20MCA132 - OBJECT ORIENTED PROGRAMMING LAB

RECORD

ASHISH WILSON
ROLL No :28
S2-REG-MCA-A

TABLE OF CONTENTS

Course Outcome 1 (CO1):

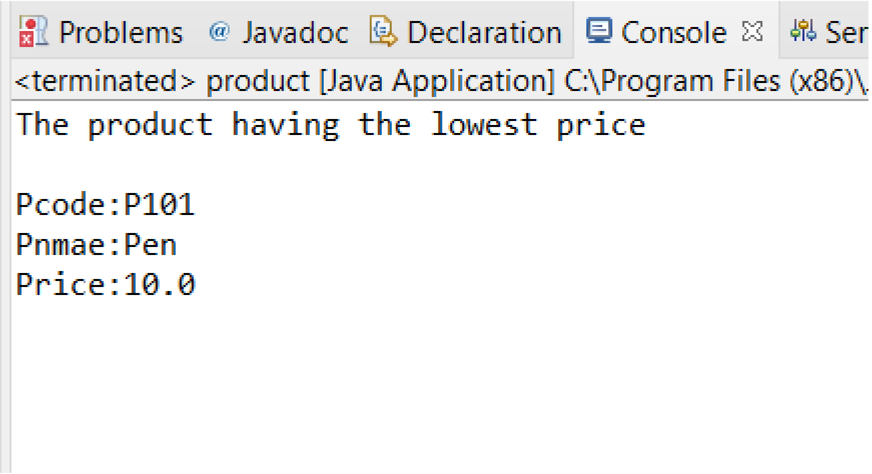
1. **Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.**

PROGRAM

Product.java

```
public class product{
        String pcode,pname; float price;

        public void setdata(String a,String b,float c)
        {
                pcode=a; pname=b; price=c;

        }
        void display()
        {
                System.out.println("\nPcode:"+pcode+"\nPnmae:"+pname+"\nPrice:"+price);
        }

        public static void main(String[] args) {
        product obj1=new product();
        product obj2=new product(); product obj3=new product(); obj1.setdata("P100","Soap",50);
        obj2.setdata("P101","Pen",10); obj3.setdata("P102","Bodywash",100);
        System.out.println("The product having the lowest price");
        if((obj1.price<obj2.price) &&(obj1.price<obj3.price))
        {
                obj1.display();
        }
        else if(obj2.price<obj3.price)
        {


        }
        else
        {

        }
        obj2.display();
        obj3.display();

        }
}
```

OUTPUT



```
<terminated> product [Java Application] C:\Program Files (x86)\.
The product having the lowest price

Pcode:P101
Pnmae:Pen
Price:10.0
```

RESULT

The program has been executed and the output was verified.

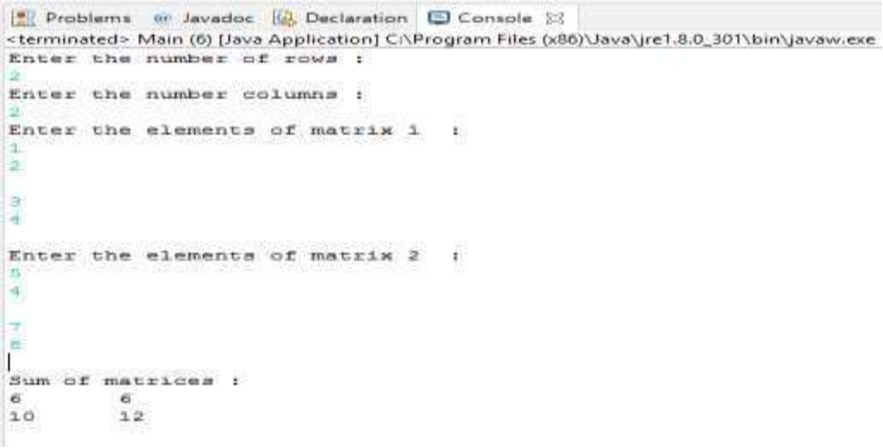
2. Read 2 matrices from the console and perform matrix addition.

PROGRAM

AddMatrix.java

```
import java.util.Scanner;

public class AddMatrix {

  public static void main(String args[])
  {
    int row, col,i,j;
    Scanner in = new Scanner(System.in);

    System.out.println("Enter the number of rows : ");
    row = in.nextInt();

    System.out.println("Enter the number columns : ");
    col = in.nextInt();

    int mat1[][] = new int[row][col];
    int mat2[][] = new int[row][col];
    int res[][] = new int[row][col];

    System.out.println("Enter the elements of matrix 1 : ");

    for ( i= 0 ; i < row ; i++ )
    {
        for ( j= 0 ; j < col ;j++ )
            mat1[i][j] = in.nextInt();

        System.out.println();
    }
    System.out.println("Enter the elements of matrix 2 : ");

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

        for ( j= 0 ; j < col ;j++ )
            mat2[i][j] = in.nextInt();

        System.out.println();
    }

    for ( i= 0 ; i < row ; i++ )
        for ( j= 0 ; j < col ;j++ )
            res[i][j] = mat1[i][j] + mat2[i][j] ;

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

    for ( i= 0 ; i < row ; i++ )
    {
        for ( j= 0 ; j < col ;j++ )
            System.out.print(res[i][j]+"\t");

        System.out.println();
    }

  }

}
```

OUTPUT



```
<terminated> Main (6) [Java Application] C:\Program Files (x86)\Java\jre1.8.0_301\bin\javaw.exe
Enter the number of rows :
2
Enter the number columns :
2
Enter the elements of matrix 1  :
1
2

3
4

Enter the elements of matrix 2  :
5
4

7
8

Sum of matrices :
6       6
10      12
```

RESULT

The program has been executed and the output was verified.

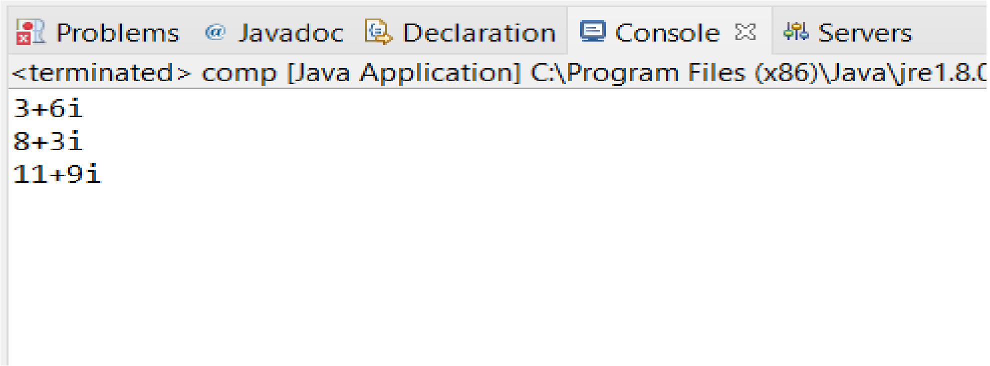
3. Add complex numbers .

PROGRAM

Comp.java

```
public class comp {
int x;
int y;
void get(int a,int b)
{
x=a; y=b;
}
void show()
{
System.out.println(x+"+"+y+"i");
}

public static void main(String[] args) {
comp obj1=new comp();
comp obj2=new comp(); comp obj3=new comp(); obj1.get(3,6);
obj2.get(8,3);
obj1.show();
obj2.show(); obj3.x=obj1.x + obj2.x; obj3.y=obj1.y + obj2.y; obj3.show();
}
}
```

OUTPUT



```
<terminated> comp [Java Application] C:\Program Files (x86)\Java\jre1.8.(
3+6i
8+3i
11+9i
```

RESULT

The program has been executed and the output was verified.

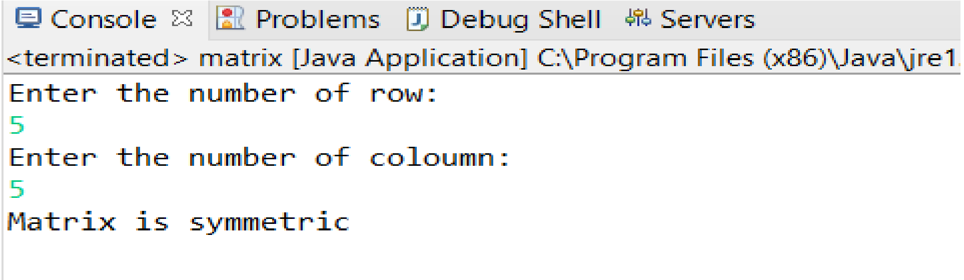
4. **Read a matrix from the console and check whether it is symmetric or not.**

PROGRAM

Matrix.java

```
import java.util.*;
public class matrix {
public static void main(String[] args) {
Scanner ip=new Scanner(System.in);
System.out.println("Enter the number of row: ");
int row=ip.nextInt();
System.out.println("Enter the number of coloumn: ");
int col=ip.nextInt();
if(row==col)
{
System.out.println("Matrix is symmetric ");
}
else
System.out.println("Matrix is not symmetric ");
}
}
```

OUTPUT



```
<terminated> matrix [Java Application] C:\Program Files (x86)\Java\jre1
Enter the number of row:
5
Enter the number of coloumn:
5
Matrix is symmetric
```

RESULT
The program has been executed and the output was verified.

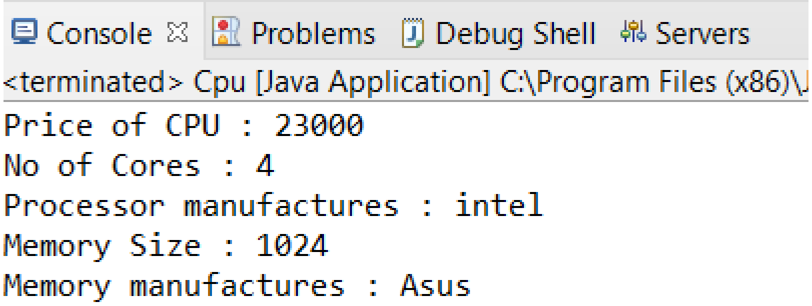
5. **Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.**

PROGRAM

Cpu.java

```
package org.springmvc;
public class Cpu {
 int price;
 Cpu(int p) {
 this.price = p;
 }
class Processor {
 int cores;
 String manufacture;
 Processor(int n, String m) {
 this.cores = n;
 this.manufacture = m;
 }
 void display() {
 System.out.println("No of Cores : " + this.cores);
 System.out.println("Processor manufactures : " + this.manufacture);
 }
 }
 static class Ram {
 int memory;
 String manufacture;
 Ram(int n, String m) {
 this.memory = n;
 this.manufacture = m;
 }
 void display() {
 System.out.println("Memory Size : " + this.memory);
 System.out.println("Memory manufactures : " + this.manufacture);
 }
 }
 void display() {
 System.out.println("Price of CPU : " + this.price);
 }
 public static void main(String[] args) {
 Cpu intel = new Cpu(23000);
 Cpu.Processor i_processor = intel.new Processor(4, "intel");
 Cpu.Ram i_ram = new Ram(1024, "Asus");
 intel.display();
 i_processor.display();
 i_ram.display();
 }
}
```

OUTPUT



```
<terminated> Cpu [Java Application] C:\Program Files (x86)\J
Price of CPU : 23000
No of Cores : 4
Processor manufactures : intel
Memory Size : 1024
Memory manufactures : Asus
```

RESULT

The program has been executed and the output was verified.

Course Outcome 2 (CO2) :

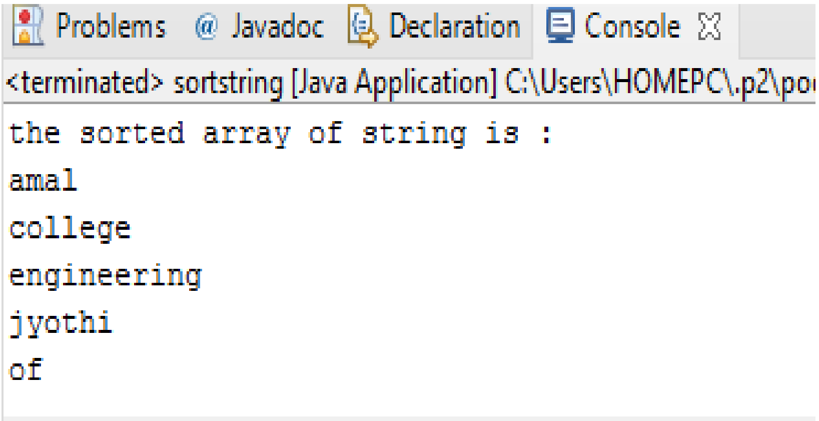
1. Program to Sort strings.

PROGRAM

Sortstring.java

```
package sample;
public class sortstring {
public static void main(String[] args){
String names[]={"amal","jyothi","college","of","engineering"}; String temp;
int n= names.length;
int i;
int j;
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
if(names[i].compareTo(names[j])>0)
{
temp=names[i]; names[i]=names[j]; names[j]=temp;
}
}
}
System.out.println("the sorted array of string is :");
for(i=0;i<n;i++)
{
System.out.println(names[i]);
}
}
}
```

OUTPUT



<terminated> sortstring [Java Application] C:\Users\HOMEPC\.p2\po

```
the sorted array of string is :
amal
college
engineering
jyothi
of
```

RESULT

The program has been executed and the output was verified.

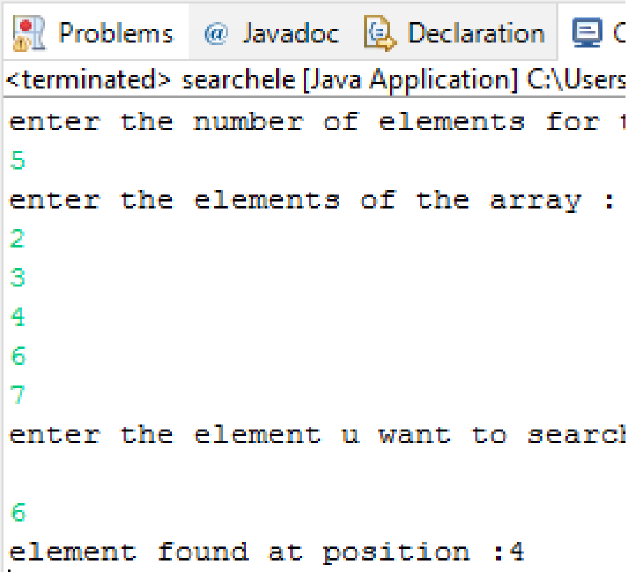
2. Search an element in an array.

PROGRAM

Searchele.java

```
package sample;

import java.util.*;

public class searchele{

public static void main(String[] args) {
int n,i,b,flag=0;

Scanner s=new Scanner(System.in);
System.out.println("enter the number of elements for the array :"); n=s.nextInt();
int a[]=new int[n];
System.out.println("enter the elements of the array :");
for(i=0;i<n;i++)
{
a[i]=s.nextInt();
}

System.out.println("enter the element u want to search :"); b=s.nextInt();
for(i=0;i<n;i++)
{
if(a[i]==b)
{
flag=1;
break;
}
else
{
flag=0;
}
}

if(flag==1)
{
System.out.println("element found at position :"+(i+1));
}
else
{
System.out.println("element not found");
}

}
}
```

OUTPUT



```
<terminated> searchele [Java Application] C:\Users
enter the number of elements for t
5
enter the elements of the array :
2
3
4
6
7
enter the element u want to searcl

6
element found at position :4
```

RESULT

The program has been executed and the output was verified.

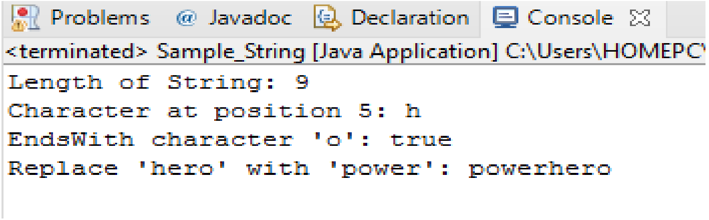
3. Perform string manipulations

PROGRAM

Sample_String.java

```
package sample;
public class Sample_String {
public static void main(String[] args) {
String str_Sample = "superhero";
System.out.println("Length of String: " + str_Sample.length());

System.out.println("Character at position 5: " + str_Sample.charAt(5));

System.out.println("EndsWith character 'o': " + str_Sample.endsWith("o"));

System.out.println("Replace 'hero' with 'power': " + str_Sample.replace("super", "power"));
}
}
```

OUTPUT



```
<terminated> Sample_String [Java Application] C:\Users\HOMEPC\
Length of String: 9
Character at position 5: h
EndsWith character 'o': true
Replace 'hero' with 'power': powerhero
```

RESULT

The program has been executed and the output was verified.

