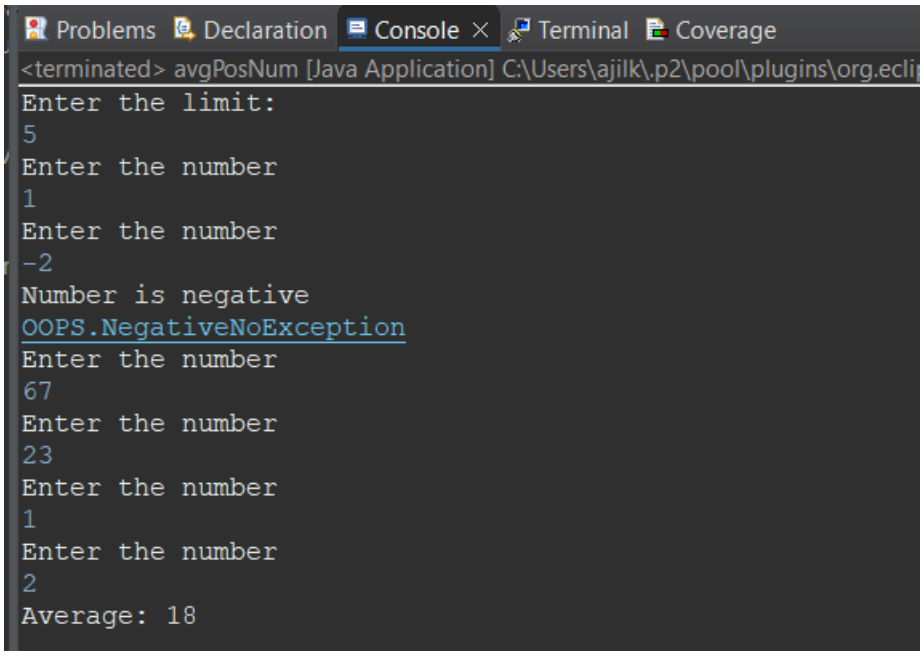
avgPosNum.java	package OOPS;
	import java.util.Scanner;
	class NegativeNoException extends Exception
	{
	public NegativeNoException(String str)
	{
	System.out.println(str);
	}
	}
	public class avgPosNum
	{
	public static void main(String[] args)
	{
	Scanner read = new Scanner(System.in);
	System.out.println("Enter the limit:\t");
	int n,i,sum = 0;
	n = read.nextInt();
	for(i=0;i<n;i++)
	{
	System.out.println("Enter the number");
	String no = read.next();

```
                        int num = Integer.parseInt(no);
                        try
                        {
                                if(num < 0)
                                {
                                        throw new
NegativeNoException("Number is negative");
                                }
                                else
                                {
                                        sum=sum+num;
                                }
                        }
                        catch (NegativeNoException m)
                        {
                                System.out.println(m);
                                i--;
                        }
                }
                System.out.println("Average: "+(sum/n));
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM 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)

ALGORITHM:

Step 1: Start

Step 2: Create a class named mul that inherits Thread class with member function as run()

Step 3: Inside run(), Print the multiplication table for 5

Step 4: Create a class named prime that inherits Thread class with memebr function run()

Step 5: Inside run(),Print the prime numbers upto the limit of user's choice

Step 6: Inside the main(), create an object for the classes and call start() using each object

Step 7: Stop

PROGRAM CODE:

co4_pg5.java	package check;
	import java.util.*;
	class t1 extends Thread
	{
	public synchronized void run()
	{
	System.out.println("Multiplication table Prime numbers");
	for(int i=1;i<=5;i++)
	{
	System.out.println(i+" * "+5+" = "+i*5);
	}
	}
	}
	class t2 extends Thread
	{
	int n;
	t2(int n)
	{
	this.n=n;
	}
	public synchronized void run()
	{
	for(int i=1;i<=n;i++)

```
                {
                        int c=0;
                        for(int j=i;j>=1;j--)
                        {
                                if(i%j==0)
                                {
                                        c=c+1;
                                }
                        }
                        if(c==2)
                        {
                                System.out.println("
"+i);
                        }
                }
        }
}
public class co4_pg5
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                int n;
                System.out.println("Enter the limit");
                n=sc.nextInt();
                t1 obj1=new t1();
                t2 obj2=new t2(n);
                obj1.start();
                obj2.start();
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Enter the limit
10
Multiplication table    Prime numbers
                           2
1 * 5 = 5
2 * 5 = 10
3 * 5 = 15
4 * 5 = 20
5 * 5 = 25
                           3
                           5
                           7
```

PROGRAM 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)

ALGORITHM:

Step 1: Start

Step 2: Create a class named even that implements Runnable interface with function run()

Step 3: Inside run(), we read the limit for printing even numbers and print it using for loop.

Step 4: Create another calss fib that implements Runnable interface with function run().

Step 5: Inside run(), Initialise n1 as 0,n2 as 1 and n3 as 0.

Step 6: Check if n<0, if true, print “Enter a positive number” else goto step 7

Step 7: Repeat step8 to 11 until n3>n

Step 8: Print n1

Step 9: n3=n1+n2

Step 10: n1=n2

Step 11: n2=n3

Step 12: Create object e of even and create an object t1 of Thread with its parameterized constructor passing e as parameter

Step 13: Call start() using t1

Step 14: Do the same for class odd with Thread object t2 and call start() using t2

Step 15: Stop

PROGRAM CODE:

co4_pg6.java	package check;
	import java.util.*;
	class th1 implements Runnable
	{
	int n,a=0,b=1,sum;
	th1(int n)
	{
	this.n=n;
	}

```
	public synchronized void run()
	{
		System.out.println("Fibonacci series    Even numbers");
		System.out.println(a+" "+b);
		for(int i=1;i<=10;i++)
		{
			sum=a+b;
			System.out.println(sum);
			a=b;
			b=sum;
		}
	}
}
class th2 implements Runnable
{
	int n;
	th2(int n)
	{
		this.n=n;
	}
	public synchronized void run()
	{
		for(int i=1;i<=n;i++)
		{
			if(i%2==0)
			{
			System.out.println("                    "+i);
			}
		}
	}
}
public class co4_pg6
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		int n;
		System.out.println("Enter the limit");
		n=sc.nextInt();
		th1 obj1=new th1(n);
		Thread o1=new Thread(obj1);
		th2 obj2=new th2(n);
		Thread o2=new Thread(obj2);
		o1.start();
		o2.start();
	}
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Enter the limit
10
Fibonacci series    Even numbers
                            2
                            4
                            6
                            8
0 1
1
2
3
5
8
13
21
34
55
89
                            10
```

PROGRAM NO: 23

AIM:

Producer/Consumer using ITC

ALGORITHM:

Step 1: Start

Step 2: In PC class (A class that has both produce and consume methods), a linked list of jobs and a capacity of the list is added to check that producer does not produce if the list is full.

Step 3: In Producer class, the value is initialized as 0.

Step 4: We have an infinite outer loop to insert values in the list. Inside this loop, we have a synchronized block so that only a producer or a consumer thread runs at a time.An inner loop is there before adding the jobs to list that checks if the job list is full, the producer thread gives up the intrinsic lock on PC and goes on the waiting state.

Step 5: If the list is empty, the control passes to below the loop and it adds a value in the list.

Step 6: In the Consumer class, we again have an infinite loop to extract a value from the list.

Inside, we also have an inner loop which checks if the list is empty.

Step 7: If it is empty then we make the consumer thread give up the lock on PC and passes the control to producer thread for producing more jobs.

Step 8: If the list is not empty, we go round the loop and removes an item from the list.

Step 9: In both the methods, we use notify at the end of all statements. The reason is simple, once you have something in list, you can have the consumer thread consume it, or if you have consumed something, you can have the producer produce something.

Step 10: sleep() at the end of both methods just make the output of program run in step wise manner and not display everything all at once so that you can see what actually is happening in the program.

Step 11: Stop

PROGRAM CODE:

co4_pg7.java	import java.util.ArrayList;
	class Producer1 implements Runnable
	{
	ArrayList<Integer>l;
	int i=0;
	Producer1(ArrayList<Integer> l)
	{

```
                this.l=l;
        }
        public void run()
        {
                try
                {
                        while(true)
                        {
                                produce1(i++);
                                if((i)==20)
                                {
                                        break;
                                }
                        }
                }
                catch(Exception e)
                {
                        System.out.println(e.getMessage());
                }
        }
        public void produce1(int i) throws Exception
        {
                synchronized(l)
                {
                        System.out.println("produced:"+i);
                        l.add(i);
                        l.notify();
                }
                synchronized(l)
                {
                        while(l.size()==5)
                        {
                                System.out.println("Production Full");
                                l.wait();
                        }
                }
        }
}
class Consumer1 implements Runnable
{
        ArrayList<Integer>l;
        Consumer1(ArrayList<Integer>l)
        {
                this.l=l;
        }
        public void run()
        {
                while(true)
                {
                        try
                        {
```

```
                                        consume1();
                                }
                                catch (Exception e)
                                {
                                        System.out.println(e.getMessage());
                                }
                        }
                }
                public void consume1() throws Exception
                {
                        synchronized(l)
                        {
                                while(l.isEmpty())
                                {
                                        System.out.println("fully Consumed");
                                        l.notify();
                                        Thread.sleep(500);
                                        l.wait();
                                }
                        }
                        synchronized(l)
                        {
                                Thread.sleep(500);
                                System.out.println("Consumed"+l.remove(0));
                        }
                }
}
public class co4_pg7
{
        public static void main(String args[])
        {
                ArrayList<Integer>l=new ArrayList<>();
                Producer1 obj=new Producer1(l);
                Thread t1=new Thread(obj);
                Consumer1 obj2=new Consumer1(l);
                Thread t2=new Thread(obj2);
                t1.start();
                t2.start();
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
produced:0
produced:1
produced:2
produced:3
produced:4
Production Full
Consumed0
Consumed1
Consumed2
Consumed3
Consumed4
fully Consumed
produced:5
produced:6
produced:7
produced:8
produced:9
Production Full
Consumed5
Consumed6
Consumed7
Consumed8
Consumed9
fully Consumed
produced:10
produced:11
produced:12
produced:13
produced:14
Production Full
Consumed10
Consumed11
Consumed12
Consumed13
Consumed14
fully Consumed
produced:15
produced:16
produced:17
produced:18
produced:19
Production Full
Consumed15
Consumed16
Consumed17
Consumed18
Consumed19
fully Consumed
```

PROGRAM NO: 24

AIM:

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

ALGORITHM:

Step 1: Start

Step 2: Create a class named stack with data members as a(an array),top(set as -1),ch,item,i; a function named menu() .

Step 3: Inside menu(), give choices to push,pop and display the stack.

Step 4: If the choice is 1, then check whether the stack is full, else add an element into the stack.

Step 5: If the choice is 2, then check whether the stack is empty, else delete an element into the stack.

Step 6: If the choice is 3, then check whether the stack is empty, else print all the elements in the stack.

Step 7: If the choice is greater than 4, then print “Invalid option”.

Step 8: Inside the main(), create an object of type stack and call the menu() function.

Step 9: Stop

PROGRAM CODE:

co4_pg8.java	import java.util.*;
	class Stack <T>
	{
	ArrayList<T> S;
	int top=-1,size;
	Stack(int s)
	{
	this.size=s;
	this.S=new ArrayList<T>(size);
	}
	void push(T newData)
	{
	if(top+1 == size)
	{
	System.out.println("Stack overflow");
	}
	else

```
                {
                        top++;
                        if(S.size()>top)
                        {
                                S.set(top,newData);
                        }
                        else
                        {
                                S.add(newData);
                        }
                }
        }
        void pop()
        {
                if(top==-1)
                {
                        System.out.println("Stack Underflow");
                }
                else
                {
                        top--;
                }
        }
        void display()
        {
                for(int i=0;i<=top;i++)
                {
                        System.out.println(S.get(i));
                }
        }
        T top()
        {
                if(top==-1)
                {
                        System.out.println("Stack Underflow");
                        return null;
                }
                else
                {
                        return S.get(top);
                }
        }
}
public class co4_pg8
{
        public static void main(String args[])
        {
                Stack<Integer> obj=new Stack<>(5);
                obj.push(10);
                obj.push(20);
```

```
                obj.push(30);
                obj.push(40);
                obj.push(50);
                System.out.println("After Push");
                obj.display();
                obj.pop();
                obj.pop();
                obj.pop();
                System.out.println("After Pop");
                obj.display();
                System.out.println("Top");
                System.out.println(obj.top());
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
After Push
10
20
30
40
50
After Pop
10
20
Top
20
```

PROGRAM NO: 25

AIM:

Using generic method perform Bubble sort.

ALGORITHM:

Step 1: Start

Step 2: Read number of numbers(N) to sort.

Step 3: Read the numbers

Step 4: Repeat step 5 for i=0 to N-1

Step 5: Repeat for j=i+1 to N

Step 6: Check if array[i] >array[j],

Step 7: if Step 6 true,swap them.End of inner loop.End of outer loop.

Step 8: Print the sorted array

PROGRAM CODE:

co4_pg9.java	import java.util.Arrays;
	import java.util.Scanner;
	public class co4_pg9
	{
	public static void main(String args[])
	{
	Scanner sc=new Scanner(System.in);
	System.out.println("Enter the number of elements");
	int n=sc.nextInt();
	System.out.println("Enter the integer array");
	Integer array[]=new Integer[n];
	for(int i=0;i<n;i++)
	{
	array[i]=sc.nextInt();
	}
	Bubble<Integer> obj=new Bubble<>(array);
	Integer arraySort[]=obj.sort();
	System.out.println(Arrays.toString(arraySort));
	System.out.println("Enter the String array");
	sc.nextLine();
	String array2[]=new String[n];
	for(int i=0;i<n;i++)
	{

```
                    array2[i]=sc.nextLine();
                }
                Bubble<String> obj2=new Bubble<>(array2);
                String arraySort2[]=obj2.sort();
                System.out.println(Arrays.toString(arraySort2));
        }
}
class Bubble<T extends Comparable<? super T>>
{
        T array[];
        Bubble(T array[])
        {
                this.array=array;
        }
        T[] sort()
        {
                for(int i=0;i<array.length;i++)
                {
                        for(int j=0;j<(array.length)-1;j++)
                        {
                                if(array[j].compareTo(array[j+1])>0)
                                {
                                        T temp=array[j];
                                        array[j]=array[j+1];
                                        array[j+1]=temp;
                                }
                        }
                }
                return array;
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Enter the number of elements
5
Enter the integer array
2
1
5
4
3
[1, 2, 3, 4, 5]
Enter the String array
binu
arun
kevin
john
prad
[arun, binu, john, kevin, prad]
```

PROGRAM NO: 26

AIM:

Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

ALGORITHM:

Step 1: Start

Step 2: Create an object of the class ArrayList.

Step 3: Adding elements to the object of ArrayList using method add() and display.

Step 4: Remove elements of object of ArrayList using method remove() and display .

Step 5: Sort elements of object of ArrayList using method sort() and display.

Step 6: Getting object of list which is present at the specified index using method get() and display.

Step 7: Checking weather an element is present in list using method contains() and display True or False.

Step 8: Display the size of list using the method size().

Step 9: Clear List using the method clear().

PROGRAM CODE:

co4_pg10.java	import java.util.*;
	public class co4_pg10
	{
	public static void main(String args[])
	{
	ArrayList<String>s=new ArrayList<>();
	s.add("my");
	s.add("name");
	s.add("is");
	s.add("Manu");
	System.out.println("Array List");
	for(String st:s)
	{
	System.out.print(st+" ");
	}
	s.remove("Manu");
	System.out.println();
	System.out.println("Array List after removing Manu");
	for(String st:s)
	{
	System.out.print(st+" ");
	}
	Collections.sort(s);

```
                System.out.println("Array List after sorting");
                for(String st:s)
                {
                        System.out.print(st+" ");
                }
                s.clear();
                System.out.println();
                System.out.println("Array List after clearing"+s);
        }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Array List
my name is Manu
Array List after removing Manu
my name is Array List after sorting
is my name
Array List after clearing[]
```

PROGRAM NO: 27

AIM:

To remove all the elements from a linked list

ALGORITHM:

Step 1: Start

Step 2: Create an object of the class LinkedList.

Step 3: Adding elements to the linked list using method add().

Step 4: Remove all the elements of LinkedList using method clear().

Step 5: Display LinkedList.

PROGRAM CODE:

co4_pg11.java

```
import java.util.*;
public class co4_pg11
{
	public static void main(String args[])
	{
		LinkedList<String>l1=new LinkedList<>();
		l1.add("My");
		l1.add("name");
		l1.add("is");
		l1.add("Amal");
		System.out.println("Linked List");
		Iterator<String> itr=l1.iterator();
		while(itr.hasNext())
		{
			System.out.println(itr.next());
		}
		l1.clear();
		System.out.println();
		System.out.println("Linked List after clearing"+l1);
	}
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Linked List
My
name
is
Amal

Linked List after clearing[]
```

PROGRAM NO: 28

AIM:

To remove an object from the Stack when the position is passed as parameter.

ALGORITHM:

Step 1: Start

Step 2: Create an object of the class Stack.

Step 3: Adding elements to the stack using method add().

Step 4: Remove the element of stack at position ‘pos’ using method remove(pos).

Step 5: Display removed element and Stack

PROGRAM CODE:

co4_pg12.java

```
import java.util.*;
public class co4_pg12
{
  public static void main(String[] args)
  {
                Stack<String> st=new Stack<>();
                Scanner sc=new Scanner(System.in);
                String str;
                System.out.println("Enter the number of items");
                int n=sc.nextInt();
                sc.nextLine();
                System.out.println("Enter the elements");
                for(int i=0;i<n;i++)
    {
                        str=sc.nextLine();
                        st.push(str);
    }
                System.out.println(st);
                System.out.println("Enter the position of element to be
deleted");
                int sp=sc.nextInt();
                st.remove(sp);
                System.out.println(st);
  }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Enter the number of items
4
Enter the elements
jai
amal
how
are
[jai, amal, how, are]
Enter the position of element to be deleted
2
[jai, amal, are]
```

PROGRAM NO: 29

AIM:

To demonstrate the creation of queue object using the PriorityQueue class.

ALGORITHM:

Step 1: Start

Step 2: Create an object of the class PriorityQueue.

Step 3: Adding elements to the PriorityQueue using method add().

Step 5: Display PriorityQueue.

PROGRAM CODE:

co4_pg13.java

```
import java.util.*;
public class co4_pg13
{
        public static void main(String[] args)
  {
                int n;
                String str;
                PriorityQueue<String> pqueue=new PriorityQueue<>();
                System.out.println("Total count");
                Scanner sc=new Scanner(System.in);
                n=sc.nextInt();
                sc.nextLine();
                System.out.println("Enter data");
                for(int i=0;i<n;i++)
   {
                        str=sc.nextLine();
                        pqueue.add(str);
   }
                System.out.println("Peek: "+pqueue.peek());
                System.out.println("Queue");
                Iterator<String> itr1=pqueue.iterator();
                while(itr1.hasNext())
   {
                  System.out.println(itr1.next());
   }
                System.out.println("Polling: "+pqueue.poll());
                System.out.println("After polling data in Queue");
                Iterator<String> itr2=pqueue.iterator();
                while(itr2.hasNext())
   {
                  System.out.println(itr2.next());
   }
  }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Total count
4
Enter data
my
name
is
kiran
Peek: is
Queue
is
kiran
my
name
Polling: is
After polling data in Queue
kiran
name
my
```

PROGRAM NO: 30

AIM:

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

ALGORITHM:

Step 1: Start

Step 2: Create an object of the class ArrayDeque.

Step 3: Adding elements to the queue using method add().

Step 4: Removing elements of queue using method pop().

Step 5: Display Queue

PROGRAM CODE:

co4_pg14.java	import java.util.*;
	public class co4_pg14
	{
	public static void main(String[] args)
	{
	Deque<Integer> dq=new LinkedList<>();
	dq.add(1);
	dq.add(2);
	dq.addFirst(3);
	dq.addLast(4);
	dq.push(5);
	dq.offer(6);
	dq.offerFirst(7);
	System.out.print("DEQUE: "+dq+" ");
	dq.removeFirst();
	System.out.println("\nDEQUE after removing first element");
	System.out.print(dq+" ");
	dq.removeLast();
	System.out.println("\nDEQUE after removing last element");
	System.out.print(dq+" ");
	}
	}

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
DEQUE:  [7, 5, 3, 1, 2, 4, 6]
DEQUE after removing first element
[5, 3, 1, 2, 4, 6]
DEQUE after removing last element
[5, 3, 1, 2, 4]
```

PROGRAM NO: 31

AIM:

Program to demonstrate the creation of Set object using the LinkedHashset class

ALGORITHM:

Step 1: Start

Step 2: Create an object of the class LinkedHashset.