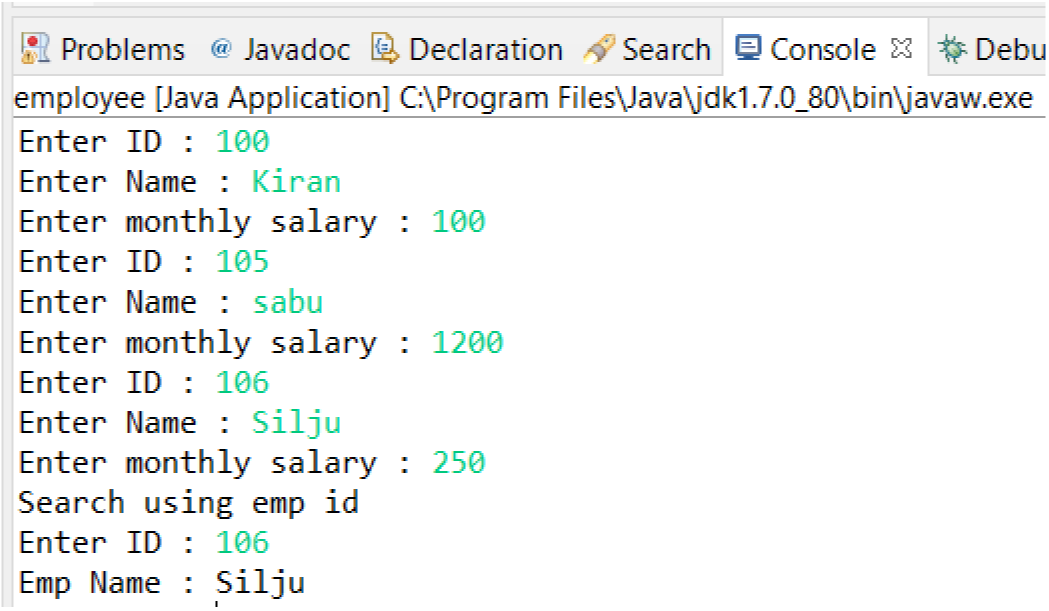
4. Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

PROGRAM

Employee.java

```
package Bankapplet;
import java.util.*;
public class employee {
int eNo;
String eName;
int eSalary;

public void read(){
 Scanner sc= new Scanner(System.in);
 System.out.print("Enter ID : ");
 eNo = Integer.parseInt(sc.nextLine());
 System.out.print("Enter Name : ");
 eName = sc.nextLine();
 System.out.print("Enter monthly salary : ");
 eSalary = Integer.parseInt(sc.nextLine());
}
public void display(){
 System.out.println("Emp Name : "+ eName );
}

public static void main(String[] args) {
int i,n=3;
int No;
employee emp[] = new employee[n];

for(i=0;i<n;i++){
 emp[i] = new employee();
 emp[i].read();
}
System.out.println("Search using emp id");

while(true){
 Scanner sc= new Scanner(System.in);
 System.out.print("Enter ID : ");
 No = Integer.parseInt(sc.nextLine());
 for(i=0;i<n;i++){
  if(emp[i].eNo == No){
   emp[i].display();
   break;
  }
}
}
}
}
```

OUTPUT



```
employee [Java Application] C:\Program Files\Java\jdk1.7.0_80\bin\javaw.exe
Enter ID : 100
Enter Name : Kiran
Enter monthly salary : 100
Enter ID : 105
Enter Name : sabu
Enter monthly salary : 1200
Enter ID : 106
Enter Name : Silju
Enter monthly salary : 250
Search using emp id
Enter ID : 106
Emp Name : Silju
```

RESULT

The program has been executed and the output was verified.

Course Outcome 3(CO3):

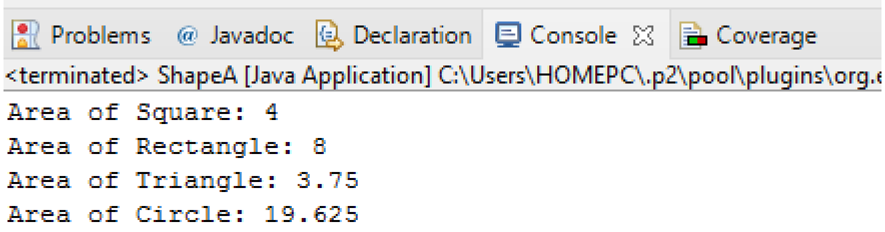
1.Area of different shapes using overloaded functions

PROGRAM

ShapeA.java

```
package sample;
public class ShapeA {
int area(int side)
{
return side*side;
}

int area(int l,int b)
{
return l*b;
}

double area(double b,double h)
{
return (0.5*(b*h));
}

double area(double r)
{
return (3.14*r*r);
}

public static void main(String[] args) {
ShapeA obj=new ShapeA();
System.out.println("Area of Square: "+obj.area(2));
System.out.println("Area of Rectangle:"+obj.area(2,4));
System.out.println("Area of Triangle: "+obj.area(2.5,3.0));
System.out.println("Area of Circle: "+obj.area(2.5));
}
}
```

OUTPUT



```
<terminated> ShapeA [Java Application] C:\Users\HOMEPC\.p2\pool\plugins\org.e
Area of Square: 4
Area of Rectangle: 8
Area of Triangle: 3.75
Area of Circle: 19.625
```

RESULT

The program has been executed and the output was verified.

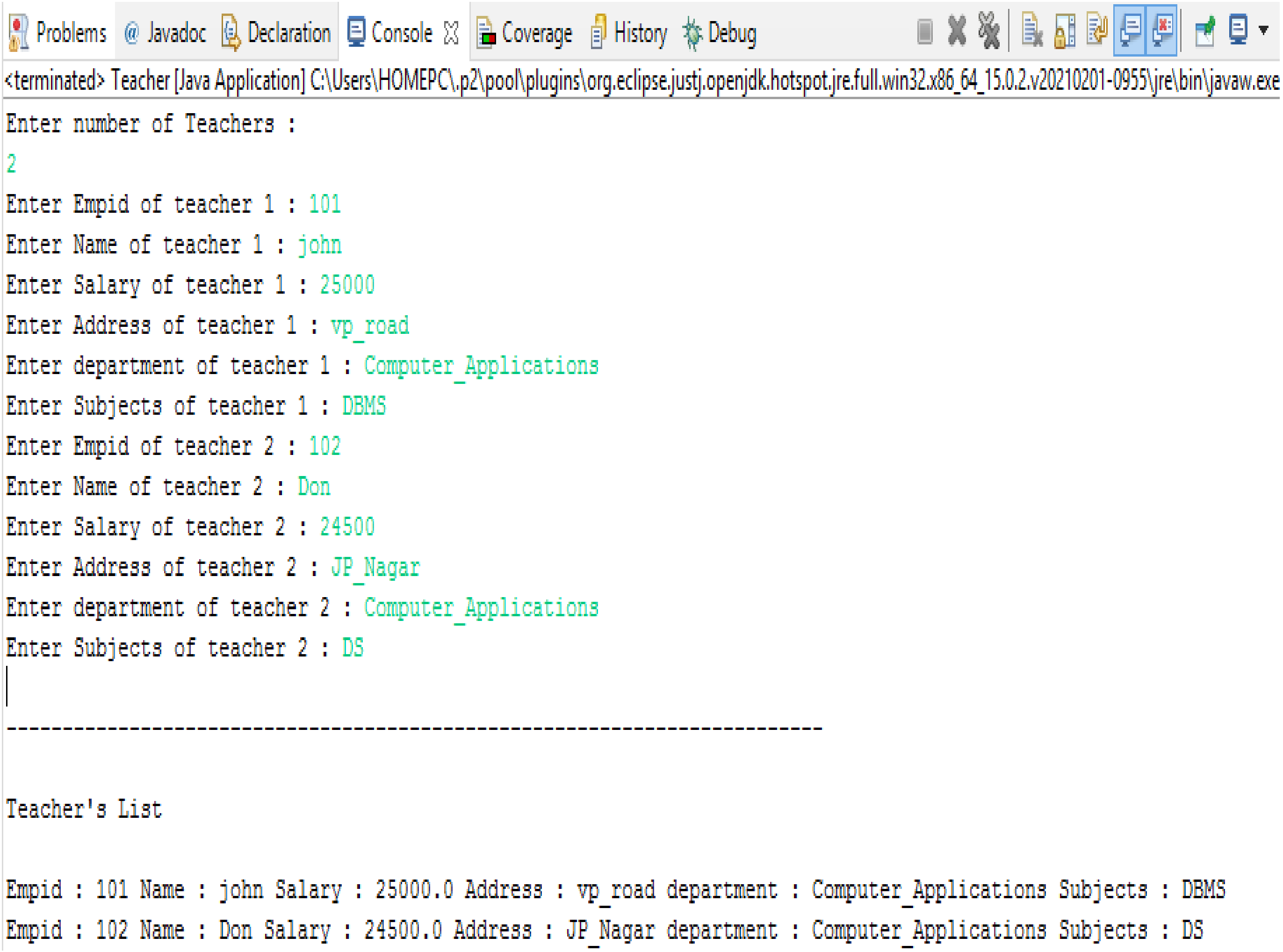
2. Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data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

PROGRAM

Employee.java

```
package sample;
import java.util.*;
class Employee {
int empid;
String name,address;
double salary;
public Employee(int empid, String name, String address, double salary) {
this.empid = empid;
this.name = name;
this.address = address;
this.salary = salary;
}
}
public class Teacher extends Employee
{
String subject,department;
public Teacher(int empid, String name, String address, doublesalary,String department,String
subject ) {
super(empid, name, address, salary);
this.subject = subject;
this.department = department;
}
void display()
{
 System.out.println("Empid : "+this.empid+" Name : "+this.name+" Salary : "+this.salary+"
Address : "+this.address+" department : "+this.department+" Subjects : "+this.subject);
}
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
int n;
System.out.println("Enter number of Teachers : ");
n=sc.nextInt();
Teacher obj[]=new Teacher[n];
for(int i=0;i<n;i++) {
 int j = i+1;
 System.out.print("Enter Empid of teacher "+j+" : ");
 int Empid = sc.nextInt();
 System.out.print("Enter Name of teacher "+j+" : ");
 String Name = sc.next();
 System.out.print("Enter Salary of teacher "+j+" : ");
 double Salary = sc.nextDouble();
 System.out.print("Enter Address of teacher "+j+" : ");
 String Address = sc.next();
 System.out.print("Enter department of teacher "+j+" : ");
 String department =sc.next();
 System.out.print("Enter Subjects of teacher "+j+" : ");
 String Subjects =sc.next();
 obj[i] = new Teacher(Empid, Name, Address, Salary, department, Subjects);
}
System.out.println("\n------------------------------------------------------------------------\n");
System.out.println("Teacher's List \n");
 for(int i=0;i<n;i++) {
 obj[i].display();
 }
}
}
```

OUTPUT



```
<terminated> Teacher [Java Application] C:\Users\HOMEPC\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe
Enter number of Teachers :
2
Enter Empid of teacher 1 : 101
Enter Name of teacher 1 : john
Enter Salary of teacher 1 : 25000
Enter Address of teacher 1 : vp_road
Enter department of teacher 1 : Computer_Applications
Enter Subjects of teacher 1 : DBMS
Enter Empid of teacher 2 : 102
Enter Name of teacher 2 : Don
Enter Salary of teacher 2 : 24500
Enter Address of teacher 2 : JP_Nagar
Enter department of teacher 2 : Computer_Applications
Enter Subjects of teacher 2 : DS

------------------------------------------------------------------------


Teacher's List


Empid : 101 Name : john Salary : 25000.0 Address : vp_road department : Computer_Applications Subjects : DBMS
Empid : 102 Name : Don Salary : 24500.0 Address : JP_Nagar department : Computer_Applications Subjects : DS
```

RESULT

The program has been executed and the output was verified.

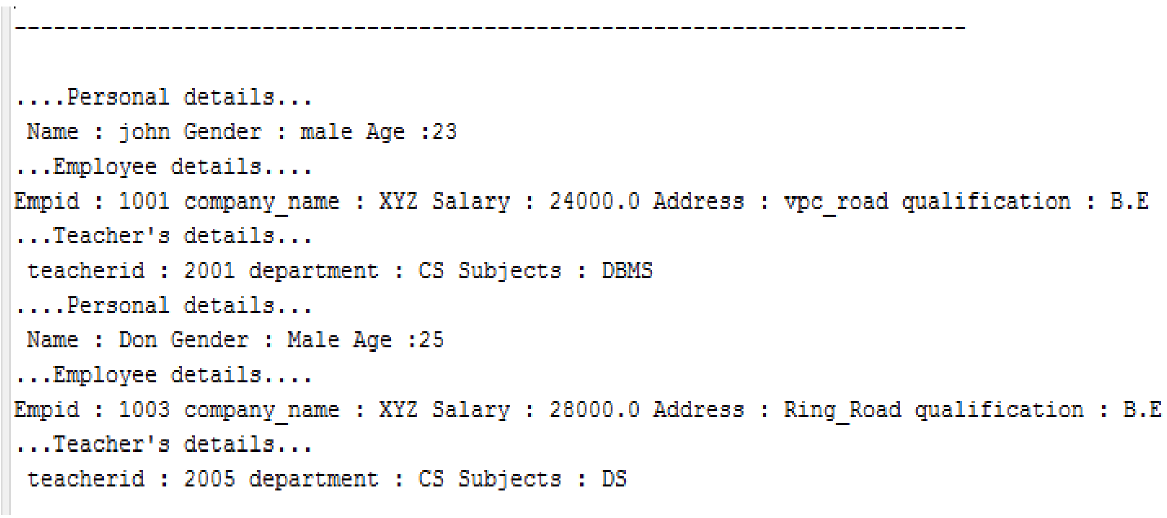
3. Create a class ‘Person’ with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class ‘Employee’ that inherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor. 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.

PROGRAM

Person.java

```
package sample;
import java.util.Scanner;
class Person
{
String name,gender,address;
int age;
public Person(String name, String gender, String address, int age) {
super();
this.name = name;
this.gender = gender;
this.address = address;
this.age = age;
}
}
class Employee extends Person {
int empid;
String company_name,qualification;
double salary;
public Employee(String name, String gender, String address, int age, int empid, String
company_name,String qualification, double salary) {
super(name, gender, address, age);
this.empid = empid;
this.company_name = company_name;
this.qualification = qualification;
this.salary = salary;
}
}
class Teacher extends Employee
{
String subject,department;
int teacherid;
public Teacher(String name, String gender, String address, int age, int empid, String
company_name,String qualification, double salary, String subject, String department, int
teacherid) {
super(name, gender, address, age, empid, company_name, qualification, salary);
this.subject = subject;
this.department = department;
this.teacherid = teacherid;
}
void display(){
System.out.println("....Personal details...");
System.out.println(" Name : "+this.name+" Gender : "+this.gender+" Age :"+this.age);
System.out.println("...Employee details. .. ");
System.out.println("Empid : "+this.empid +" company_name : "+this.company_name+" Salary
: "+this.salary+" Address : "+this.address+" qualification : "+this.qualification);
System.out.println("...Teacher's details.  ");
System.out.println(" teacherid : "+this.teacherid+ " department : "+this.department+" Subjects :
"+this.subject);
}
}
public class Main {
public static void main(String[] args) {
Scanner s=new Scanner(System.in);
int n;
System.out.println("Enter number of Teachers : ");
n=s.nextInt();
Teacher obj[]=new Teacher[n];
for(int i=0;i<n;i++) {
System.out.println("Enter the person name:");
String nam1=s.next();
System.out.println("Enter the Gender: ");
String gen1=s.next();
System.out.println("Enter the Address: ");
String adr1=s.next();
System.out.println("Enter the Age:");
int age1=s.nextInt();
System.out.println("Enter the Employee id: ");
int id1=s.nextInt();
System.out.println("Enter the Company name: ");
String cname1=s.next();
System.out.println("Enter the Salary:");
double sal1=s.nextDouble();
System.out.println("Enter the Qualification:");
String qu1=s.next();
System.out.println("Enter the Teacher id: ");
int tid1=s.nextInt();
System.out.println("Enter the Department:");
String dept1=s.next();
System.out.println("Enter the Subject:");
String sub1=s.next();
obj[i]=new
Teacher(nam1,gen1,adr1,age1,id1,cname1,qu1,sal1,sub1,dept1,tid1);
}
System.out.println("\n..........................................................................................................\n");
 for(int i=0;i<n;i++) {
 obj[i].display();
 }
}
}
```