Step 3: Adding elements to the HashSet using method add().

Step 4: Display Linked HashSet.

PROGRAM CODE:

hash_set.java	package myproject;
	import java.util.*;
	public class hash_set
	{
	public static void main (String args[])
	{
	LinkedHashSet<String>
	hashset = new LinkedHashSet<String>();
	Scanner sc=new Scanner(System.in);
	System.out.println("enter the number ofletters");
	int n=sc.nextInt();
	System.out.println("enter the letters");
	for(int i=0;i<n;i++)
	{
	String s=sc.next();
	hashset.add(s);
	}
	System.out.println("\nOriginal LinkedHashSet:" + hashset);
	System.out.println("\nRemoving 'A' from LinkedHashSet: " + hashset.remove("A"));
	System.out.println("\nSize Of LinkedHashSet: " + hashset.size());
	System.out.println("\nChecking if 'B' is present=" + hashset.contains("B"));

```
    System.out.println("\nAfter Performing Operations, Final LinkedHashSet:" + hashset);
    System.out.println("\nAfter Iterating... ");
    for (String s : hashset)
     System.out.print(s + ", ");
    System.out.println();
  }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
enter the number ofletters
5
enter the letters
A
B
C
D
E

Original LinkedHashSet:[A, B, C, D, E]

Removing 'A' from LinkedHashSet: true

Size Of LinkedHashSet: 4

Checking if 'B' is present=true

After Performing Operations, Final LinkedHashSet: [B, C, D, E]

After Iterating..
B, C, D, E,
```

PROGRAM NO: 32

AIM:

Write a Java program to compare two hash set.

ALGORITHM:

Step 1: Start

Step 2: Create two objects of the class LinkedHashset.

Step 3: Adding elements to the two objects of LinkedHashSet using method add().

Step 4: Checking weather elements of first Hashset is present in second Hashset.

Step 5: If Step 4 is true print Yes, else print No.

PROGRAM CODE:

compare_hashset.java	package myproject;
	import java.util.*;
	public class compare_hashset
	{
	public static void main(String[] args)
	{
	Set<String> s1= new HashSet<String>();
	Set<String> s2 = new HashSet<String>();
	Scanner sc=new Scanner(System.in);
	System.out.println("Enter Number Of elements in s1: ");
	int n=sc.nextInt();
	System.out.println("\nEnter Elements of s1: ");
	for(int i =0;i<n;i++)
	{
	String st=sc.next();
	s1.add(st);
	}
	System.out.println("\nEnter Number Of elements in s2: ");
	int n1=sc.nextInt();
	System.out.println("\nEnter Elements of s2: ");
	for(int i =0;i<n1;i++)
	{

```
      String str=sc.next();
      s2.add(str);
    }
    System.out.println("\nHashSet 1: " + s1);
    System.out.println("\nHashSet 2: " + s2);
    //union
    Set<String> union = new HashSet<String>(s1);
    union.addAll(s2);
    System.out.print("\nUnion of the two Set");
    System.out.println(union);
    //intersection
    Set<String> intersection = new HashSet<String>(s1);
    intersection.retainAll(s2);
    System.out.print("\nIntersection of the two Set");
    System.out.println(intersection);
    //difference
    Set<String> difference = new HashSet<String>(s1);
    difference.removeAll(s2);
    System.out.print("\nDifference of the two Set");
    System.out.println(difference);
  }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
Enter Number Of elements in s1:
3

Enter Elements of s1:
5
9
4

Enter Number Of elements in s2:
4

Enter Elements of s2:
9
8
6
3

Hash Set 1: [4, 5, 9]

Hash Set 2: [3, 6, 8, 9]

Union of the two Set[3, 4, 5, 6, 8, 9]

Intersection of the two Set[9]

Difference of the two Set[4, 5]
```

PROGRAM NO: 33

AIM:

Program to demonstrate the working of Map interface by adding, changing, and removing elements.

ALGORITHM:

Step 1: Start

Step 2: Initialization of a Map using Generics.

Step 3: Adding values into map using method put() and display.

Step 4: Updating values using method put() by mentioning index of value and display.

Step 5: Removing values from map using method remove() and display.

PROGRAM CODE:

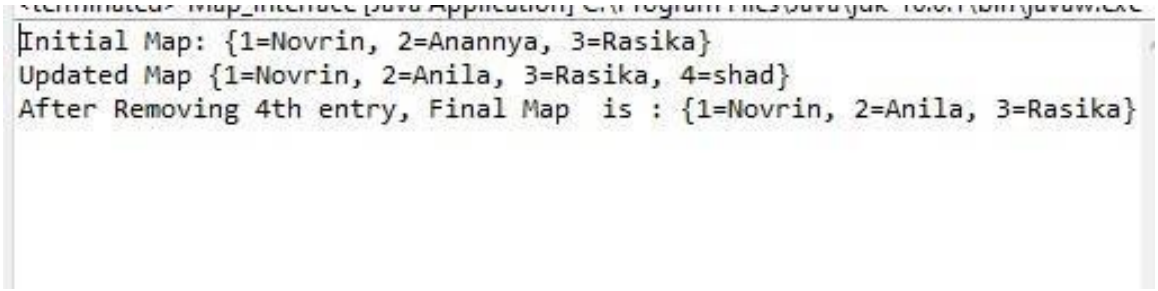
Map_interface.java	package myproject;
	import java.util.*;
	public class Map_interface
	{
	public static void main (String args[])
	{
	Map<Integer,String> hm=new HashMap<>();
	hm.put(1, "Novrin");
	hm.put(2, "Anannya");
	hm.put(3, "Rasika");
	System.out.println("Initial Map: "+ hm);
	hm.put((2), "Anila");
	hm.put((4), "shad");
	//Updating..
	System.out.println("Updated Map " + hm);
	//Removing..
	hm.remove(4);
	// Final Map..
	System.out.println("After Removing 4th entry, Final Map is : "+hm);
	}

	}

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 34

AIM:

Program to Convert HashMap to TreeMap.

ALGORITHM:

Step 1: Get the HashMap to be converted.

Step 2: Create a new TreeMap.

Step 3: Pass the hashMap to putAll() method of treeMap.

Step 4: Return the formed TreeMap.

PROGRAM CODE:

hash_to_treemap.java	package myproject;
	import java.util.*;
	public class hash_to_treemap
	{
	public static void main(String[] args)
	{
	HashMap<Integer,String>hMap=new HashMap<Integer,String>();
	hMap.put(10,"red");
	hMap.put(22, "green");
	hMap.put(3, "violet");
	hMap.put(44, "yellow");
	hMap.put(15, "black");
	System.out.println("HashMap Keys and Values: "+hMap);
	System.out.println("\n");
	TreeMap<Integer, String> tMap = new TreeMap<Integer, String>(hMap);
	System.out.println("TreeMap Keys and Values: " +tMap);
	}
	}

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
HashMap Keys and Values: {3=violet, 22=green, 10=red, 44=yellow, 15=black}

TreeMap Keys and Values: {3=violet, 10=red, 15=black, 22=green, 44=yellow}
```

COURSE OUTCOME 5

PROGRAM NO: 35

AIM:

Program to draw Circle, Rectangle, Line in Applet.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class ‘myfirstapplet’ that extends Applet class.

Step 3: Draw a line, rectangle and circle using drawLine, drawRect and drawOval methods of Graphics class respectively.

Step 4: Stop the program.

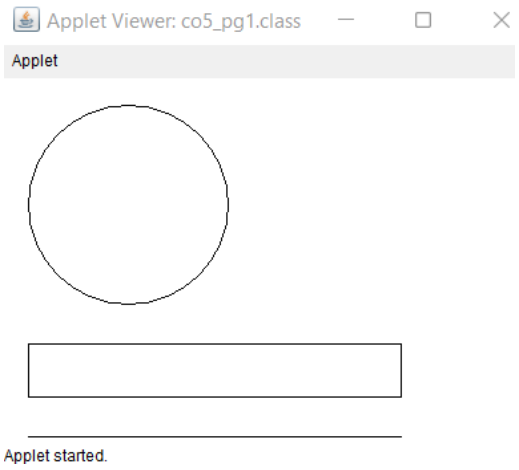
PROGRAM CODE:

co5_pg1.java	import java.applet.*;
	import java.awt.*;
	import java.awt.Graphics;
	public class co5_pg1 extends Applet
	{
	public void paint(Graphics g)
	{
	g.drawOval(20, 20, 150, 150);
	g.drawRect(20, 200, 280, 40);
	g.drawLine(20, 270, 300, 270);
	}
	}
	/*<applet height="700" code="co5_pg1.class" width="500" border="2"></applet>*/

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 36

AIM:

Program to find maximum of three numbers using AWT.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class ‘maxof3’ that extends Applet class and implements ActionListener interface.

Step 3: Using TextField class object, construct the required no. of textfields wide enough to hold the values entered by the user.

Step 4: Using Button class object, construct a labeled button that sends an instance of ActionEvent.

Step 5: Call addActionListener() method to send events from the button to the new listener.

Step 6: Get the string values from textfields and then parse them as integers.

Step 7: Compare each value using if-else statements to find the maximum value and set the result accordingly.

Step 8: Stop the program.

PROGRAM CODE:

maxNo.java	package OOPS;
	import java.awt.*;
	import java.awt.event.*;
	public class maxNo implements ActionListener
	{
	Button b1;
	TextField t1,t2,t3;
	Label l1,l2,l3,l4;
	Frame f;
	maxNo()
	{
	f = new Frame("-----Largest of 3 numbers-----");
	l1 = new Label("Enter First Number :");
	l1.setBounds(5, 50, 150, 30);
	f.add(l1);
	t1 = new TextField();
	t1.setBounds(200, 50, 150, 30);

```
f.add(t1);

t2 = new TextField();
t2.setBounds(200, 100, 150, 30);
f.add(t2);

l2 = new Label("Enter Second Number :");
l2.setBounds(5, 100, 150, 30);
f.add(l2);

t3 = new TextField();
t3.setBounds(200, 150, 150, 30);
f.add(t3);

l3 = new Label("Enter Third Number :");
l3.setBounds(5, 150, 150, 30);
f.add(l3);

l4 = new Label("Result :");
l4.setBounds(90, 200, 150, 30);
f.add(l4);

b1 = new Button("Check");
b1.setBounds(90, 250, 100, 30);
f.add(b1);

b1.addActionListener(this);

f.addWindowListener(new WindowAdapter()
{
   public void windowClosing(WindowEvent we)
   {
      System.exit(0);
   }
});
```

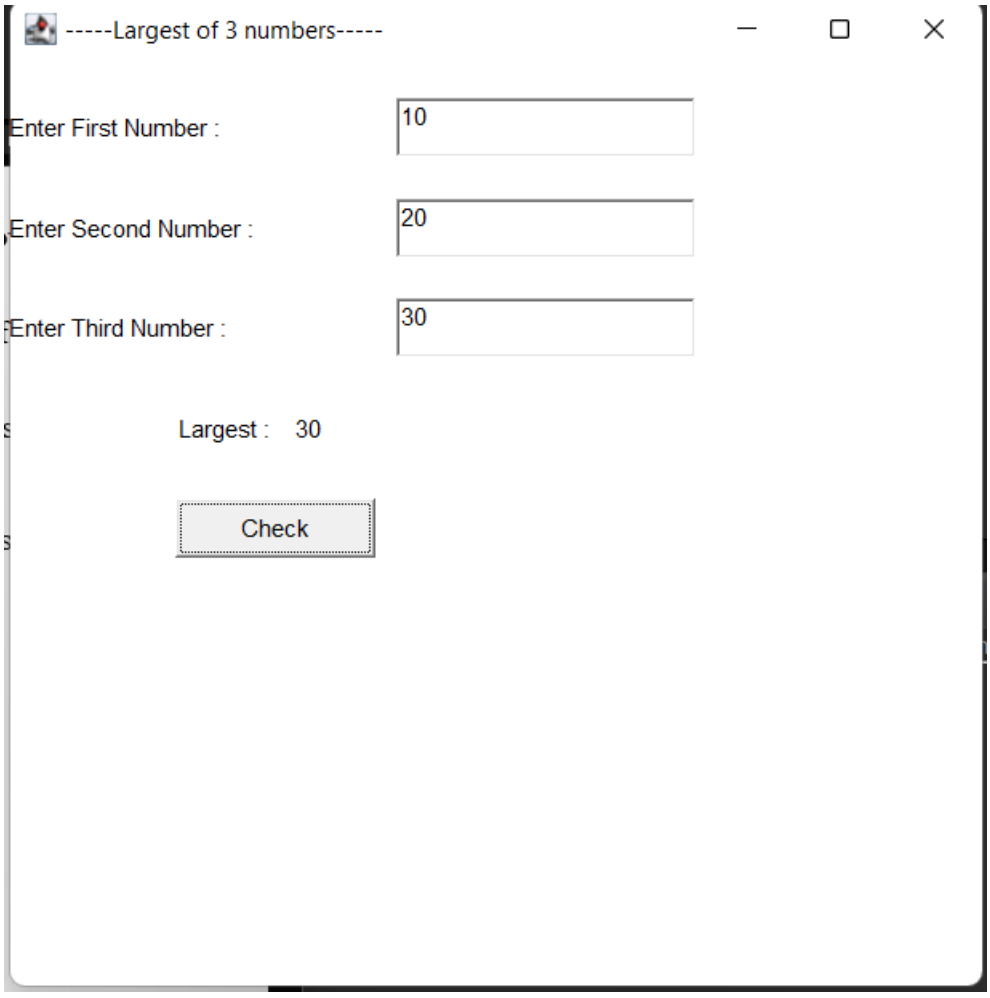
```
    f.setLayout(null);
    f.setSize(500, 500);
    f.setVisible(true);
        }
        public void actionPerformed(ActionEvent e)
        {

    int a = Integer.parseInt(t1.getText());
    int b = Integer.parseInt(t2.getText());
    int c = Integer.parseInt(t3.getText());
    int d = 0;

    if (e.getSource().equals(b1))
    {
        if (a>b && a>c)
        {
                l4.setText(String.valueOf("Largest :    " + a));
        }
        else if(b>a && b>c)
        {
                l4.setText(String.valueOf("Largest :    " + b));
        }
        else
        {
                l4.setText(String.valueOf("Largest :    " + c));
        }
    }

}

public static void main(String args[])
{
    maxNo m = new maxNo();
}
```

	}

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 37

AIM:

Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class 'Face' that extends Applet class and implements ActionListener interface.

Step 3: Using TextField class object, construct textfields to receive marks of 5 subjects from the user.

Step 4: Using Button class object, construct a labeled button that sends an instance of ActionEvent.

Step 5: Call addActionListener() method to send events from the button to the new listener.

Step 6: Get the string values from textfields and then parse them as float values.

Step 7: Calculate the percentage: Percent = ((mark1+mark2)*100)/200.

Step 8: Define a paint() method that contains functions from Graphics class to display a happy face if student secures above 50% or a sad face if otherwise.

Step 9: Stop the program.

PROGRAM CODE:

Faceapplet.java	import java.applet.*; import java.awt.*; import java.awt.event.*; import java.awt.Graphics; public class Faceapplet extends Applet implements ActionListener { Label l1=new Label("MARK 1:"); Label l2=new Label("MARK 2:"); Label l3=new Label("MARK 3:"); Label l4=new Label("Average:"); TextField t1=new TextField(); TextField t2=new TextField(); TextField t3=new TextField(); TextField t4=new TextField(); Button b=new Button("CHECK FACE"); public void init()

```
{
 l1.setBounds(60,100,100,30);
 l2.setBounds(60,140,100,25);
 l3.setBounds(60,180,100,25);
 l4.setBounds(60,220,100,25);
 t1.setBounds(200,100,180,30);
 t2.setBounds(200,140,100,25);
 t3.setBounds(200,180,100,25);
 t4.setBounds(200,220,100,25);
 b.setBounds(180,250,50,30);
 b.setBackground(Color.pink);
 b.setForeground(Color.blue);
 add(l1);
 add(l2);
 add(l3);
 add(l4);
 add(t1);
 add(t2);
 add(t3);
 add(t4);
 add(b);
 b.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
 int n1=Integer.parseInt(t2.getText());
 int n2=Integer.parseInt(t3.getText());
 int n3=Integer.parseInt(t3.getText());
 if(e.getSource()==b){
 int avg=(n1+n2+n3)/3;
 t4.setText(String.valueOf(avg));
 }
}
@Override
public void paint(Graphics g)
```