OUTPUT



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

....Personal details...
 Name : john Gender : male Age :23
...Employee details....
Empid : 1001 company_name : XYZ Salary : 24000.0 Address : vpc_road qualification : B.E
...Teacher's details...
 teacherid : 2001 department : CS Subjects : DBMS
....Personal details...
 Name : Don Gender : Male Age :25
...Employee details....
Empid : 1003 company_name : XYZ Salary : 28000.0 Address : Ring_Road qualification : B.E
...Teacher's details...
 teacherid : 2005 department : CS Subjects : DS
```

RESULT

The program has been executed and the output was verified.

4. 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.

PROGRAM

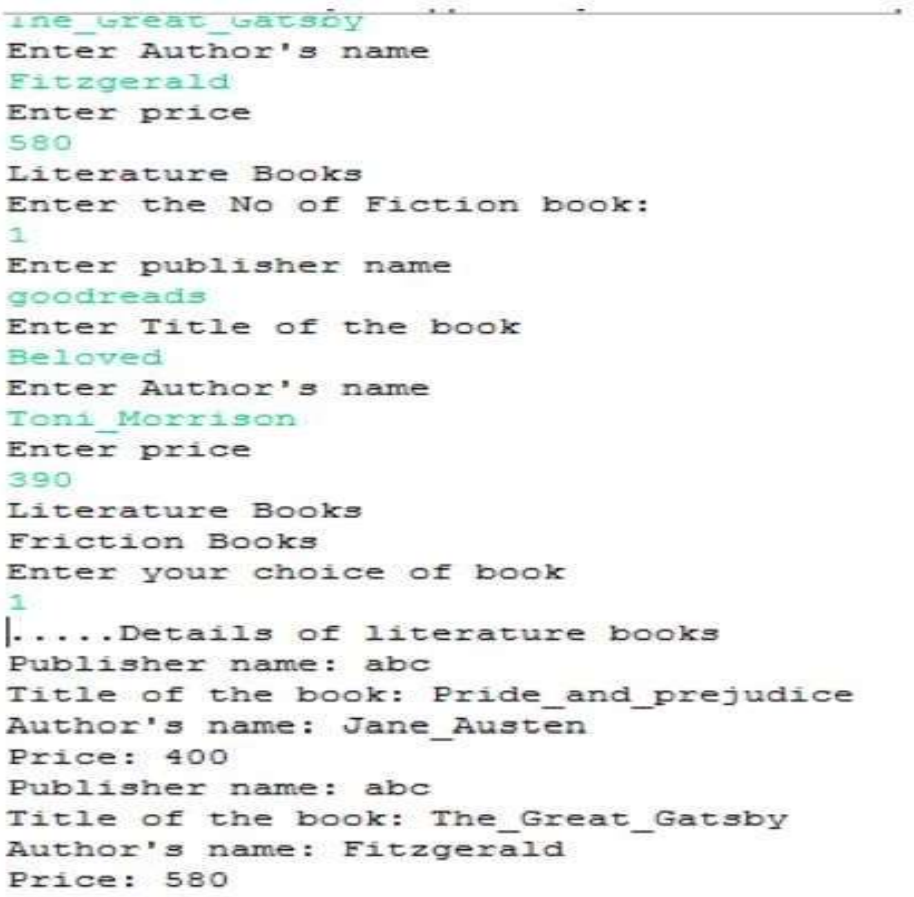
Publisher.java

```
package sample;
import java.util.Scanner;
 class Publisher {
 String Pubname;
 Publisher()
 {
 Scanner s=new Scanner(System.in);
 System.out.println("Enter publisher name");
 Pubname=s.next();
 }
 }
 class Book extends Publisher
 {
 String title, author;
 int price;

 Book()
 {
 Scanner s=new Scanner(System.in);
 System.out.println("Enter Title of the book");
 title=s.next();
 System.out.println("Enter Author's name");
 author=s.next();
 System.out.println("Enter price");
```

```
price=s.nextInt();
 }
 }
class Literature extends Book
{
Literature()
{
 System.out.println("Literature Books");
}
 void display()
 {
 System.out.println("Publisher name: "+Pubname);
 System.out.println("Title of the book: "+title);
 System.out.println("Author's name: "+author);
 System.out.println("Price: "+price);
 }
}
class Fiction extends Literature
{
Fiction()
{
 System.out.println("Friction Books");
}
void display()
{
 super.display();
}
public static void main(String args[])
{
int n;
Scanner s=new Scanner(System.in);
System.out.println("Enter the No of literature book: ");
int a=s.nextInt();
Literature L[]=new Literature[a];
for(int i=0;i<a;i++)
{
L[i]=new Literature();
}

System.out.println("Enter the No of Fiction book: ");
int b=s.nextInt();
Fiction F[]=new Fiction[b];
for(int i=0;i<b;i++)
{
F[i]=new Fiction();
}
int no;
System.out.println("Enter your choice of book");
no=s.nextInt();
int type =no;
```

```
switch (no)
{
case 1:
System.out.println(".... Details of literature books");
for(int i=0;i<a;i++)

L[i].display();
break;
case 2:
System.out.println(".... Details of fiction books");
for(int i=0;i<b;i++)
F[i].display();
break;
default:
System.out.println("Wrong input");
}
}
}
```

OUTPUT



```
The_Great_Gatsby
Enter Author's name
Fitzgerald
Enter price
580
Literature Books
Enter the No of Fiction book:
1
Enter publisher name
goodreads
Enter Title of the book
Beloved
Enter Author's name
Toni_Morrison
Enter price
390
Literature Books
Friction Books
Enter your choice of book
1
.....Details of literature books
Publisher name: abc
Title of the book: Pride_and_prejudice
Author's name: Jane_Austen
Price: 400
Publisher name: abc
Title of the book: The_Great_Gatsby
Author's name: Fitzgerald
Price: 580
```

RESULT

The program has been executed and the output was verified.

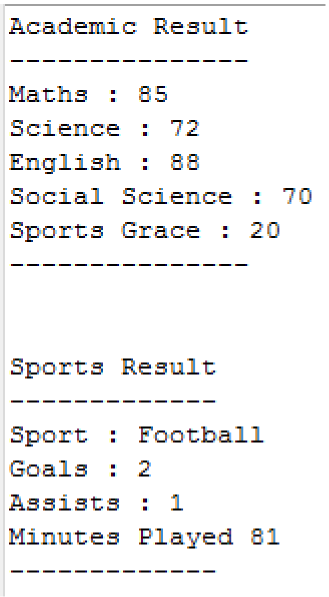
5.Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

PROGRAM

Student.java

```
package javaprj;
class Student
{
int maths = 85; int science = 72;
int english = 88;
int socialScience = 70;
}
class Sports extends Student
{
String sport = "Football"; int goals = 2;
int assists = 1;
int minutesPlayed = 81; int grace = 20;
}
public class Result extends Sports{
 public void display()
{
System.out.println("Academic Result");
System.out.println("    ");
System.out.println("Maths : "+this.maths);
System.out.println("Science : "+this.science);
System.out.println("English : "+this.english);
System.out.println("Social Science : "+this.socialScience);
System.out.println("Sports Grace : "+this.grace);
System.out.println("    ");
System.out.println("\n");
System.out.println("Sports Result");
System.out.println("    ");
System.out.println("Sport : "+this.sport);
System.out.println("Goals : "+this.goals);
System.out.println("Assists : "+this.assists);
System.out.println("Minutes Played "+this.minutesPlayed);
System.out.println("    ");
}
public static void main(String[] args) {
Result obj=new Result();
obj.display();
}
}
```

OUTPUT



```
Academic Result
---------------
Maths : 85
Science : 72
English : 88
Social Science : 70
Sports Grace : 20
---------------


Sports Result
-------------
Sport : Football
Goals : 2
Assists : 1
Minutes Played 81
-------------
```

RESULT

The program has been executed and the output was verified.

6. 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.

PROGRAM

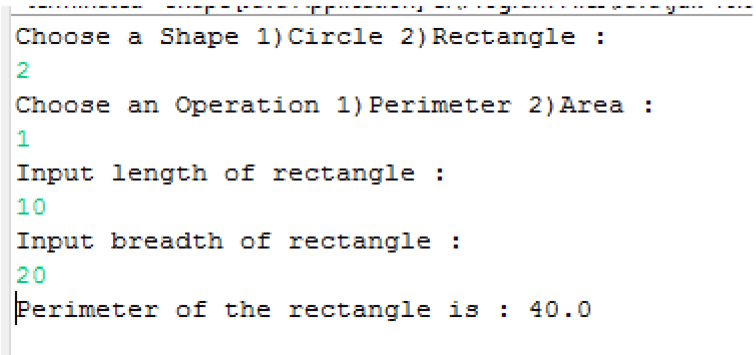
Circle.java

```
package javaprj;
import java.util.*;
interface prototype
{
public void area();
public void perimeter();
}
class Circle implements prototype
{
int radius;
Scanner scanner = new Scanner(System.in);

public void area()
{
System.out.println("Input radius of circle : ");
radius = scanner.nextInt();
String area = Double.toString(Math.PI*radius*radius);
System.out.println("Area of the circle is : "+area);
}
public void perimeter()
{
```

```
System.out.println("Input radius of circle : ");
radius = scanner.nextInt();
String perimeter = Double.toString(Math.PI*radius*2);
System.out.println("Circumference of the circle is : "+perimeter);
}
}
class Rectangle implements prototype
{

int length; int breadth;
Scanner scanner = new Scanner(System.in);
public void area()
{
System.out.println("Input length of rectangle : ");
length = scanner.nextInt();
System.out.println("Input breadth of rectangle : ");
length = scanner.nextInt();
String area = Double.toString(length*breadth);
System.out.println("Area of the rectangle is : "+area);
}
public void perimeter()
{
System.out.println("Input length of rectangle : ");
length = scanner.nextInt();
System.out.println("Input breadth of rectangle : ");
length = scanner.nextInt();
String perimeter = Double.toString(2*(length+breadth));
System.out.println("Perimeter of the rectangle is : "+perimeter);
}
}
public class Shape {
public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);
int shape,operation;

System.out.println("Choose a Shape 1)Circle 2)Rectangle : ");
shape = scanner.nextInt();
System.out.println("Choose an Operation 1)Perimeter 2)Area : ");
operation = scanner.nextInt();
if(shape==1){
Circle circle = new Circle();
if(operation==1){
circle.perimeter();
}
else if(operation==2)
{
circle.area();
}
```

```
else {
System.out.println("Operation code.");
}
}
else if(shape==2)
{
Rectangle rectangle = new Rectangle();
if(operation==1){
rectangle.perimeter();

}
else if(operation==2)
{
rectangle.area();
}
else {
System.out.println("Operation code :");
System.exit(0);
}
}
else {
System.out.println("Incorrect Shape code.");
}
}
}
```

OUTPUT

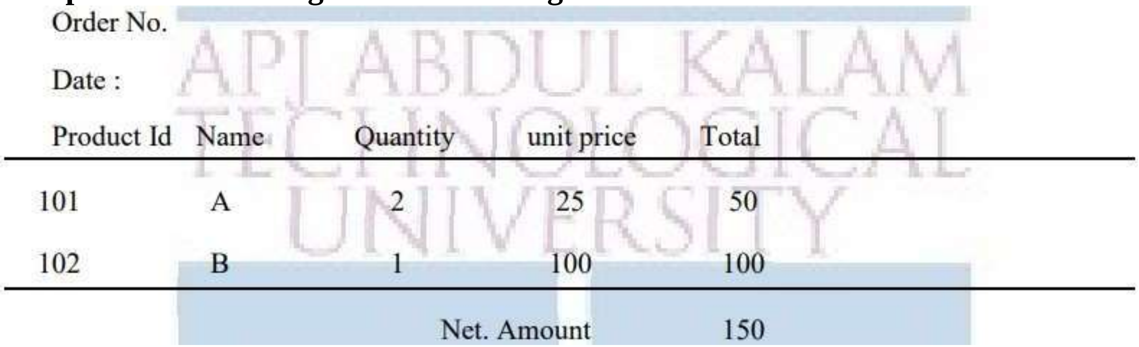


```
Choose a Shape 1)Circle 2)Rectangle :
2
Choose an Operation 1)Perimeter 2)Area :
1
Input length of rectangle :
10
Input breadth of rectangle :
20
Perimeter of the rectangle is : 40.0
```

RESULT

The program has been executed and the output was verified.

7. 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

PROGRAM

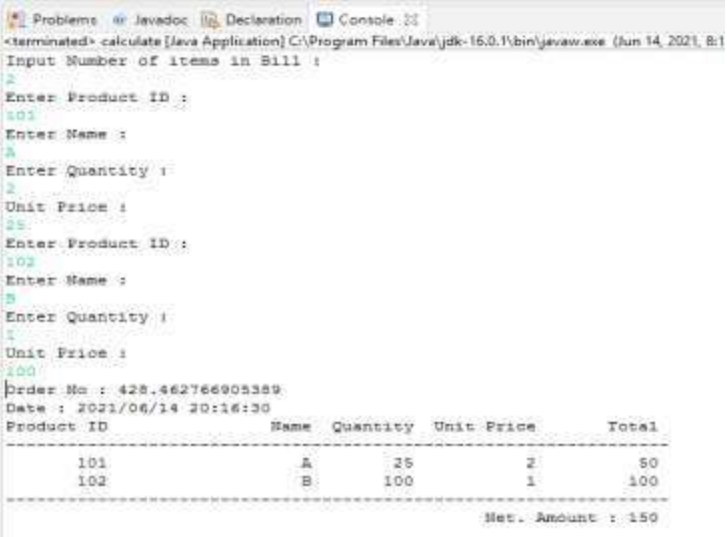
ProductBill.java

```
package javaprj;
import java.time.format.DateTimeFormatter;
import java.time.LocalDateTime;
import java.util.Scanner;
interface Bill {
String productId = "";

String productName="";
int unitPrice = 0;
int quantity = 0; int total = 0;
public void printBillItem();
public void printBillHeader();
public void printBillFooter(int billTotal);

}
class ProductBill implements Bill {

String productId = "";
String productName="";
int unitPrice = 0;
int quantity = 0;
int total = 0;

ProductBill(String productId,String productName,int unitPrice,int quantity){
this.productId = productId;
this.productName = productName;
this.unitPrice = unitPrice;
this.quantity = quantity;
this.total = unitPrice*quantity;
}

public void printBillHeader() {
System.out.println("Order No : " + Math.random() * 1000);
DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
LocalDateTime now = LocalDateTime.now();
System.out.println("Date : " + dtf.format(now));
System.out.println("Product ID        Name  Quantity Unit Price              Total ");
System.out.println("              ");
}
public void printBillItem()
{
System.out.format("%10s%20s%10d%12d%12d\n",this.productId,this.productName,this.unitPrice,this.quantity,this.total);
}
public void printBillFooter(int billTotal)
{
System.out.println("    ");
System.out.format("%64s \n","Net. Amount : "+billTotal);
}
}
```

Calculate.java

```
public class calculate {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.println("Input Number of items in Bill : ");
int count = scanner.nextInt();
ProductBill[] productBill=new ProductBill[count];
int billTotal=0;
for(int i=0;i<count;i++) {
System.out.println("Enter Product ID : ");
String productId = scanner.next();
System.out.println("Enter Name : ");
String name = scanner.next();
System.out.println("Enter Quantity : ");
int qty = scanner.nextInt();
System.out.println("Unit Price : ");
int up = scanner.nextInt();
productBill[i]=new ProductBill(productId,name,up,qty);
}
if(count>0){
productBill[0].printBillHeader();
for(int i=0;i<count;i++) {
productBill[i].printBillItem();
billTotal += productBill[i].total;
}
productBill[0].printBillFooter(billTotal);
}
}
}
```

OUTPUT



```
<terminated> calculate [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (Jun 14, 2021, 8:1
Input Number of items in Bill :
2
Enter Product ID :
101
Enter Name :
A
Enter Quantity :
2
Unit Price :
25
Enter Product ID :
102
Enter Name :
B
Enter Quantity :
1
Unit Price :
100
Order No : 428.462766905389
Date : 2021/06/14 20:16:30
Product ID                    Name  Quantity  Unit Price       Total
---------------------------------------------------------------------
       101                       A        25           2          50
       102                       B       100           1         100
---------------------------------------------------------------------
                                                Net. Amount : 150
```

RESULT

The program has been executed and the output was verified.

Course Outcome 4 ( ):

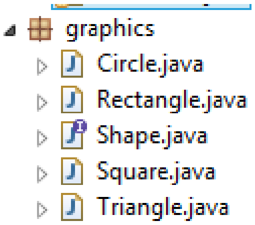
1.Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

<u>PROGRAM</u>

Mainclass.java

```
import graphics.*;
import java.util.Scanner; public class Mainclass {

public static void main(String[] args){
Scanner scanner = new Scanner(System.in); int shape;
System.out.println("Choose a Shape 1)Circle 2)Rectangle 3)Square 4)Triangle : ");
shape = scanner.nextInt();
if(shape==1){
Circle circle = new Circle(); circle.area();
}
else if(shape==2)
{
Rectangle rectangle = new Rectangle(); rectangle.area();
}
else if(shape==3)
{
Square square = new Square(); square.area();
}
else if(shape==4)
{
Triangle triangle = newTriangle(); triangle.area();
}
else {
System.out.println("Incorrect Shape code.");
}
}
}
```

Graphic package



Cricle.java

```
package graphics;
import java.util.Scanner;
public class Circle implements Shape { int radius;
Scanner scanner = new Scanner(System.in);
public void area() {
System.out.println("Input radius of circle : "); radius = scanner.nextInt();
String area = Double.toString(Math.PI*radius*radius); System.out.println("Area of the circle is : "+area);
}
}
```

Rectangle.java

```
package graphics;
import java.util.Scanner;
public class Rectangle implements Shape {
int length; int breadth;
Scanner scanner = new Scanner(System.in); public void area() {
System.out.println("Input length of rectangle : ");
length = scanner.nextInt();
System.out.println("Input breadth of rectangle : ");
length = scanner.nextInt();
String area = Double.toString(length*breadth);
System.out.println("Area of the rectangle is : "+area);
}
}
```

Shape .java