```
  {
   int n1= Integer.parseInt(t1.getText());
   int n2= Integer.parseInt(t2.getText());
   int n3= Integer.parseInt(t3.getText());
   int avg=(n1+n2+n3)/3;
   if(avg > 50)
   {
     g.setColor(Color.YELLOW);
     g.fillOval(10, 10, 200, 200);
     // draw Eyes
     g.setColor(Color.BLACK);
     g.fillOval(55, 65, 30, 30);
     g.fillOval(135, 65, 30, 30);
     // draw Mouth
     g.fillOval(50, 110, 120, 60);
     // adding smile
     g.setColor(Color.YELLOW);
     g.fillRect(50, 110, 120, 30);
     g.fillOval(50, 120, 120, 40);
   }
   else
   {
     g.setColor(Color.yellow);
     g.fillOval(0,0,300,300);
     g.setColor(Color.black );
     g.fillOval(80,75,30,30);//sad face
     g.fillOval(190,75,30,30);
     g.setColor(Color.black);
     g.drawArc(75,150,150,150,0,180);
     g.fillArc(75,150,150,150,0,180);
   }
 }
}
/*
<applet code="Faceapplet.class" width="400"
```

	height="400" border="2"> </applet> */

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

PROGRAM NO: 38

AIM:

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

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class ‘house’ that extends Applet and implements MouseListener.

Step 3: Define methods to add MouseListener to the panel.

Step 4: Using getX() and getY() methods, get the coordinates of the door to repaint when the MousePressed event occurs.

Step 5: Stop the program.

PROGRAM CODE:

MouseEvent_house.java	import java.awt.*;
	import java.applet.*;
	import java.awt.event.*;
	public class MouseEvent_house extends Applet implements MouseListener
	{
	int a,b;
	public void init()
	{
	addMouseListener(this);
	}
	public void paint(Graphics g)
	{
	int x[]={130,320,225};
	int y[]={150,150,25};
	g.drawPolygon(x,y,3);
	g.setColor(Color.gray);
	g.fillPolygon(x,y,3);
	g.drawRect(150,150,150,200);//House
	g.setColor(Color.green);
	g.fillRect(150,150,150,200);
	g.drawRect(200, 200,50,150);//Door

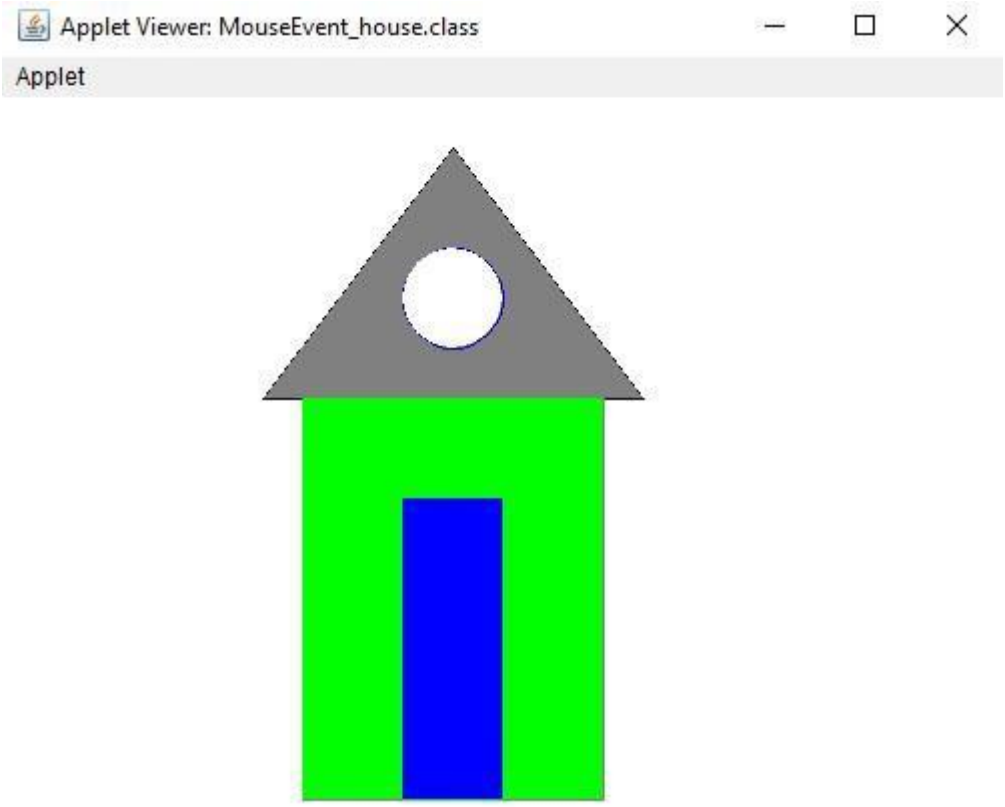
```
    g.setColor(Color.blue);
    g.fillRect(200,200,50,150);
    g.drawOval(200,75,50,50);
    g.setColor(Color.white);
    g.fillOval(200,75,50,50);
    if(a>200 && a<300 && b>200 && b<300)
    {
       g.setColor(Color.red);
       g.fillRect(200, 200, 50, 150);
    }
  }
  public void mouseClicked(MouseEvent e)
  {
  }
  public void mouseEntered(MouseEvent e)
  {
  }
  @Override
  public void mouseExited(MouseEvent e)
  {
  }
  public void mousePressed(MouseEvent e)
  {
    a=e.getX();
    b=e.getY();
    repaint();
  }
  public void mouseReleased(MouseEvent e)
  {
  }
 }
/*
<applet code="MouseEvent_house.class" width="500"
height="700" border="2">
</applet>
```

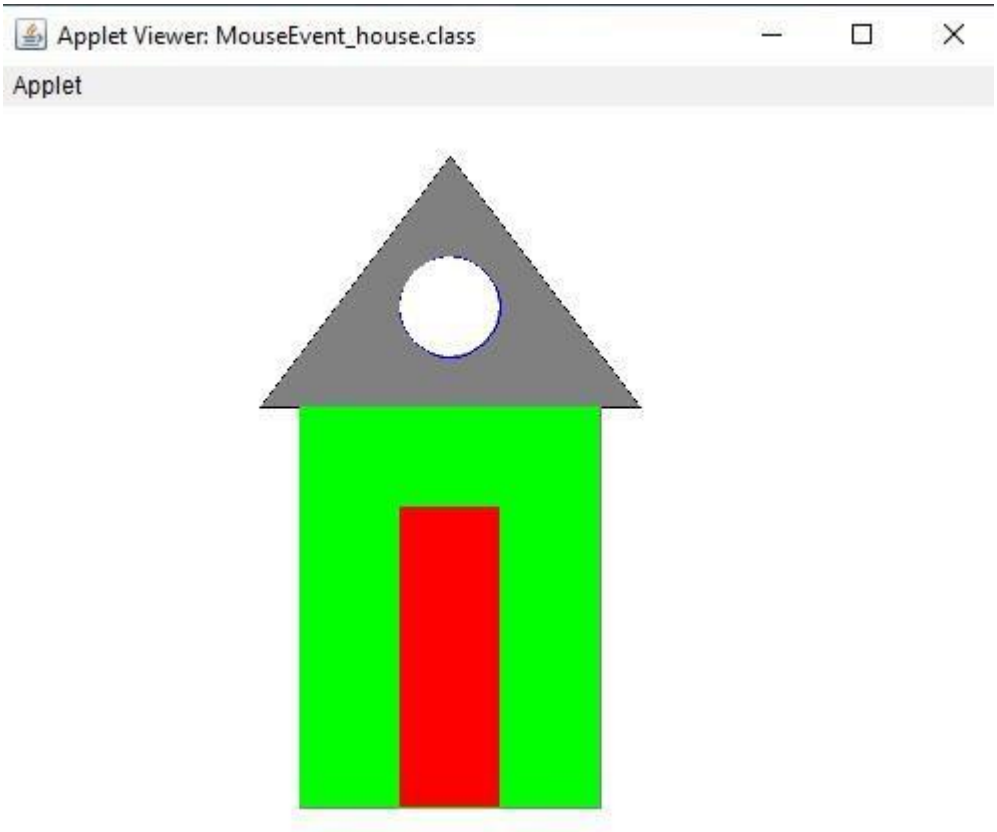
	*/

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:


Applet Viewer: MouseEvent_house.class
Applet


Applet Viewer: MouseEvent_house.class
Applet

PROGRAM NO: 39

AIM:

Implement a simple calculator using AWT components.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class 'calculator' that extends Frame and implements ActionListener interface.

Step 3: Using TextField class object, construct the required no. of textfields wide enough to hold the values entered by the user.

Step.4: Using Label class object, construct and provide the appropriate labels.

Step.5: Using Button class object, construct labeled buttons that send the instances of ActionEvent.

Step.6: Call addActionListener() method to send events from the button to the new listener.

Step.7: Get the string values from textfields and then parse them as integers.

Step.8: Perform various methods to add, subtract, multiply and divide those integers.

Step 9: Stop the program.

PROGRAM CODE:

calculator.java	import java.awt.*;
	import java.awt.event.*;
	class calculator implements ActionListener
	{
	//declaring object
	Frame f=new Frame();
	Label l1=new Label("fisrt number");
	Label l2=new Label("second number");
	Label l3=new Label("result");
	TextField t1=new TextField();
	TextField t2=new TextField();
	TextField t3=new TextField();
	Button b1=new Button("Add");
	Button b2=new Button("Mult");
	Button b3=new Button("Sub");
	Button b4=new Button("Div");
	Button b5=new Button("cancel");
	calculator()

```
{
  //giving coordinates
  l1.setBounds(50,100,100,20);
  l2.setBounds(50,150,100,20);
  l3.setBounds(50,190,100,20);
  t1.setBounds(200,100,100,20);
  t2.setBounds(200,150,100,20);
  t3.setBounds(200,190,100,20);
  b1.setBounds(50,250,50,20);
  b1.setBackground(Color.yellow);
  b2.setBounds(110,250,50,20);
  b2.setBackground(Color.yellow);
  b3.setBounds(170,250,50,20);
  b3.setBackground(Color.yellow);
  b4.setBounds(230,250,50,20);
  b4.setBackground(Color.yellow);
  b5.setBounds(290,250,50,20);
  b5.setBackground(Color.yellow);
  //adding to frame
  f.add(l1);
  f.add(l2);
  f.add(l3);
  f.add(t1);
  f.add(t2);
  f.add(t3);
  f.add(b1);
  f.add(b2);
  f.add(b3);
  f.add(b4);
  f.add(b5);
  b1.addActionListener(this);
  b2.addActionListener(this);
  b3.addActionListener(this);
  b4.addActionListener(this);
  b5.addActionListener(this);
```

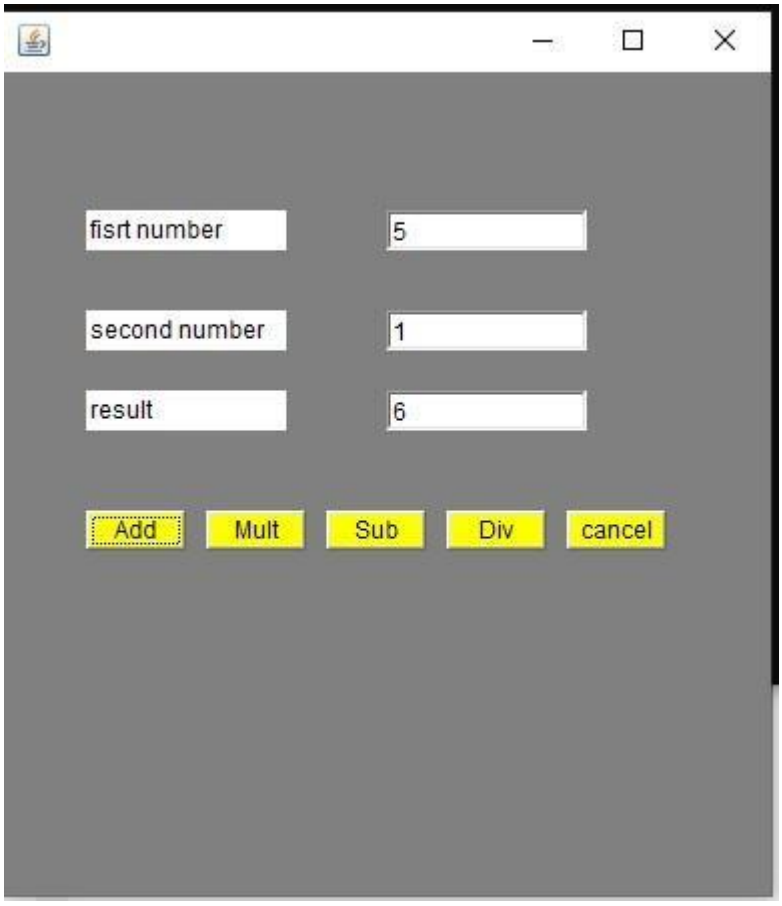
```
 f.setLayout(null);
 f.setVisible(true);
 f.setSize(400,450);
 f.setLocation(500,200);
 f.setBackground(Color.gray);
}
public void actionPerformed(ActionEvent e)
{
 int n1=Integer.parseInt(t1.getText());
 int n2=Integer.parseInt(t2.getText());
 if(e.getSource()==b1)
 {
  t3.setText(String.valueOf(n1+n2));
 }
 if(e.getSource()==b3)
 {
    t3.setText(String.valueOf(n1-n2));
 }
 if(e.getSource()==b2)
 {
   t3.setText(String.valueOf(n1*n2));
 }
 if(e.getSource()==b4)
 {
    t3.setText(String.valueOf(n1/n2));
 }
 if(e.getSource()==b5)
 {
   System.exit(0);
 }
}
public static void main(String args[])
{
  new calculator();
}
```

	}

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



PROGRAM NO: 40

AIM:

Develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user's choice.

ALGORITHM:

Step 1: Start the program.

Step 2: Define an interface 'shapes' that extends Applet class and implements ItemListener interface.

Step 3: Declare a new constructor of the Choice class to create an empty Choice menu.

Step 4: Use add() method to include items in the menu.

Step 5: Using getSelectedItem() method, get the item chosen by the user from the menu and repaint accordingly

Step 6: Stop the program.

PROGRAM CODE:

Choice_pgm.java	import java.applet.*;
	import java.awt.*;
	import java.awt.Graphics;
	import java.awt.event.*;
	public class Choice_pgm extends Applet implements
	ItemListener
	{
	Choice choice;
	int c;
	public void init()
	{
	choice = new Choice();
	choice.addItem("Shapes");
	choice.addItem("RECTANGLE");
	choice.addItem("SQUARE");
	choice.addItem("CIRCLE");
	choice.addItem("TRIANGLE");
	add(choice);
	choice.addItemListener(this);

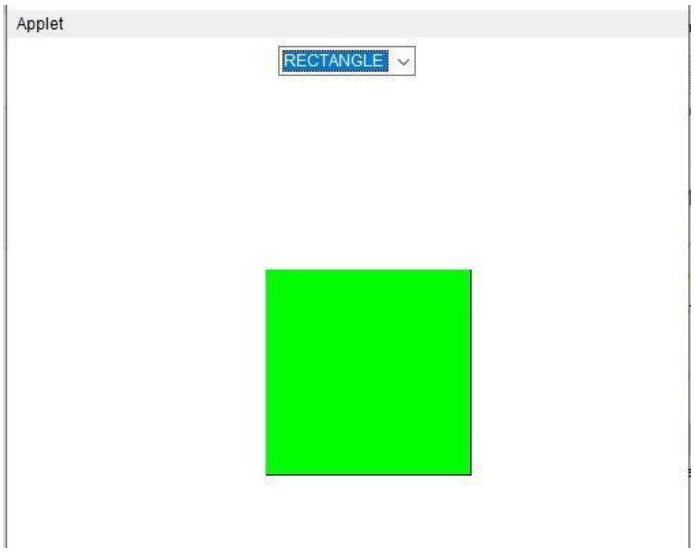
```
}
public void itemStateChanged (ItemEvent e)
{
  c= choice.getSelectedIndex();
  repaint();
}
public void paint(Graphics g)
{
  super.paint(g);
  if (c == 1)
  {
    g.drawRect(190,170,150,150);
    g.setColor(Color.green);
    g.fillRect(190,170,150,150);
  }
  if (c == 2)
  {
    g.drawRect(200,200,50,50);
    g.fillRect(200,200,50,50);
  }
  if (c == 3)
  {
    g.drawOval(180,180,100,100);
    g.setColor(Color.yellow);
    g.fillOval(180,180,100,100);
  }
  if (c ==4)
  {
    int[] x={120,210,0};
    int[] y={0,210,210};
    g.drawPolygon(x,y,3);
    g.setColor(Color.blue);
    g.fillPolygon(x,y,3);
  }
}
```

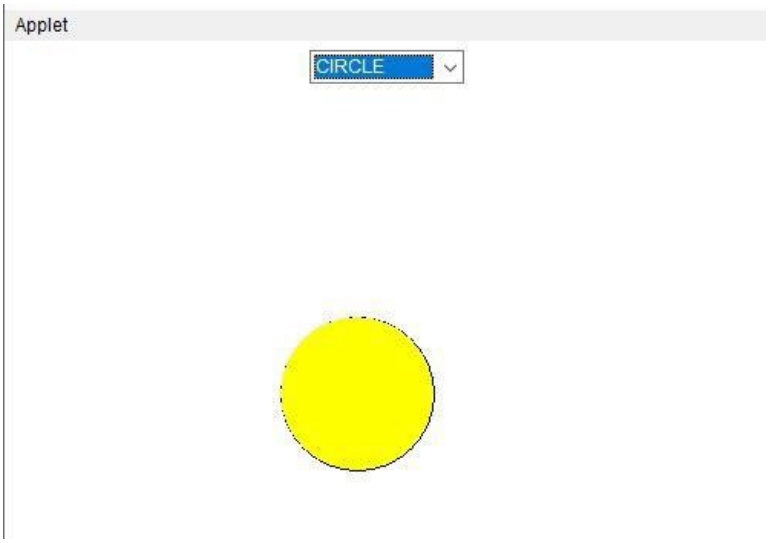
	} /* <applet code="Choice_pgm.class" width="500" height="700" border="2"> </applet> */

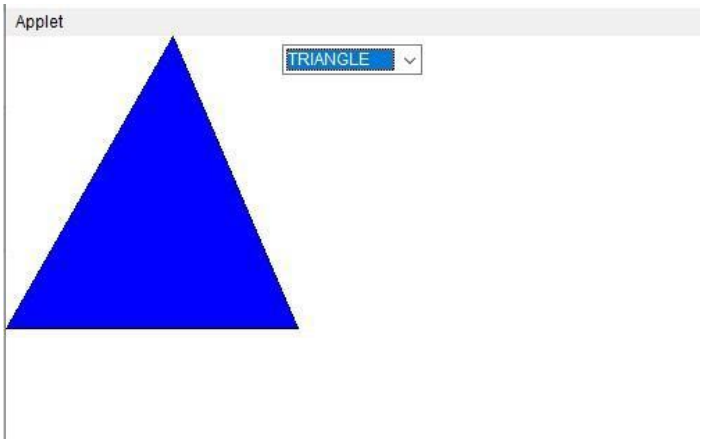
RESULT:

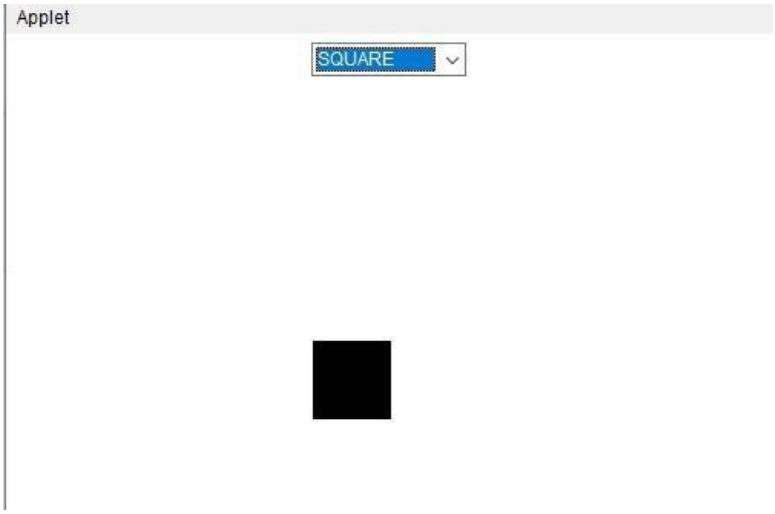
The above program is successfully executed and obtained the output.

OUTPUT:






Applet
TRIANGLE


Applet
SQUARE

PROGRAM NO: 41

AIM:

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

ALGORITHM:

Step 1: Start the program

Step 2: Define a class MouseDemo that extends Applet class and implements MouseListener interface.

Step 3: Define methods to add MouseListener to the panel.

Step 4: Using getX() and getY() methods, get the location (or movements) of mouse pointer on the panel. Use them to display the necessary message in the output.

Step 5: Define another class WindowEvents that extends Applet class and implements WindowListener interface.

Step 6: Define methods to add WindowListener to the panel.

Step 7: Display the appropriate message in the output.

Step 8: Stop the program.

PROGRAM CODE:

events.java	import java.awt.*;
	import java.applet.*;
	import java.awt.event.*;
	public class events extends Applet implements
	MouseListener,MouseMotionListener
	{
	int mx=0;
	int my=0;
	String msg="";
	public void init()
	{
	addMouseListener(this);
	addMouseMotionListener(this);
	}
	public void mouseClicked(MouseEvent me)
	{
	mx=20;
	my=40;

```
msg="Mouse Clicked";
repaint();
}
public void mousePressed(MouseEvent me)
{
mx=30;
my=60;
msg="Mouse Pressed";
repaint();
}
public void mouseReleased(MouseEvent me)
{
mx=30;
my=60;
msg="Mouse Released";
repaint();
}
public void mouseEntered(MouseEvent me)
{
mx=40;
my=80;
msg="Mouse Entered";
repaint();
}
public void mouseExited(MouseEvent me)
{
mx=40;
my=80;
msg="Mouse Exited";
repaint();
}
public void mouseDragged(MouseEvent me)
{
mx=me.getX();
my=me.getY();
```

```
showStatus("Currently mouse dragged"+mx+" "+my);
repaint(); }
public void mouseMoved(MouseEvent me)
{
mx=me.getX();
my=me.getY();
showStatus("Currently mouse is at"+mx+" "+my);
repaint();
}
public void paint(Graphics g)
{
g.drawString("Handling Mouse Events",30,20);
g.drawString(msg,60,40);
g.setColor(Color.red);
}
}
/*<applet code="events" width=300 height=300>
</applet>*/
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

PROGRAM NO: 42

AIM:

Develop a program to handle Key events.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class keys that extends Applet and implements KeyListener.

Step.3: Define methods to add KeyListener to the panel which will have the following methods: void keyTyped(KeyEvent e) – Invoked when a key has been typed. void keyPressed(KeyEvent e) - Invoked when a key has been pressed. void keyReleased(KeyEvent e) - Invoked when a key has been released.

Step.4: Using getKeyChar(), get the unicode and character representation of the key pressed. Use them to display the necessary message in the output.

Step.5: Stop the program.