```
package graphics;
public interface Shape {
public void area();
}
```

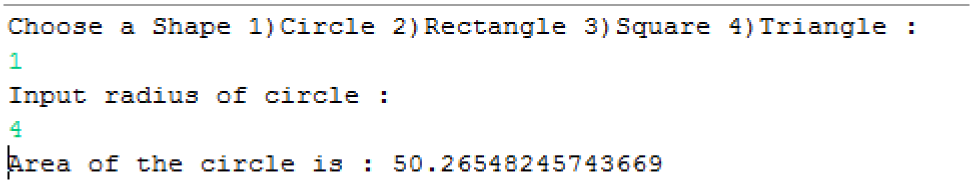
Square.java

```
package graphics;
import java.util.Scanner;
public class Square {
int side;
Scanner scanner = new Scanner(System.in);

public void area() {
System.out.println("Input side length of square : ");
side = scanner.nextInt();
String area = Double.toString(side*side);
System.out.println("Area of the square : "+area);
}
}
```

Triangle.java

```
package graphics;
import java.util.Scanner;
public class Triangle {
int height;
int breadth;
Scanner scanner = new Scanner(System.in）；
public void area() {
System.out.println("Input height of the triangle : ");
height = scanner.nextInt();
System.out.println("Input breadth of triangle : ");
breadth = scanner.nextInt();
String area = Double.toString((height*breadth)/2f);
 System.out.println("Area of the triangle is : "+area);
}
}
```

OUTPUT



```
Choose a Shape 1)Circle 2)Rectangle 3)Square 4)Triangle :
1
Input radius of circle :
4
Area of the circle is : 50.26548245743669
```

RESULT

The program has been executed and the output was verified.

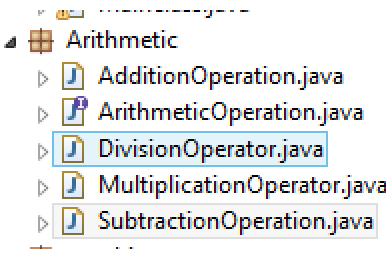
2. Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers.

PROGRAM

Main.java

```
import Arithmetic.*;
import java.util.Scanner;

public class Main {
public static void main(String[] args)
{
Scanner scanner = new Scanner(System.in);
System.out.println("Choose an operation : 1.Add, 2.Subtract, 3.Multiply, 4.Divide"); int choice
= scanner.nextInt();

System.out.println("Enter the numbers : "); int number1 = scanner.nextInt();
int number2= scanner.nextInt();
System.out.println("The Result is : "); switch (choice){
```

```
case 1:
AdditionOperation additionOperation = new AdditionOperation();
System.out.println(additionOperation.operateNumbers(number1,number2)); break;
case 2:
SubtractionOperation subtractionOperation = new SubtractionOperation();
System.out.println(subtractionOperation.operateNumbers(number1,number2)); break;
case 3:
MultiplicationOperator multiplicationOperator = new MultiplicationOperator();
System.out.println(multiplicationOperator.operateNumbers(number1,number2)); break;
case 4:
DivisionOperator divisionOperator = new DivisionOperator();
System.out.println(divisionOperator.operateNumbers(number1,number2)); break;
default:
System.out.println("Invalid Code");
}
}
}
//AdditionOperation.java package Arithmetic;
public class AdditionOperation implements ArithmeticOperation { public int
operateNumbers(int number1,int number2)
{
return number1+number2;
}
}
```



```
Arithmetic
    AdditionOperation.java
    ArithmeticOperation.java
    DivisionOperator.java
    MultiplicationOperator.java
    SubtractionOperation.java
```

ArithmeticOperation.java

```
package Arithmetic;
public interface ArithmeticOperation {
public int operateNumbers(int number1,int number2);
}
```

DivisionOperator.java

```
package Arithmetic;
public class DivisionOperator implements ArithmeticOperation{
public int operateNumbers(int number1,int number2)
{
return number1*number2;
}
}
```

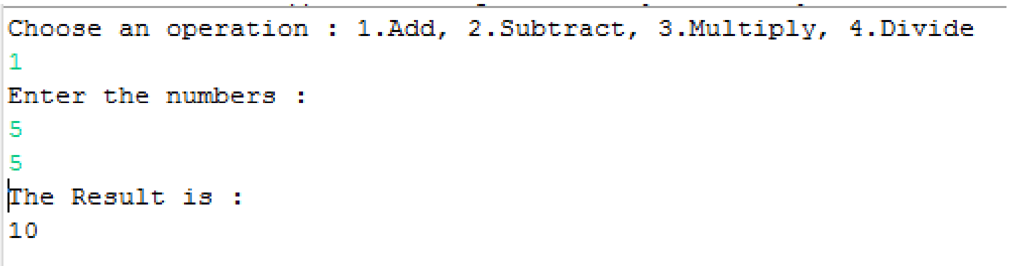
MultiplicationOperator.java

```
package Arithmetic;
public class MultiplicationOperator implements ArithmeticOperation{ public int
operateNumbers(int number1,int number2)
{
return number1*number2;
}
}
```

SubtractionOperation.java

```
package Arithmetic;
public class SubtractionOperation implements ArithmeticOperation{

public int operateNumbers(int number1,int number2)
{
return number1-number2;
}
}
```

OUTPUT



```
Choose an operation : 1.Add, 2.Subtract, 3.Multiply, 4.Divide
1
Enter the numbers :
5
5
The Result is :
10
```

RESULT

The program has been executed and the output was verified.

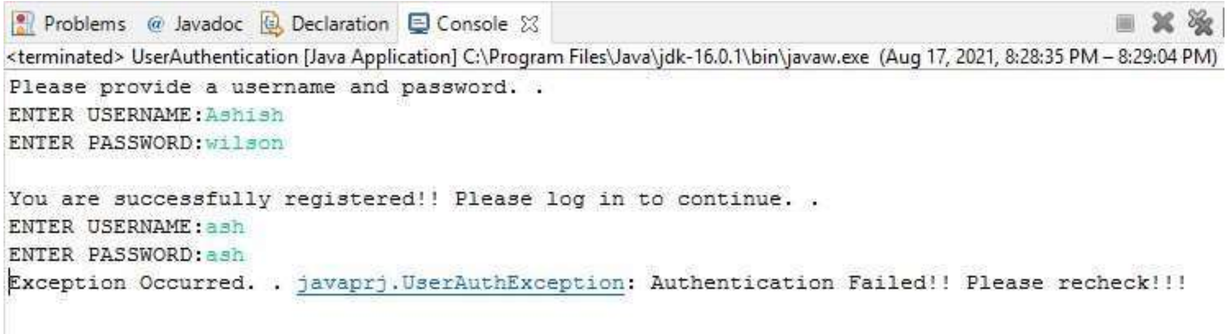
3. Write a user defined exception class to authenticate the user name and password.

PROGRAM

UserAuthException.java

```
package javaprj;
import java.util.Scanner;
class UserAuthException extends Exception {
private static final long serialVersionUID = 1L;
UserAuthException(String s){
super(s);
}
}
public class UserAuthentication {
public static void main(String[] args) {
String username, u_name, password,p_wrd;
 Scanner sc = new Scanner(System.in);
 System.out.print("Please provide a username and password. . ");
 System.out.print("\nENTER USERNAME:");
 username = sc.nextLine();
 System.out.print("ENTER PASSWORD:");

 password = sc.nextLine();
 System.out.print("\nYou are successfully registered!! Please log in to continue. .");
 System.out.print("\nENTER USERNAME:");
 u_name = sc.nextLine();
 System.out.print("ENTER PASSWORD:");
 p_wrd = sc.nextLine();
 sc.close();
 try
 {
 authentication(username,password,u_name,p_wrd);
 }catch(Exception e)
 {
 System.out.println("Exception Occurred. . "+e);
 }
}
public static void authentication(String uname, String pwd, String u, String p) throws
UserAuthException{
if((u=="") || (p=="")) {
throw new UserAuthException("Fields cannot be empty!!!");
}
else if(u.equals(uname) && p.equals(pwd)){
System.out.println("Authentication Successful!!!");
}
else if((u!=uname) || (p!=pwd)) {
throw new UserAuthException("Authentication Failed!! Please
recheck!!!");
}
else {
System.exit(0);
}
 }
}
```

OUTPUT



```
<terminated> UserAuthentication [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (Aug 17, 2021, 8:28:35 PM – 8:29:04 PM)
Please provide a username and password. .
ENTER USERNAME:Ashish
ENTER PASSWORD:wilson

You are successfully registered!! Please log in to continue. .
ENTER USERNAME:ash
ENTER PASSWORD:ash
Exception Occurred. . javaprj.UserAuthException: Authentication Failed!! Please recheck!!!
```

RESULT

The program has been executed and the output was verified.

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

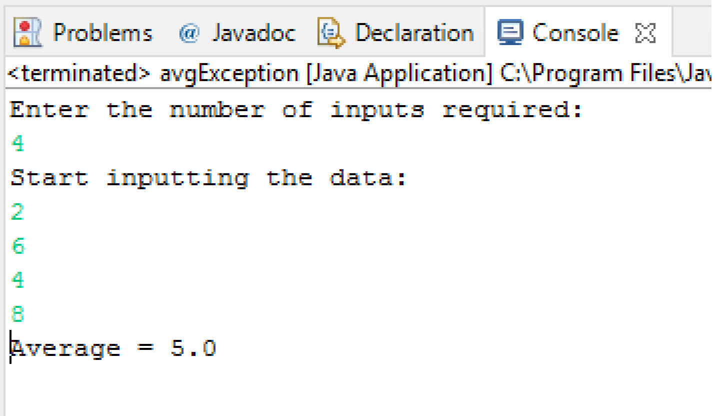
PROGRAM

NegInputException.java

```
package javaprj;
import java.util.Scanner;
class NegInputException extends Exception {
private static final long serialVersionUID = 1L;
NegInputException(String s){
super(s);
}

}
public class avgException {

public static void main(String[] args) {
int N;

float Sum = 0,avg;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the number of inputs required:");
N = sc.nextInt();
float[] numbers = new float[N];
System.out.println("Start inputting the data:");

for( int i=0; i < N ; i++)
{
numbers[i]=sc.nextInt();
try{ if(numbers[i]<0)
{
throw new NegInputException("Negative inputs not allowed!");
}
else
{
Sum += numbers[i];
}
}catch(NegInputException e)
{
System.out.println("Exception Occurred. . "+e); System.exit(0);
}
}
sc.close();
avg = Sum / N;
System.out.println("Average = "+ avg);
}
}
```

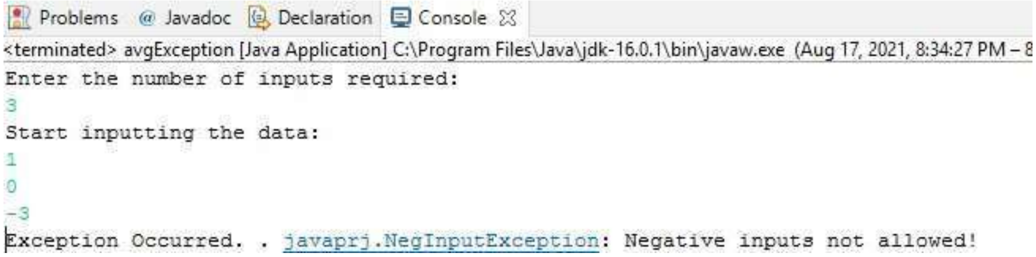
OUTPUT

1.



```
<terminated> avgException [Java Application] C:\Program Files\Ja
Enter the number of inputs required:
4
Start inputting the data:
2
6
4
8
Average = 5.0
```

2.



```
<terminated> avgException [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (Aug 17, 2021, 8:34:27 PM – 8
Enter the number of inputs required:
3
Start inputting the data:
1
0
-3
Exception Occurred. . javaprj.NegInputException: Negative inputs not allowed!
```

RESULT

The program has been executed and the output was verified.

6.Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class) .

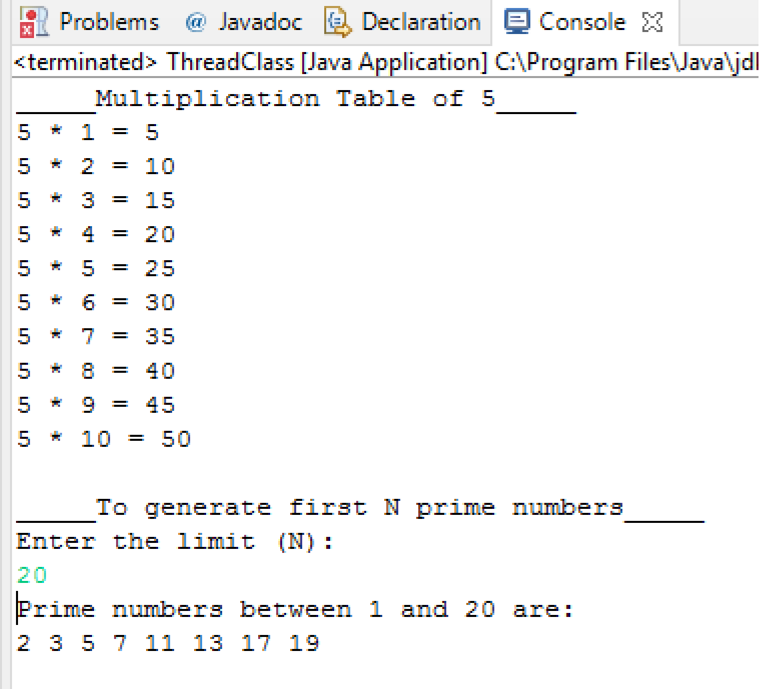
PROGRAM

MulTable.java

```
package javaprj;

import java.util.Scanner;

class MulTable extends Thread{
public void run() {
int num = 5;
System.out.printf("     _Multiplication Table of 5      \n");
for(int i = 1; i <= 10; ++i)
{
```

```
System.out.printf("%d * %d = %d \n", num, i, num * i);
}
}
}

class PrimeNo extends Thread{
public void run() {
int i, j,flag;
Scanner s = new Scanner(System.in);
System.out.println("\n            To generate first N prime numbers     ");
System.out.println("Enter the limit (N):");
int N = s.nextInt();
System.out.println("Prime numbers between 1 and " + N + " are:");

for (i = 1; i <= N; i++)
{
if (i == 1 || i == 0)
continue;
flag = 1;
for (j = 2; j <= i / 2; ++j)
{
if (i % j == 0)
{
flag = 0;
break;

}
}

if (flag == 1)
System.out.print(i + " ");
}
}
}

public class ThreadClass {
public static void main(String[] args) throws InterruptedException {
MulTable m = new MulTable();
m.start();
m.sleep(200);

PrimeNo p = new PrimeNo(); p.start();
p.sleep(200);
}

}
```

OUTPUT



```
<terminated> ThreadClass [Java Application] C:\Program Files\Java\jdl
_____Multiplication Table of 5_____
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50

_____To generate first N prime numbers_____
Enter the limit (N):
20
Prime numbers between 1 and 20 are:
2 3 5 7 11 13 17 19
```

RESULT

The program has been executed and the output was verified.

6.Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)

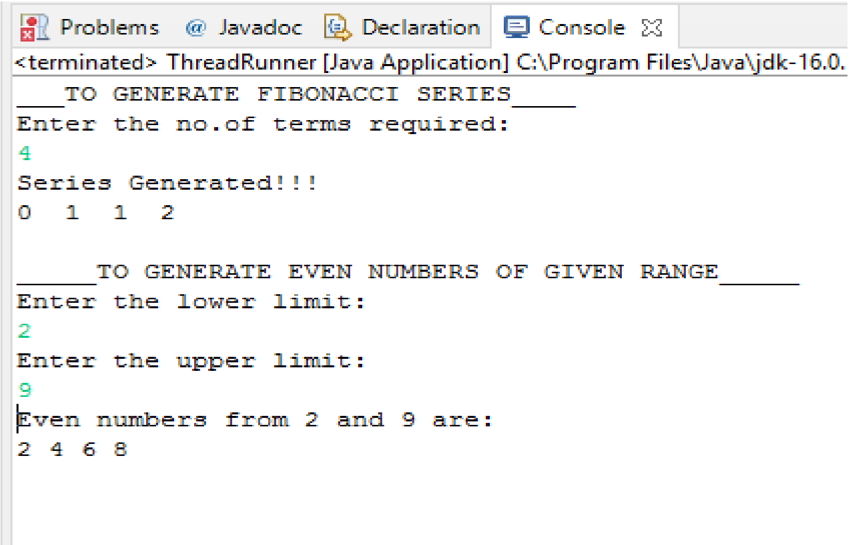
PROGRAM

Fibonacci.java

```
package javaprj;
import java.util.Scanner;
class Fibonacci implements Runnable{
public void run(){
int first = 0, second = 1, next;
Scanner sc= new Scanner(System.in);
System.out.println("___TO GENERATE FIBONACCI SERIES____");
System.out.println("Enter the no.of terms required:");
int n=sc.nextInt();
System.out.println("Series Generated!!!");
for (int i = 1; i <= n; ++i){
System.out.print(first + " ");
 next = first + second;
first = second;
second = next;
  }
}
}
```

```
class EvenNo implements Runnable{
public void run(){
Scanner sc= new Scanner(System.in);
int lower, upper;
System.out.println("\n\n_____TO GENERATE EVEN NUMBERS OF GIVEN
RANGE_____");
System.out.println("Enter the lower limit:");
lower=sc.nextInt();
System.out.println("Enter the upper limit:");
upper=sc.nextInt();
System.out.println("Even numbers from " + lower + " and " + upper + " are:");
for (int i = lower; i <= upper; i++){
 if (i%2!=0)
 continue;
 else
 {
 System.out.print(i+" ") ;
 }
 }
 }
 }
public class ThreadRunner {
public static void main(String arg[]) throws InterruptedException
{
Fibonacci obj1 = new Fibonacci();
Thread a=new Thread(obj1);
a.start();
a.sleep(2000);
EvenNo obj2 = new EvenNo();
Thread b= new Thread(obj2);
b.start();
b.sleep(1000);
}
}
```

OUTPUT



```
<terminated> ThreadRunner [Java Application] C:\Program Files\Java\jdk-16.0.
___TO GENERATE FIBONACCI SERIES____
Enter the no.of terms required:
4
Series Generated!!!
0  1  1  2

_____TO GENERATE EVEN NUMBERS OF GIVEN RANGE_____
Enter the lower limit:
2
Enter the upper limit:
9
Even numbers from 2 and 9 are:
2 4 6 8
```

RESULT

The program has been executed and the output was verified.

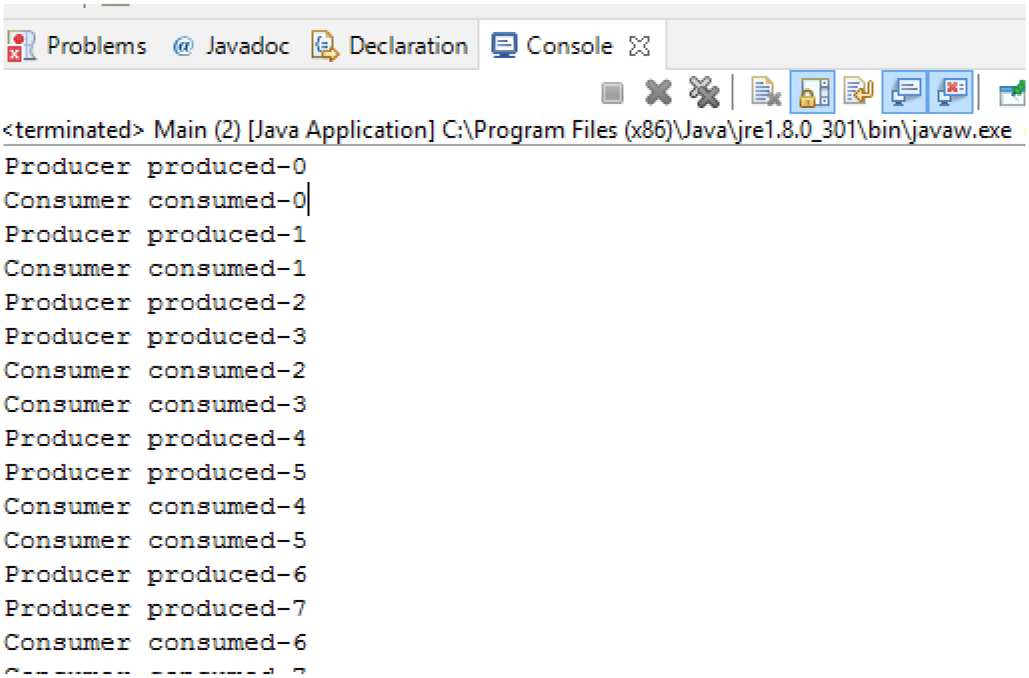
7. Producer/Consumer using ITC

PROGRAM

Main.java

```
import java.util.LinkedList;
public class Main {
    public static void main(String[] args) throws InterruptedException {
        final PC pc = new PC();

        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    pc.produce();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });

        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    pc.consume();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });
        t1.start();
        t2.start();
        t1.join();
        t2.join();
    }
    public static class PC {
        LinkedList<Integer> list = new LinkedList<>();
        int capacity = 2;
        public void produce() throws InterruptedException {
            int value = 0;
            while (true) {
                synchronized (this) {
                   while (list.size() == capacity)
                        wait();
                    System.out.println("Producer produced-"
                            + value);
                    list.add(value++);


                    notify();
                    Thread.sleep(1000);
                }
            }
        }


        public void consume() throws InterruptedException {
            while (true) {
                synchronized (this) {
                    while (list.size() == 0)
                        wait();
                    int val = list.removeFirst();
                    System.out.println("Consumer consumed-"+ val);
                    notify();
                    Thread.sleep(1000);
                }
            }
        }
PC(){
    }
PC(int count){
     this.count=count;
}
    public void run() {
        for(int i=1;primeCount<count;i++){
            boolean isPrime=true;
        if(i==1)
          {
            i++;
          }

          for (int j=2;j<i;j++){
            if(i%j==0){
              isPrime = false;
              break;
            }
          }

          if (isPrime){
            System.out.println("Prime Number : "+i);
            primeCount++;
          }
        }

      }
    }
    }
```

OUTPUT



```
<terminated> Main (2) [Java Application] C:\Program Files (x86)\Java\jre1.8.0_301\bin\javaw.exe
Producer produced-0
Consumer consumed-0
Producer produced-1
Consumer consumed-1
Producer produced-2
Producer produced-3
Consumer consumed-2
Consumer consumed-3
Producer produced-4
Producer produced-5
Consumer consumed-4
Consumer consumed-5
Producer produced-6
Producer produced-7
Consumer consumed-6
Consumer consumed-7
```

RESULT

The program has been executed and the output was verified.

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

A stack class is provided by the Java collection framework and it implements the Stack data structure. The stack implements LIFO i.e. Last In First Out. This means that the elements

pushed
last are the ones that are popped first.
1. push() Method adds element x to the stack.
2. pop() Method removes the last element of the stack.
3. top() Method returns the last element of the stack.
4. empty() Method returns whether the stack is empty or not.

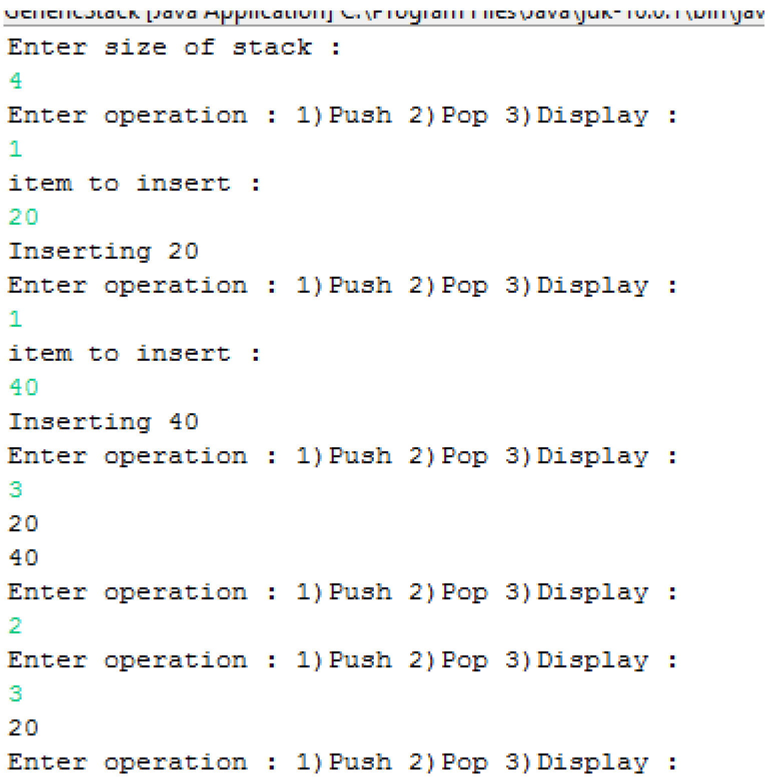
PROGRAM

GenericStack.java

```
package javaprj;
import java.util.Scanner;
public class GenericStack {
private int[] arr;
private int top;
private int capacity;
// Creating a stack
GenericStack(int size) {
arr = new int[size];
capacity = size;
top = -1;

}
// Add elements into stack
public void push(int x) {
if (isFull()) {
System.out.println("OverFlow");
}
else{
System.out.println("Inserting " + x);
arr[++top] = x;
}
}
// Remove element from stack
public int pop() {
if (isEmpty()) {
System.out.println("STACK EMPTY");
return -1;
}
else {
return arr[top--];
}
}
// Utility function to return the size of the stack
public int size() {
```

```
return top + 1;
}
// Check if the stack is empty
public Boolean isEmpty() {

return top == -1;
}
// Check if the stack is full
public Boolean isFull() {
return top == capacity - 1;
}
public void printStack() {
for (int i = 0; i <= top; i++) {
System.out.println(arr[i]);
}
}
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.println("Enter size of stack : ");
int count = scanner.nextInt();
GenericStack stack = new GenericStack(count);
while (true) {
System.out.println("Enter operation : 1)Push 2)Pop 3)Display : ");
int choice = scanner.nextInt();
switch (choice) {
case 1:
System.out.println("item to insert :");
int item = scanner.nextInt();


stack.push(item);
break;
case 2:
stack.pop();
break;
case 3:
stack.printStack();
}
}
}
}
```

OUTPUT



```
GenericStack [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\jav
Enter size of stack :
4
Enter operation : 1)Push 2)Pop 3)Display :
1
item to insert :
20
Inserting 20
Enter operation : 1)Push 2)Pop 3)Display :
1
item to insert :
40
Inserting 40
Enter operation : 1)Push 2)Pop 3)Display :
3
20
40
Enter operation : 1)Push 2)Pop 3)Display :
2
Enter operation : 1)Push 2)Pop 3)Display :
3
20
Enter operation : 1)Push 2)Pop 3)Display :
```

RESULT

The program has been executed and the output was verified.

9. Using generic method perform Bubble sort.

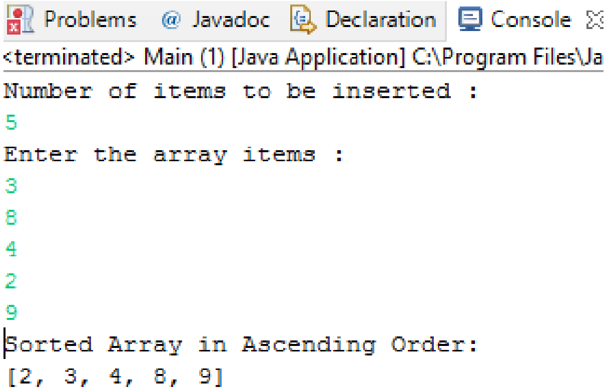
Bubble sort is a simple sorting algorithm. This sorting algorithm is a comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large datasets as its average and worst case complexity is of O(n2) where n is the number of items.

PROGRAM

Main.java

```
package javaprj;
import java.util.Arrays;
import java.util.Scanner;
public class Main {
static void bubbleSort(int array[]) {
int size = array.length;
for (int i = 0; i < size - 1; i++)
for (int j = 0; j < size - i - 1; j++)
if (array[j] > array[j + 1]) {
int temp = array[j];
array[j] = array[j + 1];
array[j + 1] = temp;

}
}
public static void main(String args[]) {
Scanner scanner = new Scanner(System.in);
System.out.println("Number of items to be inserted : ");
int count = scanner.nextInt();
int[] data = new int[count];
System.out.println("Enter the array items : ");
for(int i=0;i<count;i++)
{
data[i] = scanner.nextInt();
}
Main.bubbleSort(data);
System.out.println("Sorted Array in Ascending Order:");
System.out.println(Arrays.toString(data));
}
}
```

OUTPUT



```
<terminated> Main (1) [Java Application] C:\Program Files\Ja
Number of items to be inserted :
5
Enter the array items :
3
8
4
2
9
Sorted Array in Ascending Order:
[2, 3, 4, 8, 9]
```

RESULT

The program has been executed and the output was verified.

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

The ArrayList class extends AbstractList and implements the List interface. ArrayList supports dynamic arrays that can grow as needed.

Standard Java arrays are of a fixed length. After arrays are created, they cannot grow or shrink, which means that you must know in advance how many elements an array will hold.

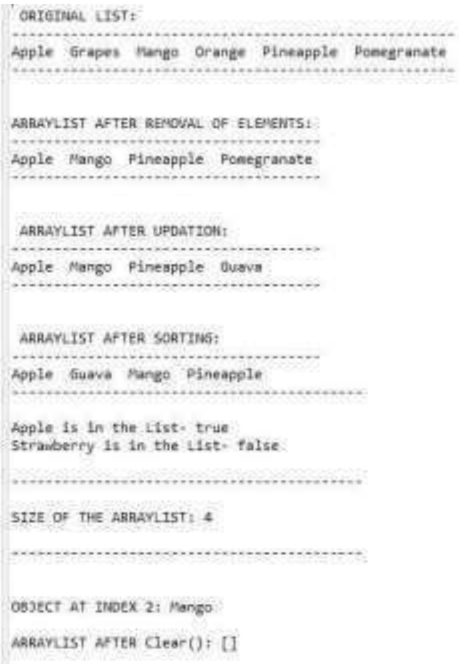
Array lists are created with an initial size. When this size is exceeded, the collection is automatically enlarged. When objects are removed, the array may be shrunk.

PROGRAM

arrayList.java

```
package javaprj;
import java.util.*;
public class arrayList {
public static void main(String[] args) {
// Creating ArrayList of type "String" which means we can only add "String"elements
ArrayList<String> fruits = new ArrayList<String>();
//adding elements to an ArrayList
fruits.add("Apple");
fruits.add("Grapes");
fruits.add("Mango");
fruits.add("Pineapple");
fruits.add("Pomegranate");
fruits.add(3, "Orange");
// Displaying elements
System.out.println("\n ORIGINAL LIST:");
System.out.println(" ");
for(String str : fruits)
System.out.printf(str+" ");
//Remove elements from ArrayList
fruits.remove("Grapes");
fruits.remove(2);
// Displaying elements
System.out.println("\n ");
System.out.println("\n\nARRAYLIST AFTER REMOVAL OF ELEMENTS:");
System.out.println(" ");
for(String str : fruits )
System.out.printf(str+" ");
//Updating the ArrayList
fruits.set(3,"Guava");
System.out.println("\n ");
System.out.println("\n\n ARRAYLIST AFTER UPDATION:");
System.out.println(" ");
for(String str : fruits )
System.out.printf(str+" ");
System.out.println("\n ");

//Sorting the ArrayList
Collections.sort(fruits);
System.out.println("\n\n ARRAYLIST AFTER SORTING:");
System.out.println(" ");
for (String str : fruits)
System.out.printf(str+" ");
// Checks whether the object is in the ArrayList
System.out.println("\n ");
System.out.println("\nApple is in the List- "+ fruits.contains("Apple"));
System.out.println("Strawberry is in the Lis"+fruits.contains("Strawberry"));
//Size of the ArrayList
System.out.println("\n ");
System.out.println("\nSIZE OF THE ARRAYLIST: "+ fruits.size());
//returns the object of list which is present at the specified index

System.out.println("\n ");
System.out.println("\n\nOBJECT AT INDEX 2: "+ fruits.get(2));
// removing all the elements of the ArrayList
fruits.clear();
System.out.println("\nARRAYLIST AFTER Clear(): "+ fruits);
}
```

<u>OUTPUT</u>



```
 ORIGINAL LIST:
----------------------------------------------------
Apple  Grapes  Mango  Orange  Pineapple  Pomegranate
----------------------------------------------------


ARRAYLIST AFTER REMOVAL OF ELEMENTS:
-----------------------------------
Apple  Mango  Pineapple  Pomegranate
-----------------------------------


 ARRAYLIST AFTER UPDATION:
-----------------------------------
Apple  Mango  Pineapple  Guava
-----------------------------------


 ARRAYLIST AFTER SORTING:
-----------------------------------
Apple  Guava  Mango  Pineapple
-----------------------------------------

Apple is in the List- true
Strawberry is in the List- false

-----------------------------------------

SIZE OF THE ARRAYLIST: 4

-----------------------------------------


OBJECT AT INDEX 2: Mango

ARRAYLIST AFTER Clear(): []
```

<u>RESULT</u>

The program has been executed and the output was verified.

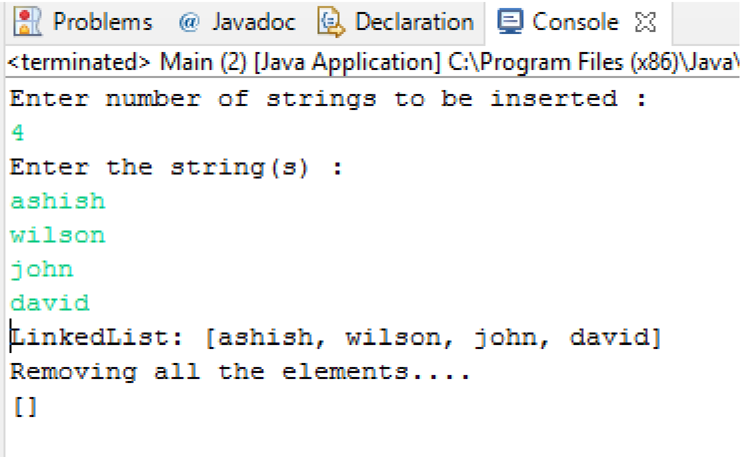
11. Program to remove all the elements from a linked list

<u>PROGRAM</u>

Main.java

```
import java.util.*;
public class Main {
public static void main(String[] args) {
int n;
String data;
LinkedList<String> linkedList = new LinkedList<String>();
System.out.println("Enter number of strings to be inserted : ");
Scanner scanner = new Scanner(System.in);
n = scanner.nextInt();
System.out.println("Enter the string(s) : ");
scanner.nextLine();

for (int i = 0; i < n; i++) {
data = scanner.nextLine();
linkedList.add(data);
}
System.out.println("LinkedList: " + linkedList);
System.out.println("Removing all the elements. .. ");
linkedList.clear();
System.out.println(linkedList);
}
}
```

<u>OUTPUT</u>



```
<terminated> Main (2) [Java Application] C:\Program Files (x86)\Java\
Enter number of strings to be inserted :
4
Enter the string(s) :
ashish
wilson
john
david
LinkedList: [ashish, wilson, john, david]
Removing all the elements....
[]
```

<u>RESULT</u>

The program has been executed and the output was verified.

12. Program to demonstrate the creation of queue object using the PriorityQueue class.

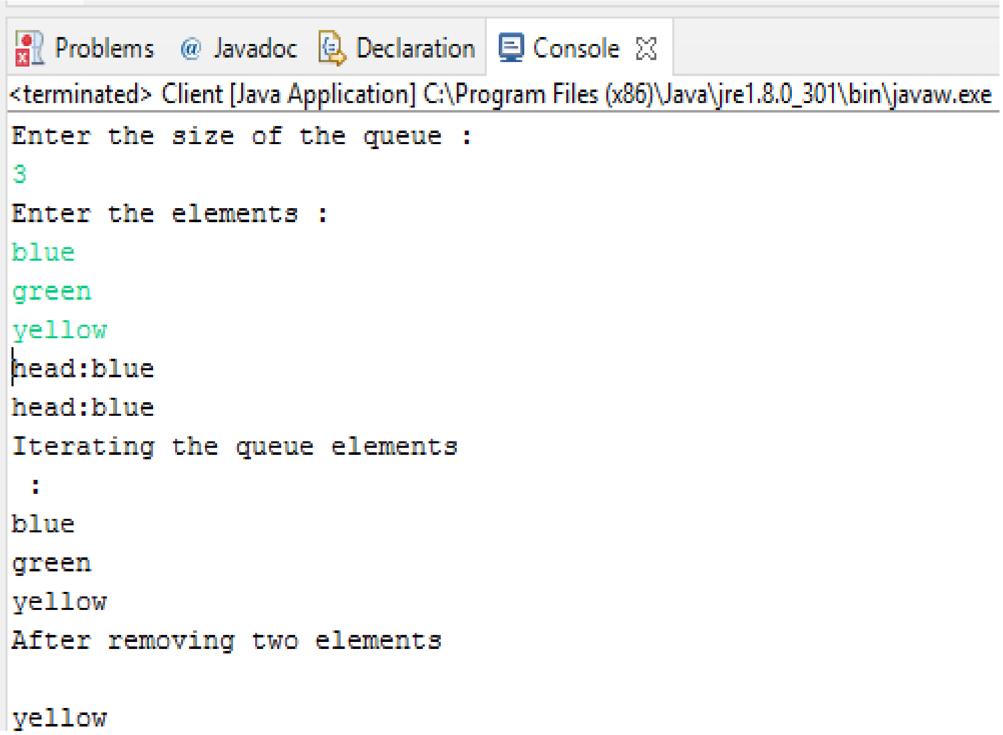
<u>PROGRAM</u>

Main.java

```
import java.util.PriorityQueue;
import java.util.Scanner;

public class Main {

    public static void main(String[] args)
    {
        PriorityQueue<String> queue=new PriorityQueue<String>();
```

```
      Scanner scanner=new Scanner(System.in);
      System.out.println("Enter the size of the queue : ");
      int n=scanner.nextInt();
      System.out.println("Enter the elements : ");

      for(int i =0;i<n;i++)
      {
         String st=scanner.next();
         queue.add(st);
      }
      System.out.println("head:"+queue.element());
      System.out.println("head:"+queue.peek());
      System.out.println("Iterating the queue elements\n : ");
      for (String value : queue) {
         System.out.println(value);
      }
      queue.remove();
      queue.poll();
      System.out.println("After removing two elements \n");
      for (String s : queue) {
         System.out.println(s);
      }
   }

}
```

OUTPUT



```
<terminated> Client [Java Application] C:\Program Files (x86)\Java\jre1.8.0_301\bin\javaw.exe
Enter the size of the queue :
3
Enter the elements :
blue
green
yellow
head:blue
head:blue
Iterating the queue elements
 :
blue
green
yellow
After removing two elements

yellow
```

RESULT

The program has been executed and the output was verified.

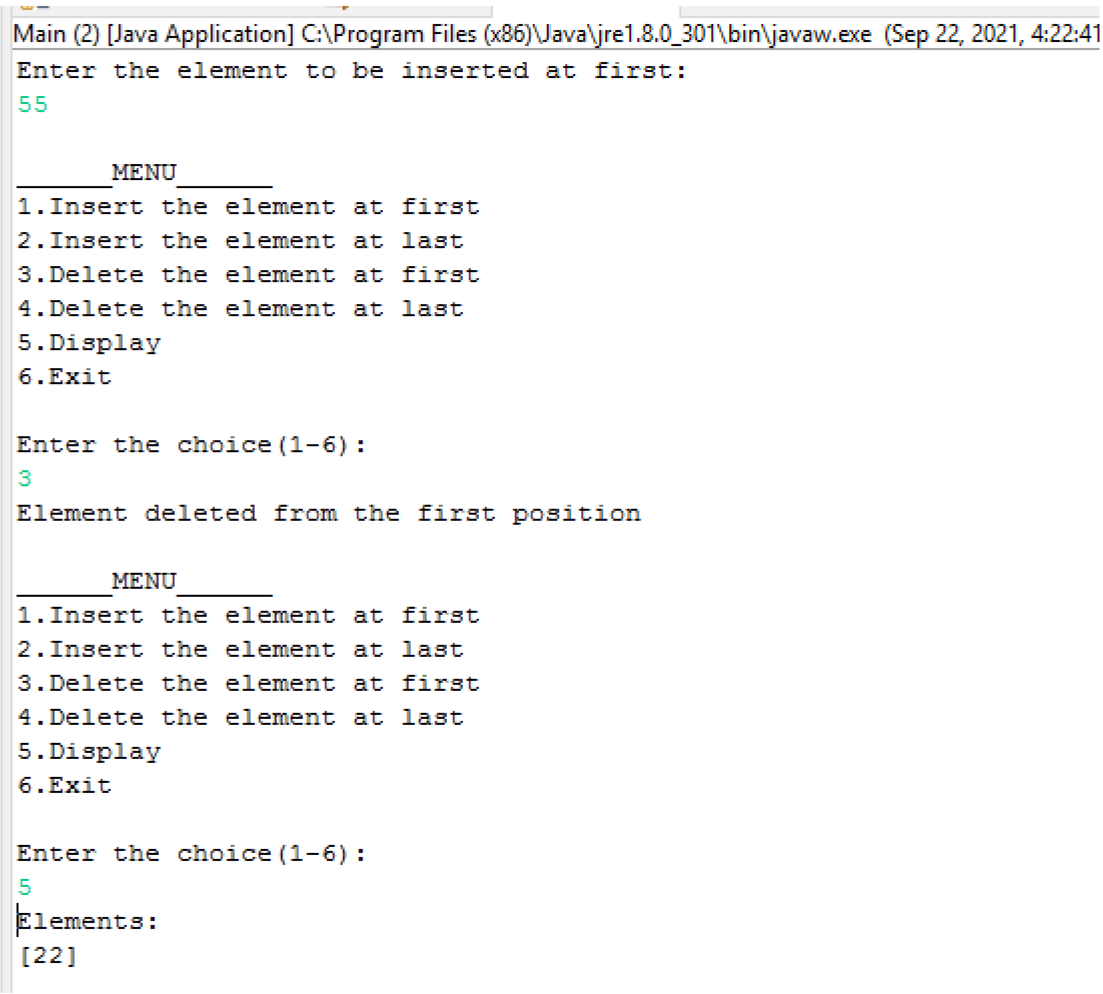
13. Program to demonstrate the addition and deletion of elements in deque

PROGRAM

Main.java

```
public class Main {
 public static void main(String[] args) {
 int ch, data;
 Deque<Integer> dq = new LinkedList<Integer>();
 Scanner sc = new Scanner(System.in);
 do
 {

 System.out.println("\n______MENU______");
 System.out.println("1.Insert the element at first");
 System.out.println("2.Insert the element at last");
 System.out.println("3.Delete the element at first");
 System.out.println("4.Delete the element at last");
 System.out.println("5.Display");
 System.out.println("6.Exit");
 System.out.println("\nEnter the choice(1-6):");
 ch = sc.nextInt();
 sc.nextLine();
 switch(ch)
 {
 case 1: System.out.println("Enter the element to be inserted at first:");
 data = sc.nextInt();
 dq.addFirst(data);
 break;
 case 2: System.out.println("Enter the element to be inserted at last:");
 data = sc.nextInt();
 dq.addLast(data);
 break;
 case 3: System.out.println("Element deleted from the first position");
 dq.removeFirst();
 break;
 case 4: System.out.println("Element deleted from the last position");
 dq.removeLast();
 break;
 case 5: System.out.println("Elements:");
 System.out.println(dq);
 break;
 case 6: System.exit(0);
 break;
 default:System.out.println("Invalid choice...");
}
 }while(true);
 }
}
```

OUTPUT



```
Main (2) [Java Application] C:\Program Files (x86)\Java\jre1.8.0_301\bin\javaw.exe (Sep 22, 2021, 4:22:41
Enter the element to be inserted at first:
55

______MENU______
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter the choice(1-6):
3
Element deleted from the first position

______MENU______
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter the choice(1-6):
5
Elements:
[22]
```

RESULT

The program has been executed and the output was verified.

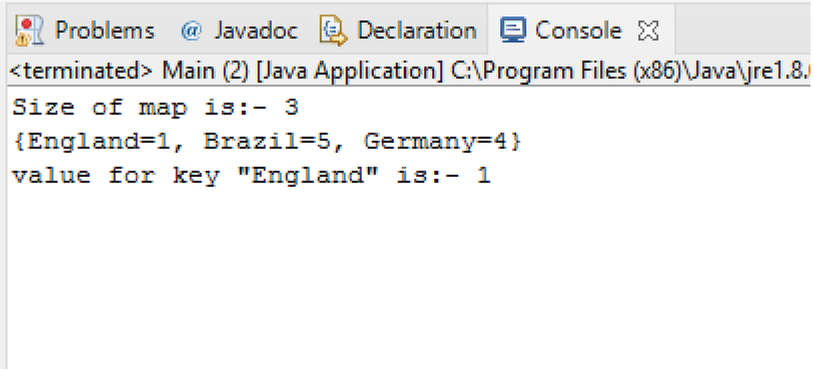
14. Program to demonstrate the working of Map interface by adding, changing and removing elements.

PROGRAM

Main.java

```
import java.util.HashMap;
// Main class
public class Main {
 // Main driver method
 public static void main(String[] args)
 {
 // Create an empty hash map by declaring object
 // of string and integer type

HashMap<String, Integer> map = new HashMap<>();
 // Adding elements to the Map
 // using standard add() method
 map.put("Germany", 4);

 map.put("England", 1);
 map.put("Brazil", 5);
 // Print size and content of the Map
 System.out.println("Size of map is:- "+ map.size());
 // Printing elements in object of Map
 System.out.println(map);
 // Checking if a key is present and if
 // present, print value by passing
 // random element
 if (map.containsKey("England")) {
 // Mapping
 Integer a = map.get("England");
 // Printing value fr the corresponding key
 System.out.println("value for key"+ " \"England\" is:- " + a);
 }
 }
}
```

OUTPUT



```
<terminated> Main (2) [Java Application] C:\Program Files (x86)\Java\jre1.8.
Size of map is:- 3
{England=1, Brazil=5, Germany=4}
value for key "England" is:- 1
```

RESULT

The program has been executed and the output was verified.

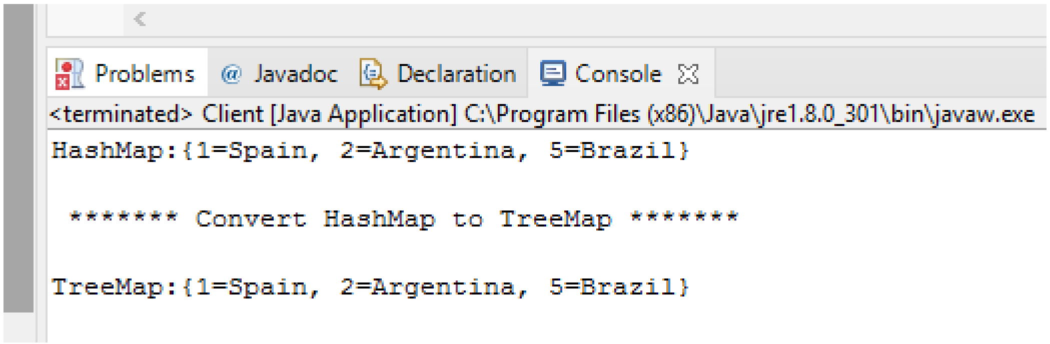
15. Program to Convert HashMap to TreeMap.

PROGRAM

Main.java

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Map<Integer,String> hm=new LinkedHashMap<>();
        hm.put(1,"England");
        hm.put(1,"Spain");
        hm.put(2,"France");
        hm.put(5,"Brazil");
        hm.put(2,"Argentina");
        System.out.println("HashMap:"+hm);
        Map<Integer,String> tm=new TreeMap<>(hm);
        System.out.println("\n ******* Convert HashMap to TreeMap *******\n");
        System.out.println("TreeMap:"+tm);
    }
}
```

OUTPUT



```
<terminated> Client [Java Application] C:\Program Files (x86)\Java\jre1.8.0_301\bin\javaw.exe
HashMap:{1=Spain, 2=Argentina, 5=Brazil}

 ******* Convert HashMap to TreeMap *******

TreeMap:{1=Spain, 2=Argentina, 5=Brazil}
```

RESULT

The program has been executed and the output was verified.

Course Outcome 5 (CO5):

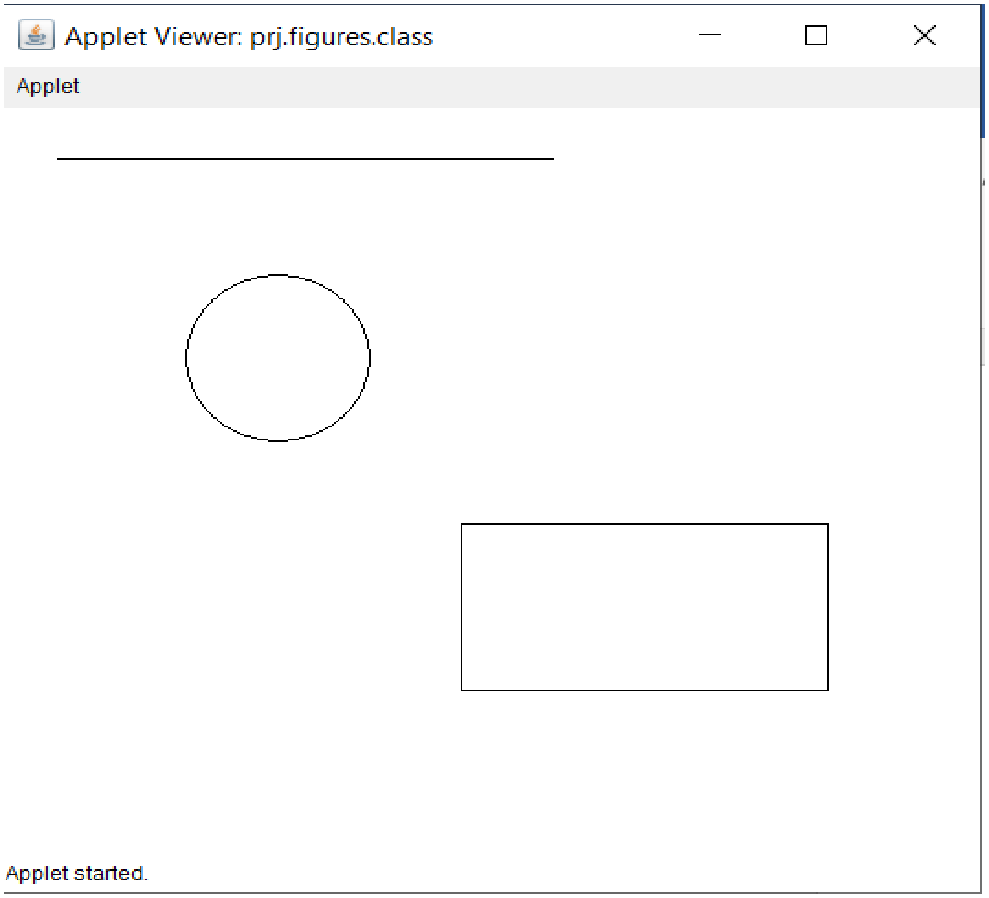
1. Program to draw Circle, Rectangle, Line in Applet.

PROGRAM

Figures.java

```
package prj;
import java.applet.*;
import java.awt.Graphics;
public class figures extends Applet {
 public void paint(Graphics g)
{
g.drawLine(30,30,300,30);
g.drawOval(100,100,100,100);
g.drawRect(250, 250, 200, 100);
}
}
```

OUTPUT



RESULT

The program has been executed and the output was verified.

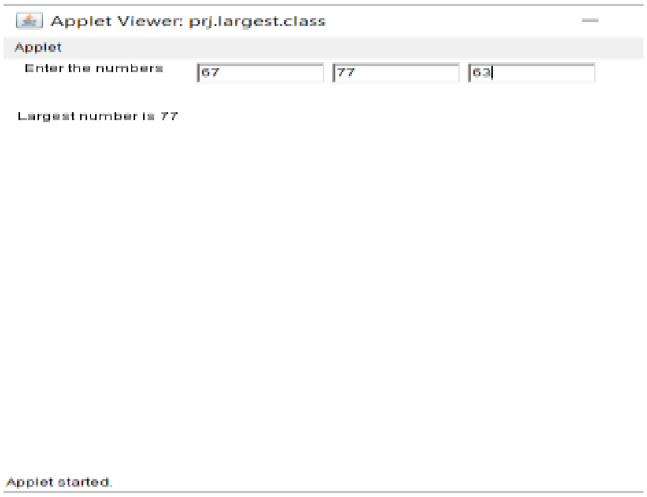
2. Program to find maximum of three numbers using AWT.

PROGRAM

Largest.java

```
package prj;
import java.awt.*;
import java.awt.Event;
import java.applet.*;
public class largest extends Applet
{
 TextField Txt1,Txt2,Txt3;
 public void init(){
 Txt1 = new TextField(10);
 Txt2 = new TextField(10);

 Txt3 = new TextField(10);
 add(Txt1);
 add(Txt2);
 add(Txt3);
 }
 public void paint(Graphics g){
 int a, b, c,result;
 String str;
 g.drawString("Enter the numbers ",15,15);
 str=Txt1.getText();
 a=Integer.parseInt(str);
 str=Txt2.getText();
 b=Integer.parseInt(str);
 str=Txt3.getText();
 c=Integer.parseInt(str);
 if (a>=b && a>=c)
 {
 result=a;
 }
 else if(b>=a && b>=c)
 {
 result=b;
 }
 else
 {
 result=c;
 }
 g.drawString("Largest number is "+result,10,70);
 }
 public boolean action(Event e, Object o){
 repaint();
 return true;
 }
}
```

OUTPUT



```
Enter the numbers    67    77    63

Largest number is 77

Applet started.
```

RESULT

The program has been executed and the output was verified.

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

PROGRAM

Marks.java

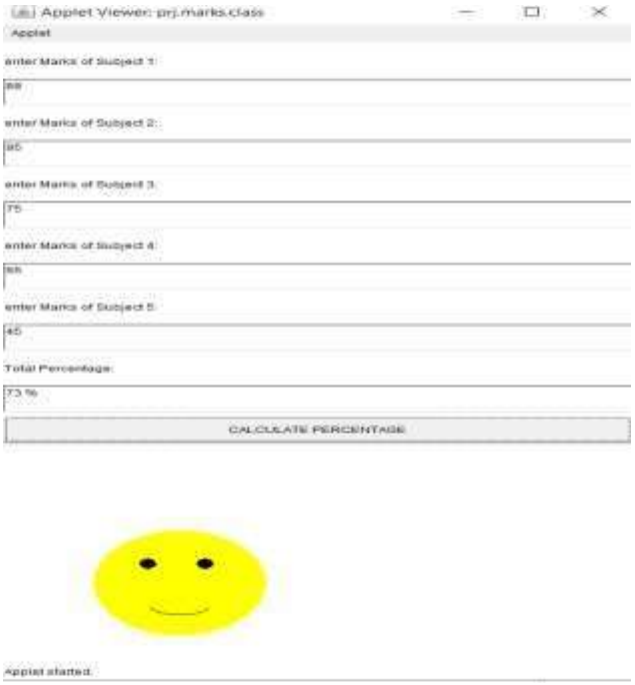
```
package prj;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class marks extends Applet implements ActionListener {
public int per =0;
Label l1 = new Label("enter Marks of Subject 1: ");
 Label l2 = new Label("enter Marks of Subject 2: ");
 Label l3 = new Label("enter Marks of Subject 3: ");
 Label l4 = new Label("enter Marks of Subject 4: ");
 Label l5 = new Label("enter Marks of Subject 5: ");
 Label l6 = new Label("Total Percentage: ");

 TextField t1 = new TextField(10);
 TextField t2 = new TextField(10);
 TextField t3 = new TextField(10);
 TextField t4 = new TextField(10);
 TextField t5 = new TextField(10);
 TextField t6 = new TextField(10);

 Button b1 = new Button("CALCULATE PERCENTAGE");
 public marks()
 {
```

```
 l1.setBounds(50, 100, 280, 20);
 l2.setBounds(50, 150, 280, 20);
 l3.setBounds(50, 200, 280, 20);
 l4.setBounds(50, 250, 280, 20);
 l5.setBounds(50, 300, 280, 20);
 l6.setBounds(50, 350, 280, 20);
 t1.setBounds(200, 100, 300, 20);
 t2.setBounds(200, 150, 300, 20);
 t3.setBounds(200, 200, 300, 20);
 t4.setBounds(200, 250, 300, 20);
 t5.setBounds(200, 300, 300, 20);
 t6.setBounds(200, 350, 300, 20);
 b1.setBounds(200,400, 200, 20);
 GridLayout g1 = new GridLayout(20, 2, 5, 5);
 setLayout(g1);
 add(l1);
 add(t1);
 add(l2);
 add(t2);
 add(l3);
 add(t3);
 add(l4);
 add(t4);
 add(l5);
 add(t5);
 add(l6);
 add(t6);
 add(b1);
 b1.addActionListener(this);
 }
@Override
public void actionPerformed(ActionEvent e) {
 int m1 = Integer.parseInt(t1.getText());
 int m2= Integer.parseInt(t2.getText());
 int m3= Integer.parseInt(t3.getText());
 int m4= Integer.parseInt(t4.getText());
 int m5= Integer.parseInt(t5.getText());

 if(e.getSource()==b1)
 {
 int add=m1+m2+m3+m4+m5;
 per=add/5;
 t6.setText(String.valueOf(per)+" %");

 repaint();
 }
}
public void paint(Graphics g)
 {
```

```
if(per>=50)
{
g.setColor(Color.yellow);
g.drawOval(80, 700, 150, 150);
g.fillOval(80, 700, 150, 150);
g.setColor(Color.BLACK);
g.fillOval(120, 740, 15, 15);
g.fillOval(170, 740, 15, 15);
g.drawArc(130, 800, 50, 20, 180, 180);
}
else if(per>0 && per<50)
{
g.setColor(Color.yellow);
g.drawOval(80, 700, 150, 150);
g.fillOval(80, 700, 150, 150);
 g.setColor(Color.BLACK);
 g.fillOval(120, 740, 15, 15);
 g.fillOval(170, 740, 15, 15);
g.drawArc(130,820,50,20,0,180);
 }
 }
public static void main(String args[]) {
 new marks();
 }
}
```

OUTPUT

Case 1



Applet Viewer: prj.marks.class

Applet

enter Marks of Subject 1: 88

enter Marks of Subject 2: 95

enter Marks of Subject 3: 75

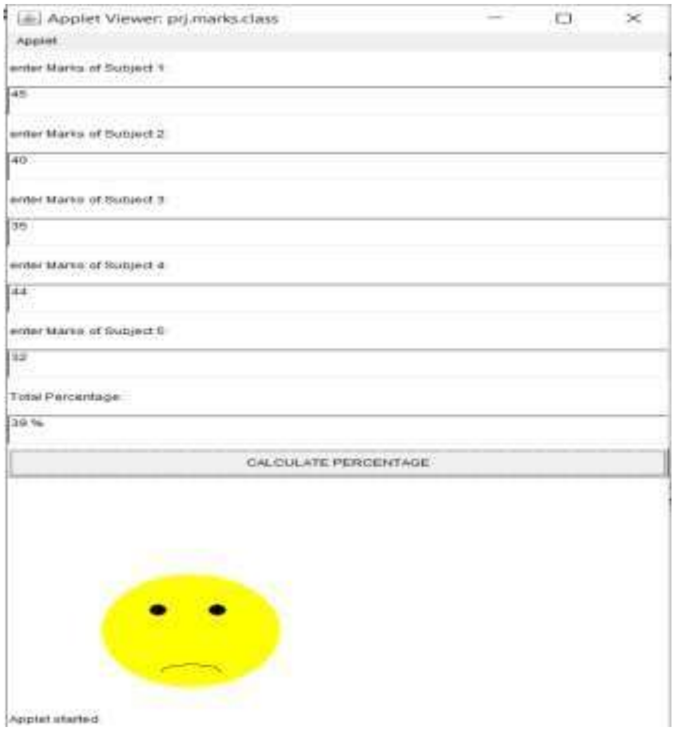
enter Marks of Subject 4: 65

enter Marks of Subject 5: 45

Total Percentage: 73 %

Applet started.

Case 2



enter Marks of Subject 1
45
enter Marks of Subject 2
40
enter Marks of Subject 3
35
enter Marks of Subject 4
44
enter Marks of Subject 5
32
Total Percentage
39 %

Applet started

RESULT

The program has been executed and the output was verified.

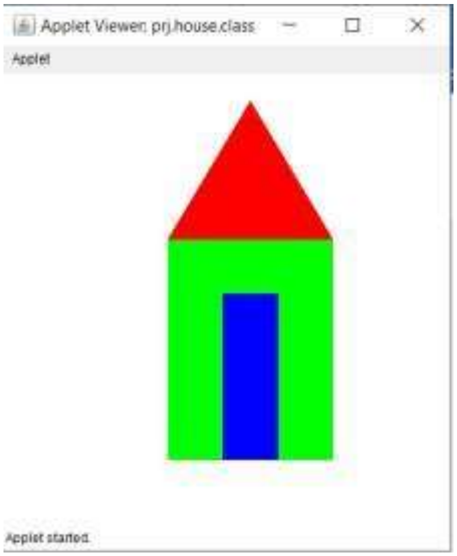
4. Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

PROGRAM

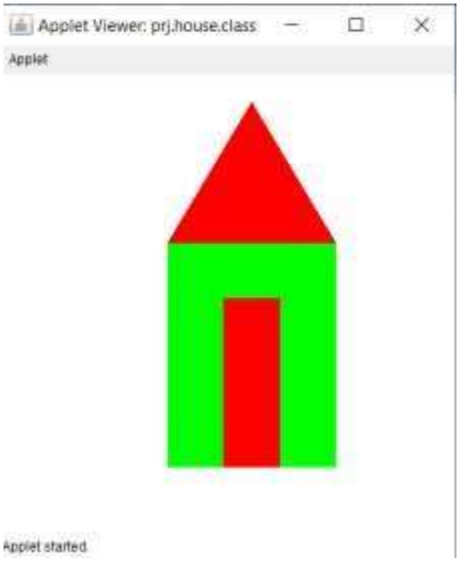
House.java

```
package prj;
import java.applet. *;
import java.awt. *;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
public class house extends Applet implements MouseListener,Runnable
{
private Color door=Color.blue;
public void paint (Graphics g)
{
int x[]= {150,300,225};
int y[]= {150,150,25};
g.setColor(Color.green);
g.fillRect(150, 150, 150, 200);
g.drawRect(150,150,150,200);
g.setColor(door);
g.fillRect(200, 200, 50, 150);
g.drawRect(200,200,50, 150);
g.setColor(Color.red);
g.fillPolygon(x,y,3);
g.drawPolygon(x,y,3);
}
public void init()
{
this.setSize(200,200);
addMouseListener(this);
}
public void run()
{
while(true)
{
repaint();
try
{
Thread.sleep(5);
}
catch(InterruptedException e)
{
e.printStackTrace();
}
}
}
public void mouseClicked(MouseEvent e)
{
int x=e.getX(), y=e.getY();
if(x<=300)
door=Color.red;
else
door=Color.blue;
repaint();
}
public void mousePressed(MouseEvent e){}
public void mouseReleased(MouseEvent e){}
public void mouseEntered(MouseEvent e){}
public void mouseExited(MouseEvent e){}
}
```

OUTPUT



Applet started.



Applet started.

RESULT

The program has been executed and the output was verified.

5. Implement a simple calculator using AWT components.

PROGRAM

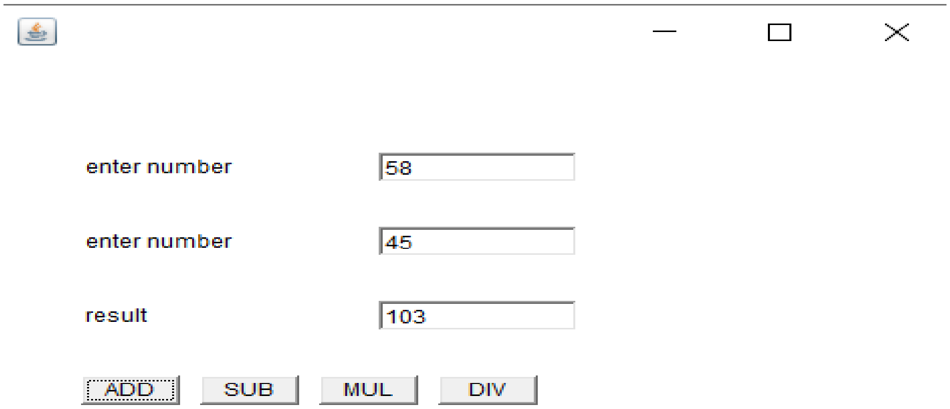
Calc.java

```
package prj;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
```

```
public class calc extends Applet implements ActionListener {
Frame f = new Frame();
Label l1 = new Label("enter number");
Label l2 = new Label("enter number");
Label l3 = new Label("result");

TextField t1 = new TextField(10);
TextField t2 = new TextField(10);
TextField t3 = new TextField(10);

Button b1 = new Button("ADD");
Button b2 = new Button("SUB");
Button b3 = new Button("MUL");
Button b4 = new Button("DIV");
calc()
{
l1.setBounds(50, 100, 100, 20);
l2.setBounds(50, 150, 100, 20);
l3.setBounds(50, 200, 100, 20);
t1.setBounds(200, 100, 100, 20);
t2.setBounds(200, 150, 100, 20);
t3.setBounds(200, 200, 100, 20);
b1.setBounds(50, 250, 50, 20);
b2.setBounds(110, 250, 50, 20);
b3.setBounds(170, 250, 50, 20);
b4.setBounds(230, 250, 50, 20);
f.add(l1);
f.add(t1);
f.add(l2);
f.add(t2);
f.add(l3);
f.add(t3);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(b4);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
f.setLayout(null);
f.setVisible(true);
f.setSize(500, 500);
}
public void actionPerformed(ActionEvent e) {
int i = Integer.parseInt(t1.getText());
int j = Integer.parseInt(t2.getText());
if (e.getSource() == b1) {
```

```
t3.setText(String.valueOf(i + j));
 }
 if (e.getSource() == b2) {
 t3.setText(String.valueOf(i - j));
 }
 if (e.getSource() == b3) {


 t3.setText(String.valueOf(i * j));
 }
 if (e.getSource() == b4) {
 t3.setText(String.valueOf(i / j));
 }
 }

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

OUTPUT



```
enter number    58
enter number    45
result          103
```

RESULT

The program has been executed and the output was verified.

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

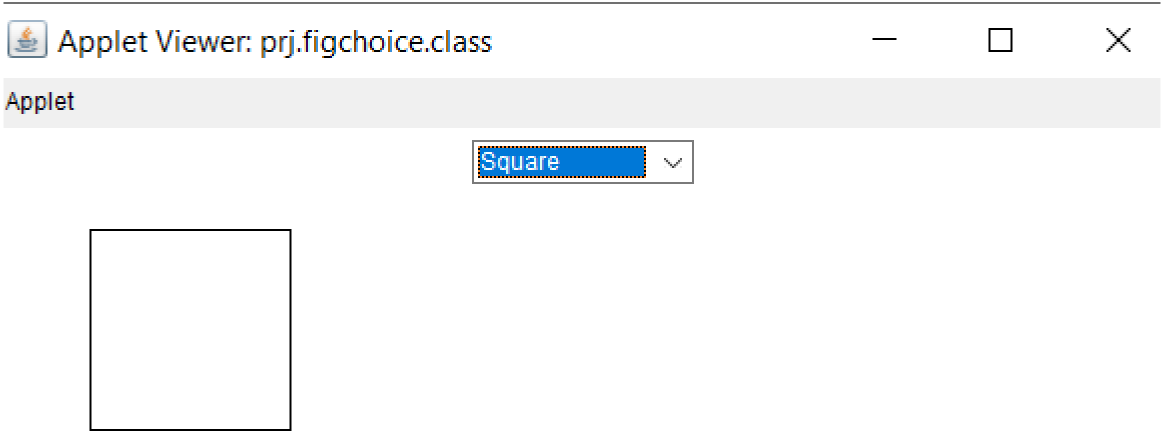
PROGRAM

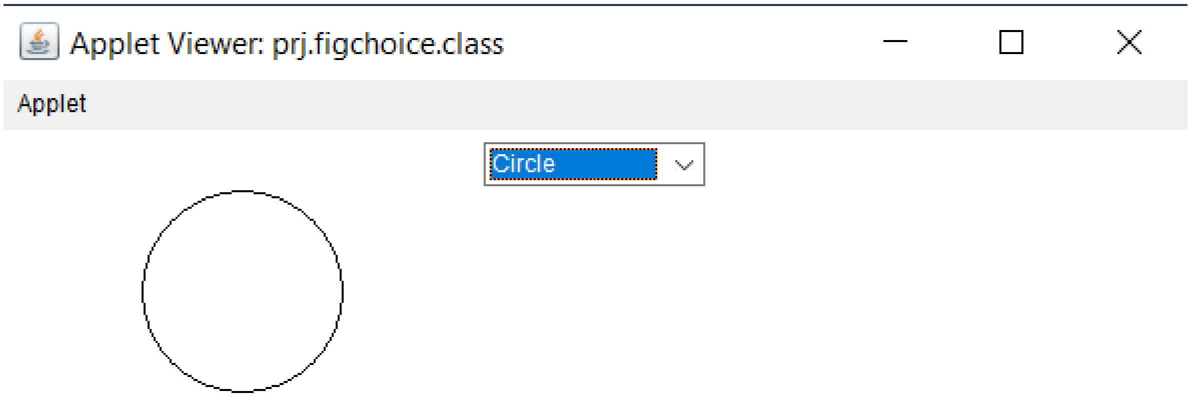
Figchoice.java

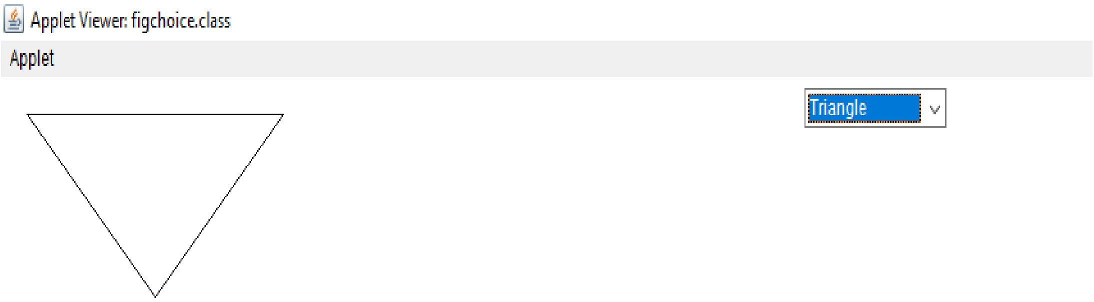
```
package prj;
import java.applet.Applet;
import java.awt.*;
import java.awt.Graphics;
import java.awt.event.*;
```

```
 public class figchoice extends Applet implements ItemListener {
 Choice ch;
 int x1[]= {50,120,220,20};
int y1[]= {50,120,20,20};
int n=4;
 int Selection;
 public void init()
 {
 ch = new Choice();
 ch.addItem("Select a Shape");

 ch.addItem("Rectangle");
 ch.addItem("Triangle");
 ch.addItem("Square");
 ch.addItem("Circle");
 add(ch);
 ch.addItemListener(this);
 }
public void itemStateChanged (ItemEvent e)
 {
 Selection = ch.getSelectedIndex();
 repaint();
 }
public void paint(Graphics g)
{
 super.paint(g);
 if (Selection == 1)
 {
 g.drawRect(50,50,100,150);
 }
 if (Selection == 2)
 {
 g.drawPolygon(x1,y1,n);
 }
 if (Selection == 3)
 {
 g.drawRect(50,50,100,100);
 }
 if (Selection == 4)
 {
 g.drawOval(70,30,100,100);
 }
}
}
```

OUTPUT







RESULT

The program has been executed and the output was verified.

7. Develop a program to handle all window events,

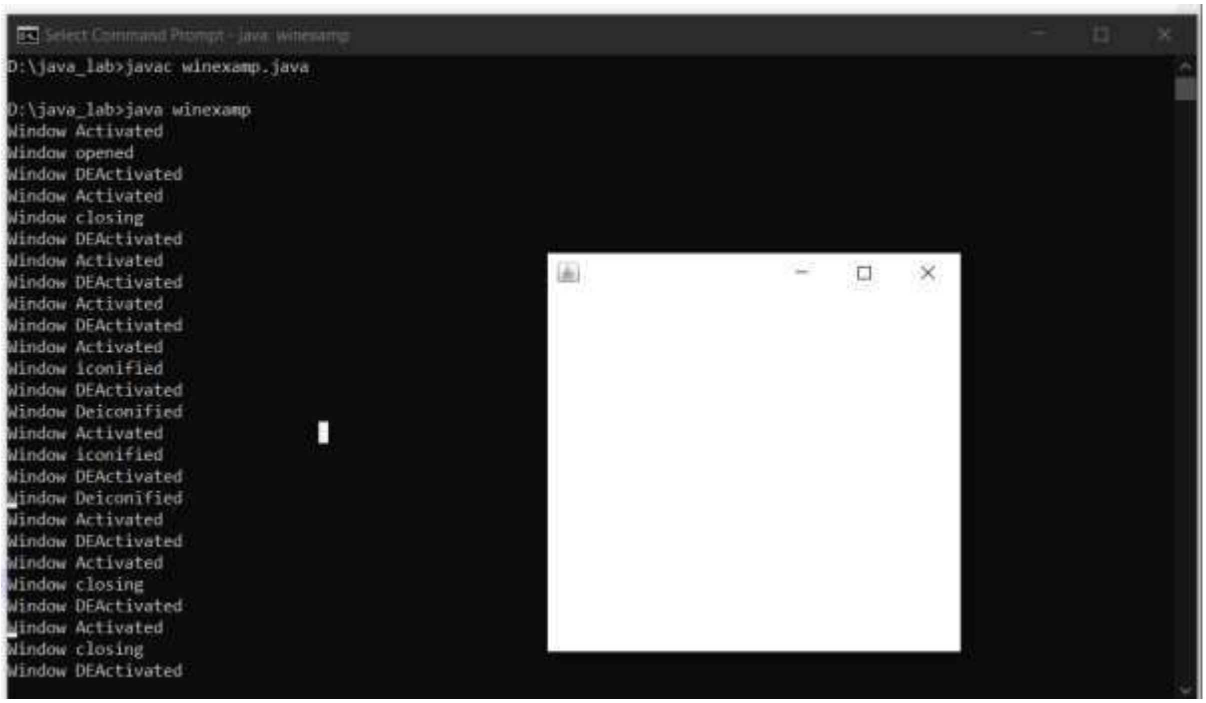
PROGRAM

winexamp.java

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

```
public class winexamp extends Frame implements WindowListener
{
winexamp()
{
addWindowListener(this);
setSize(400,400);
setLayout(null);
setVisible(true);
}
public static void main(String args[])
{
new winexamp();
}
public void windowActivated(WindowEvent arg0)
{
System.out.println("Window Activated");
}
public void windowClosed(WindowEvent args0)
{
System.out.println("Window closed");
}
public void windowClosing(WindowEvent arg0)
{
System.out.println("Window closing");
}
public void windowDeactivated(WindowEvent arg0)
{
System.out.println("Window DEActivated");
}
public void windowDeiconified(WindowEvent arg0)
{
System.out.println("Window Deiconified");
}
public void windowIconified(WindowEvent arg0)
{
System.out.println("Window iconified");
}
public void windowOpened(WindowEvent arg0)
{
System.out.println("Window opened");
}
}
```

OUTPUT



Select Command Prompt - java winexamp

```
D:\java_lab>javac winexamp.java

D:\java_lab>java winexamp
Window Activated
Window opened
Window DEActivated
Window Activated
Window closing
Window DEActivated
Window Activated
Window DEActivated
Window Activated
Window DEActivated
Window Activated
Window iconified
Window DEActivated
Window Deiconified
Window Activated
Window iconified
Window DEActivated
Window Deiconified
Window Activated
Window DEActivated
Window Activated
Window closing
Window DEActivated
Window Activated
Window closing
Window DEActivated
```

RESULT

The program has been executed and the output was verified.

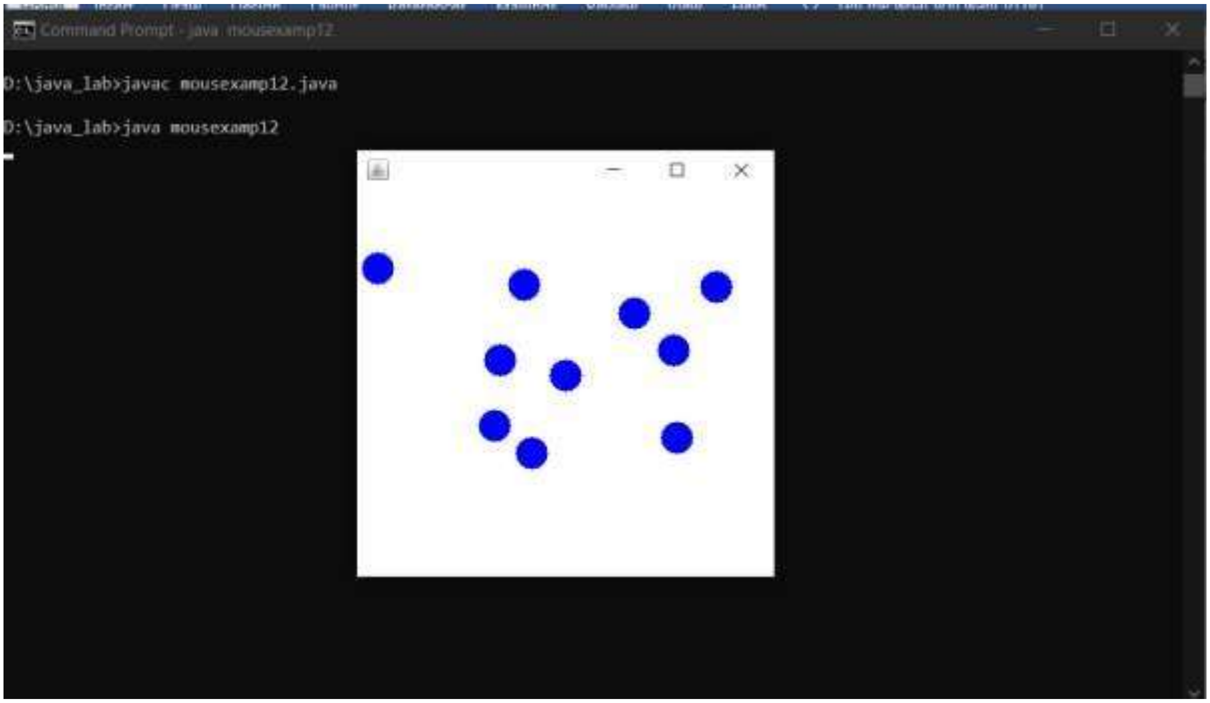
8. Develop a program to handle all mouse events .

PROGRAM

mousexamp12.java

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

OUTPUT



```
D:\java_lab>javac mousexamp12.java

D:\java_lab>java mousexamp12
```

RESULT

The program has been executed and the output was verified.

9. Develop a program to handle Key events.

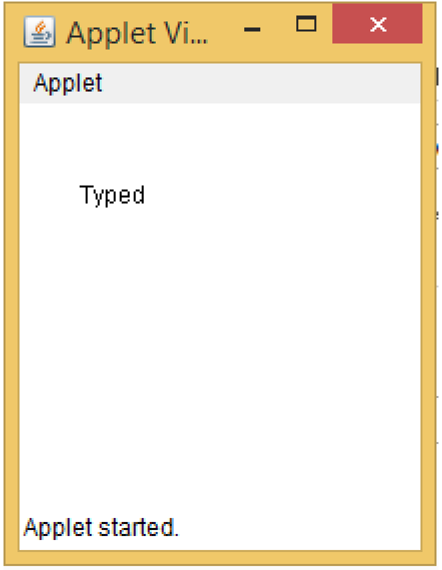
PROGRAM

Main.java

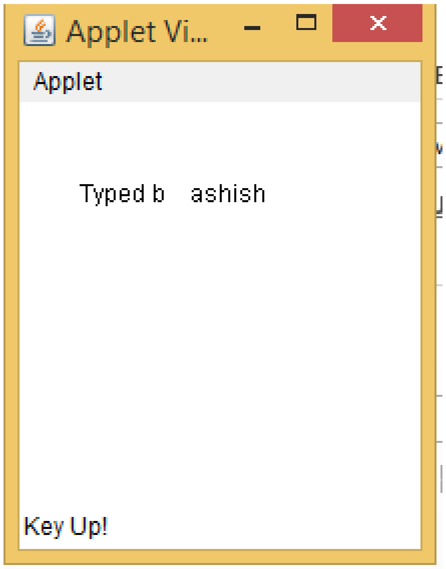
```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class Main extends Applet implements KeyListener
```

```
{
String msg="Typed";
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);
}
}
```

OUTPUT



```
Typed

Applet started.
```



```
Typed b   ashish

Key Up!
```

RESULT

The program has been executed and the output was verified.

Course Outcome 6 (CO6):

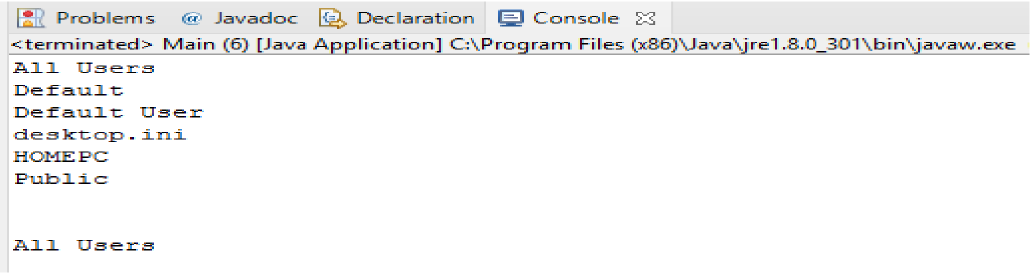
1.Program to list the sub directories and files in a given directory and also search for a file name .

PROGRAM

Main.java

```
import java.io.*;
public class Main {

public static void main(String[] args) {
        File file = new File("../../../");
        String[] fileList = file.list();
        for(String str : fileList) {
            System.out.println(str);
        }
        FilenameFilter filter = new FilenameFilter() {
            public boolean accept (File dir, String name) {
                return name.startsWith("A");
            }
        };
        System.out.println("\n");
        String[] children = file.list(filter);
        if (children == null) {
            System.out.println("Either dir does not exist or is not a directory");
        } else {
            for (int i = 0; i< children.length; i++) {
                String filename = children[i];
                System.out.println(filename);
            }
        }
    }
}
```

OUTPUT



```
<terminated> Main (6) [Java Application] C:\Program Files (x86)\Java\jre1.8.0_301\bin\javaw.exe
All Users
Default
Default User
desktop.ini
HOMEPC
Public


All Users
```

RESULT

The program has been executed and the output was verified.

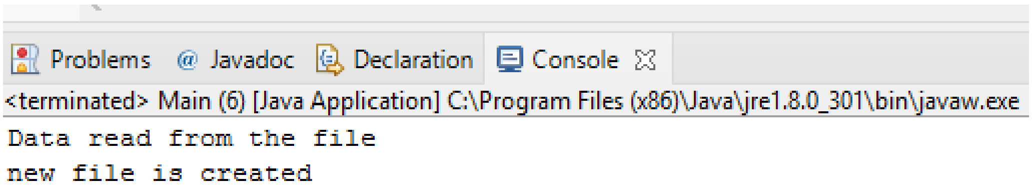
2. Write a program to write to a file, then read from the file and display the contents on the console.

PROGRAM

Main.java

```
import java.io.BufferedReader;
        import java.io.FileReader;
        import java.io.FileWriter;
        import java.io.IOException;

public class Main {

    public static void main(String[] args) {

        try {
            FileWriter writer = new FileWriter("java_write.txt",true);
            writer.write("new file is created");
            writer.close();
            FileReader reader = new FileReader("java_write.txt");
            BufferedReader br= new BufferedReader(reader);
            String line;
            System.out.println("Data read from the file");
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
            reader.close();

        } catch (IOException e) {
            System.out.println("-----Error ---- ");
        }

    }

}
```

OUTPUT



```
<terminated> Main (6) [Java Application] C:\Program Files (x86)\Java\jre1.8.0_301\bin\javaw.exe
Data read from the file
new file is created
```

RESULT

The program has been executed and the output was verified.