PROGRAM CODE:

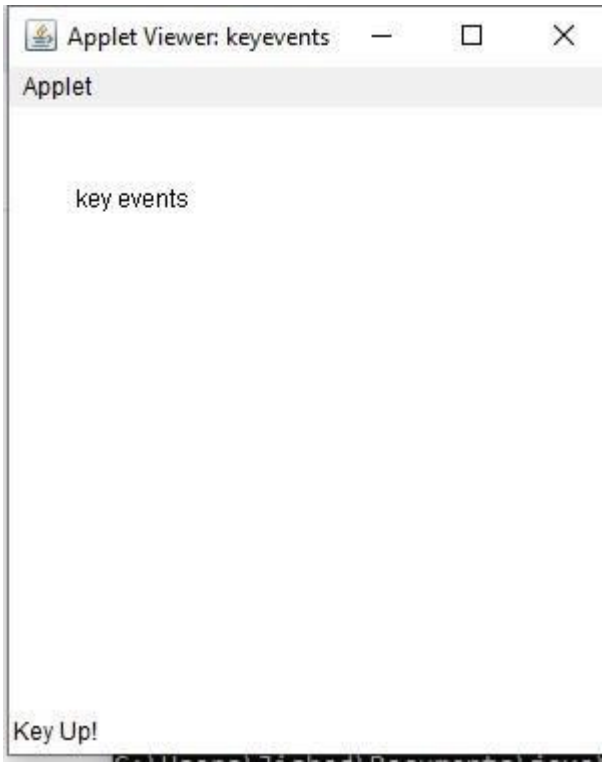
keyevents.java	import java.awt.event.*;
	import java.applet.*;
	/*<applet code="keyevents" width=300
	height=300></applet>*/
	public class keyevents extends Applet implements
	KeyListener
	{
	String msg=" ";
	int x=30,y=50;
	public void init()
	{
	addKeyListener(this);
	requestFocus();
	}
	public void keyTyped(KeyEvent ke)
	{
	msg+=ke.getKeyChar();
	repaint();
	}
	public void keyReleased(KeyEvent ke)

```
{
showStatus("Key Up!");
}
public void keyPressed(KeyEvent ke)
{
showStatus("Key Down!");
}
public void paint(Graphics G)
{
G.drawString(msg,x,y);
}
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:



COURSE OUTCOME 6

PROGRAM NO: 43

AIM:

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

ALGORITHM:

Step 1: Start the program

Step 2: Create a class named ‘FilesList’ that implements FilenameFilter interface.

Step 3: Create an object for the class File to to initialize its constructor with the file source.

Step 4: Using list(), get the names of all the files present in the directory.

Step 5: Create an object for the FileNameFilter interface that contains the method Boolean accept (File dir, String name) to test if a specified file should be included in the file list or not.

Step 6: Filter accordingly and store the file names to the list.

Step 7: Display the list.

Step 8: Stop the program

PROGRAM CODE:

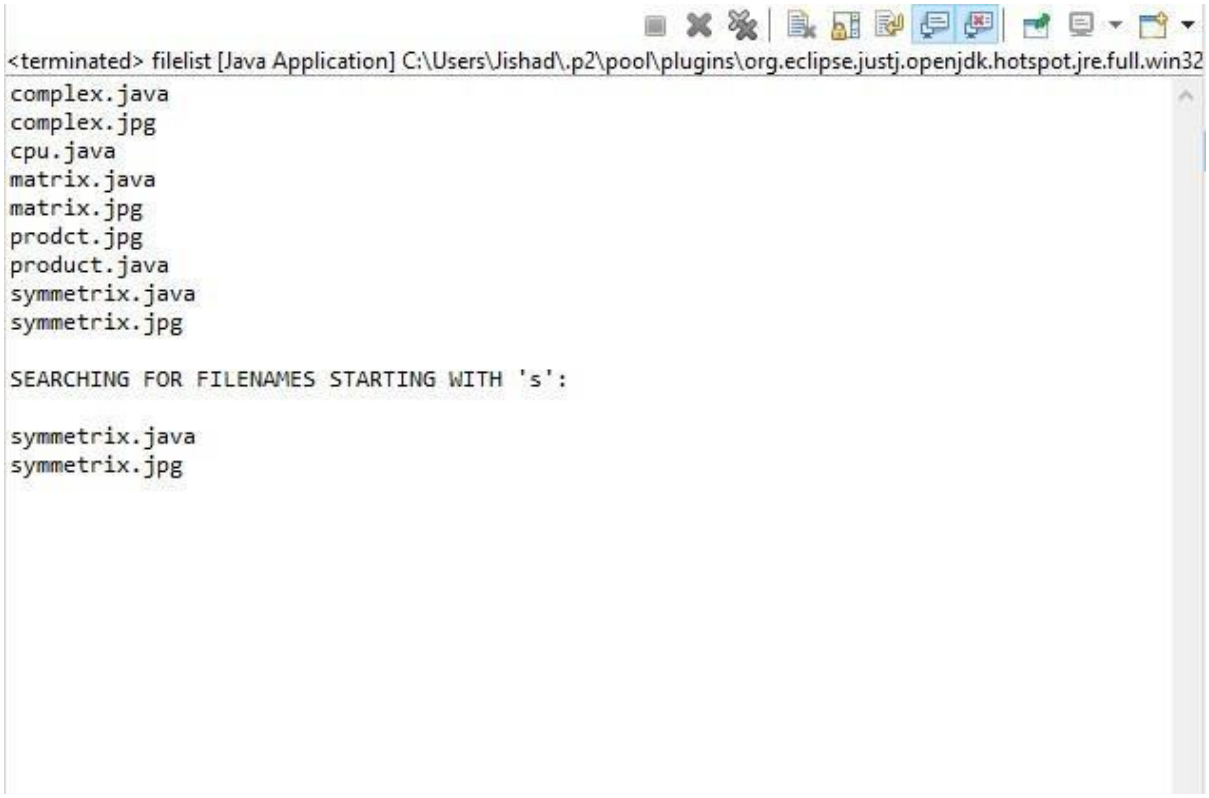
filelist.java	package myproject;
	import java.io.File;
	import java.io.*;
	import java.util.*;
	public class filelist
	{
	public static void main(String[] args)
	{
	File file = new File("C:\\Users\\AJIL\\Documents\\java\\cycle1 op");
	String[] list = file.list();
	for(String str : list)
	{
	System.out.println(str);
	}
	System.out.println("\nSEARCHING FOR FILENAMES STARTING WITH 's':\n");

```
  FilenameFilter filter = new FilenameFilter()
  {
    public boolean accept(File dir, String fname)
    {
      return fname.startsWith("s");
    }
  };
  String[] search = file.list(filter);
  if(search == null)
  {
    System.out.println("File does not exist. .");
  }
  else
  {
    for(int i=0; i<search.length;i++)
    {
      String fn = search[i];
      System.out.println(fn);
    }
  }
 }
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:


<terminated> filelist [Java Application] C:\Users\Jishad\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32
complex.java
complex.jpg
cpu.java
matrix.java
matrix.jpg
prodct.jpg
product.java
symmetrix.java
symmetrix.jpg
SEARCHING FOR FILENAMES STARTING WITH 's':
symmetrix.java
symmetrix.jpg

PROGRAM NO: 44

AIM:

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

ALGORITHM:

Step 1: Start the program.

Step 2: Create a class named ‘FileReadWrite’.

Step 3: Create an object of the class File to initialize its constructor with the file source.

Step 4: Create and use an object for the FileWriter class to write the file.

Step 5: Create and use an object for the BufferedReader class to read the stream of characters the specified file.

Step 6: Display the contents read from the file on the console.

Step 7: Stop the program

PROGRAM CODE:

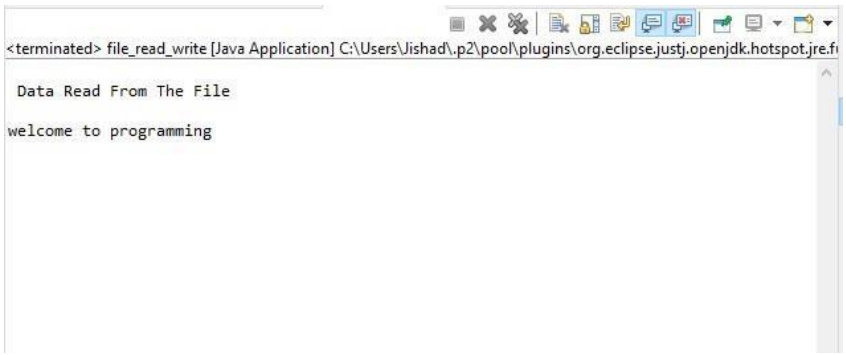
File_read_write.java	package myproject;
	import java.io.BufferedReader;
	import java.io.FileReader;
	import java.io.FileWriter;
	import java.io.IOException;
	public class file_read_write
	{
	public static void main(String[] args)
	{
	try
	{
	FileWriter fw=new FileWriter("C:\\Users\\AJIL\\Documents\\java\\cycle6\\sample",true);
	fw.write("welcome to programming");
	fw.close();
	FileReader reader = new FileReader("C:\\Users\\AJIL\\Documents\\java\\cycle6\\sample");
	BufferedReader b= new BufferedReader(reader);
	String line;

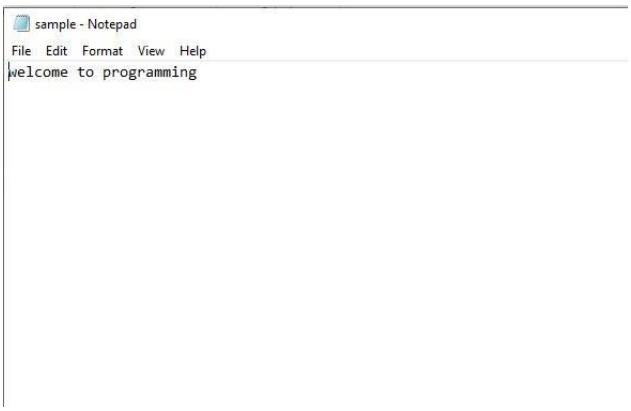
```
System.out.println("\n Data Read From The File \n");
while ((line = b.readLine()) != null)
{
System.out.println(line);
}
reader.close();
}
catch (IOException e)
{
System.out.println("\n Error Occured...");
}
}
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:





PROGRAM NO: 45

AIM:

Write a program to copy one file to another.

ALGORITHM:

Step 1: Start the program.

Step 2: Create a class named ‘copy’.

Step 3: Create and use an object for the BufferedReader class to read the stream of characters from the specified file.

Step 4: Create and use an object for the FileWriter class to write the stream of characters read by the BufferedReader, to the file. while ((s = br.readLine()) != null) { fw.write(s); }

Step 5: Display the appropriate message on the console.

Step 6: Stop the program

PROGRAM CODE:

copy_file.java	package myproject;
	import java.io.*;
	import java.util.*;
	public class copy_file
	{
	public static void main(String args[]) throws Exception
	{
	Scanner sc= new Scanner(System.in);
	System.out.println("enter the first file:");
	String file1=sc.next();
	System.out.println("enter the Second file:");
	String file2 =sc.next();
	sc.close();
	FileReader fin = new FileReader(file1);
	FileWriter fout = new FileWriter(file2, true);
	int c;
	while ((c = fin.read()) != -1)
	{
	fout.write(c);
	}

	System.out.println("copy file1 to file2 ");
	fin.close();
	fout.close();
	}
	}

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:

```
PS C:\Users\USER\Desktop\Qs3> javac copy_file.java
PS C:\Users\USER\Desktop\Qs3> java copy_file
enter the first file:
file1.txt
enter the Second file:
file2.txt
copy file1 to file2
PS C:\Users\USER\Desktop\Qs3>
```

PROGRAM NO: 46

AIM:

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

ALGORITHM:

Step 1: Start the program

Step 2: Create a class named ‘CopySep’

Step 3: Create an object for the class File to initialize its constructor with the given file.

Step 4: Get user inputs via the console, for the integers to be inserted into the file.

Step 5: Using an object for the FileWriter class, write those integers into the file.

Step 6: Using objects for the FileOutputStream class, create two separate files to store even and odd integers respectively and copy the integers accordingly to separate files just created.

Step 7: Stop the program.

PROGRAM CODE:

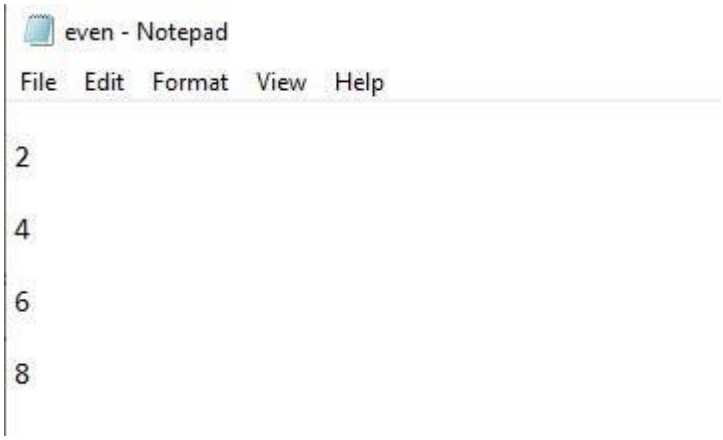
copy_evnod.java	package myproject;
	import java.io.File;
	import java.io.FileInputStream;
	import java.io.FileOutputStream;
	import java.io.IOException;
	import java.util.Scanner;
	public class copy_evnod
	{
	public static void main(String args[]) throws IOException
	{
	FileInputStream fr = new FileInputStream("C:\\Users\\AJIL\\Documents\\java\\cycle6\\integer.txt");
	FileOutputStream fw1 = new FileOutputStream("C:\\Users\\AJIL\\Documents\\java\\cycle6\\even.txt");
	FileOutputStream fw2 = new FileOutputStream("C:\\Users\\AJIL\\Documents\\java\\cycle6\\odd.txt");
	System.out.println("\nFrom the file 'integers.txt' ,Odd Numbers are copied to 'odd.txt'file and Even numbers are copied to 'even.txt' file\n");
	int i;
	while((i=fr.read()) != -1)

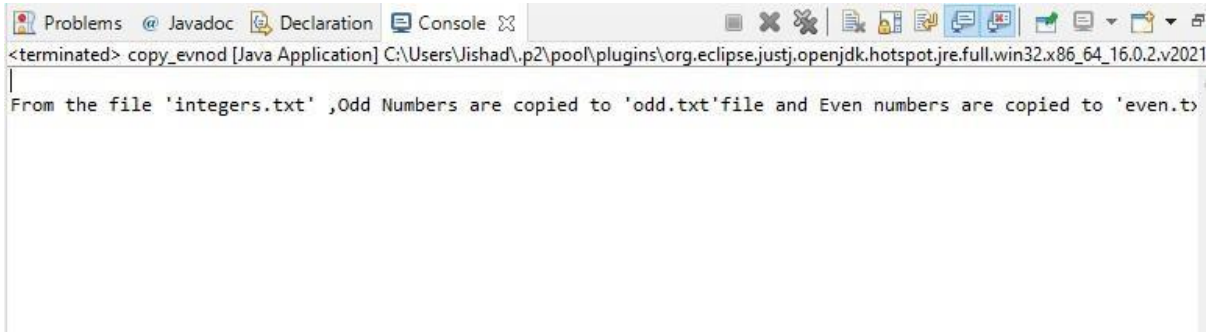
	{
	if(i%2==0)
	fw1.write(i);
	Else
	fw2.write(i);
	}
	fr.close();
	fw1.close();
	fw2.close();
	}
	}

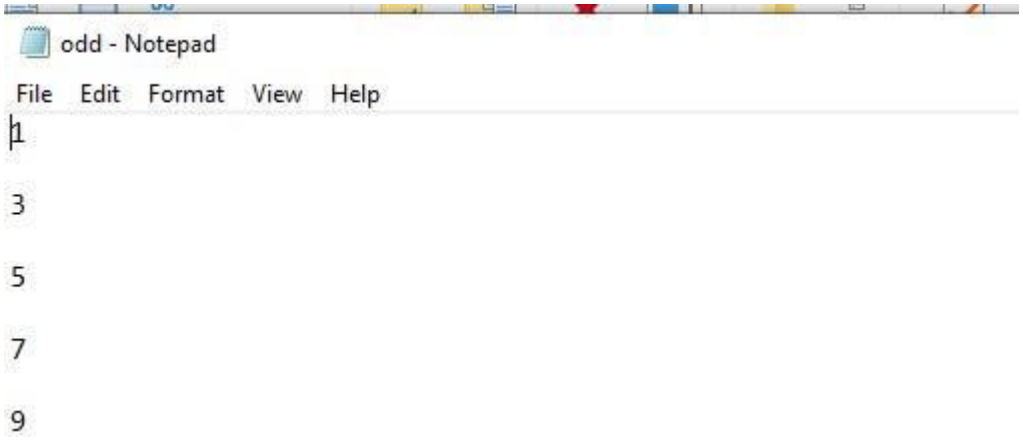
RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:


even - Notepad
File Edit Format View Help
2
4
6
8


Problems Javadoc Declaration Console
<terminated> copy_evnod [Java Application] C:\Users\Jishad\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v2021
From the file 'integers.txt' ,Odd Numbers are copied to 'odd.txt'file and Even numbers are copied to 'even.t


odd - Notepad
File Edit Format View Help
1
3
5
7
9

PROGRAM NO: 47

AIM:

Client server communication using Socket – TCP/IP.

ALGORITHM:

Step 1: Start
Step 2: To create the Client application, create an instance of ClientSocket class.
Step 3: To create the Server application, create an instance of ServerSocket class.
Step 4: Stop

PROGRAM CODE:

client.java

```
package myproject;
import java.net.*;
import java.io.*;
public class client
{
public static void main(String args[]) throws Exception
{
try
{
Socket sk = new Socket("localhost", 1234);
PrintWriter pw = new PrintWriter(sk.getOutputStream(), true);
pw.println("HELLO SERVER ..!!!!!");
//Client is reading from its InputStream
InputStreamReader isr = new InputStreamReader(sk.getInputStream());
BufferedReader br = new BufferedReader(isr);
String str=br.readLine();
System.out.println("MESSAGE FROM SERVER: "+str);
pw.close();
sk.close();
}
catch(Exception e)
{
System.out.println("An error occured.." +e);
}
}
```

	}

server.java	package myproject;
	import java.net.*;
	import java.io.*;
	public class server
	{
	public static void main(String[] args) throws Exception
	{
	try
	{
	ServerSocket ss = new ServerSocket(1234);
	System.out.println("SERVER IS WAITING FOR THE CLIENT.....");
	Socket sk = ss.accept();
	System.out.println("CONNECTION ESTABLISHED !!!");
	InputStreamReader isr = new InputStreamReader(sk.getInputStream());
	BufferedReader br = new BufferedReader(isr);
	String str = br.readLine();
	System.out.println("MESSAGE FROM CLIENT: "+str);
	//Server is responding through its OutputStream
	PrintWriter pw = new PrintWriter(sk.getOutputStream(), true);
	pw.println("HI CLIENT....");
	pw.close();
	}
	catch(Exception e)
	{
	System.out.println("An error occured.."+e);
	}
	}
	}

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT:


server [Java Application] C:\Users\Jishad\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.
SERVER IS WAITING FOR THE CLIENT.....


<terminated> client [Java Application] C:\Users\Jishad\.p2\pool\plugins\org.eclipse.justj.openjdk
MESSAGE FROM SERVER: HI CLIENT....

PROGRAM NO: 48

AIM:

Client Server communication using DatagramSocket – UDP.

ALGORITHM:

Step 1: Start

Step 2: Create the Client application

Step 3: Create the Server application

Step 4: Stop

PROGRAM CODE:

client_udp.java	package myproject;
	import java.io.*;
	import java.net.*;
	public class client_udp
	{
	public static void main(String[] args) throws IOException
	{
	DatagramSocket client= new DatagramSocket();
	InetAddress add=InetAddress.getByName("localhost");
	String str ="Hello...Server";
	byte[] bufBytes = str.getBytes();
	DatagramPacket datagramPacket=new DatagramPacket(bufBytes,bufBytes.length,add,1234);
	client.send(datagramPacket);
	client.close();
	}
	}

server_udp.java	package myproject; import java.io.*; import java.net.*; public class server_udp { public static void main(String[] args) throws IOException {

```
DatagramSocket server=new DatagramSocket(1234);
byte[] buf=new byte[256];
DatagramPacket packet=new DatagramPacket(buf,buf.length);
server.receive(packet);
String reply =new String(packet.getData());
System.out.println("\n Client Says : "+reply);
server.close();
}
}
```

RESULT:

The above program is successfully executed and obtained the output.

OUTPUT: