Object Oriented Programming

Lab Record

Submitted by

Aksa Anna Jose

Roll no:06

Regular MCA

Reg. no:12620

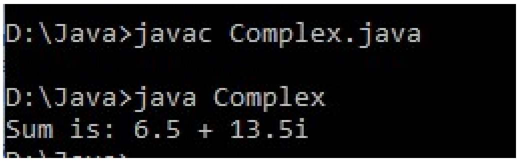
Program no:1

AIM: Add complex numbers

PROGRAM

```
public class Complex
{
 double real, img;
 Complex(double r, double i)
 {
 this.real = r;
 this.img= i;
 }
 public static Complex sum(Complex c1, Complex c2)
 {
 Complex temp = new Complex(0, 0);
 temp.real = c1. real+ c2.real;
 temp.img = c1.img+ c2.img;
 return temp;
 }
 public static void main(String args[])
 {
 Complex c1 = new Complex(2, 10);
 Complex c2 = new Complex(4.5, 3.5);
 Complex temp = sum(c1, c2);
 System.out.printf("Sum is: "+ temp.real+" + "+ temp.img +"i");
 }
}
```

Output



```
D:\Java>javac Complex.java

D:\Java>java Complex
Sum is: 6.5 + 13.5i
D:\Java>
```

Program no:2

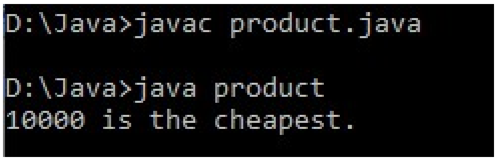
AIM: **Define a class product with data members pcode,pname,price. Create 3 objects of the class and find the product having the lowest price.**

PROGRAM

```
public class product
 {
 int pcode;
 String pname;
 int price;

 public static void main(String[] args)
 {
 int smallest;
 product p1 = new product();
 product p2 = new product();
 product p3 = new product();
 p1.pcode=2000;
 p1.pname="laptop";
 p1.price=10000;
 p2.pcode=1110;
 p2.pname="hp";
 p2.price=35000;
 p3.pcode=2002;
 p3.pname="intel i3";
 p3.price=40000;
 if(p1.price<p2.price)
 {
 if(p3.price<p1.price)
 {
 smallest = p3.price;
 } else {
 smallest = p1.price;
 }
 } else {
 if(p2.price<p3.price)
 {
```

```
smallest = p2.price;
} else {
smallest = p3.price;
}
}

System.out.println(smallest + " is the cheapest.");
}
}
```

Output



```
D:\Java>javac product.java

D:\Java>java product
10000 is the cheapest.
```

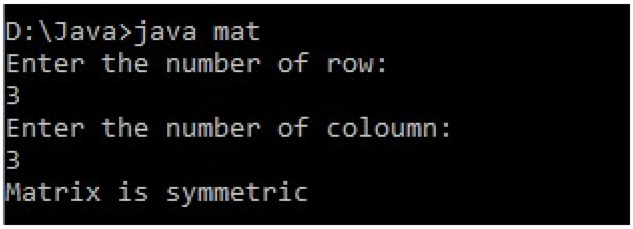
Program no:3

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

Program

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



```
D:\Java>java mat
Enter the number of row:
3
Enter the number of coloumn:
3
Matrix is symmetric
```

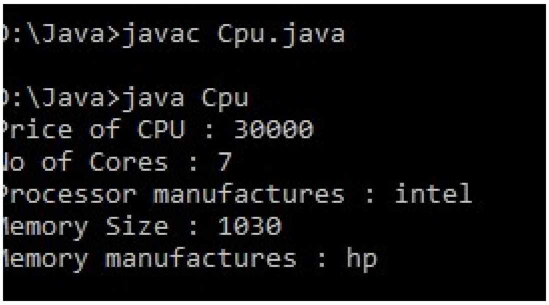
Program no:4

AIM: Create CPU with attribute price. create inner class processor (no.of cores , manufactures) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of processor and ram.

Program

```
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(30000);
Cpu.Processor i_processor = intel.new Processor(7, "intel");
Cpu.Ram i_ram = new Ram(1030, "hp");
intel.display();
i_processor.display();
i_ram.display();
}
}
```

Output



```
):\Java>javac Cpu.java

):\Java>java Cpu
'rice of CPU : 30000
Jo of Cores : 7
'rocessor manufactures : intel
lemory Size : 1030
lemory manufactures : hp
```

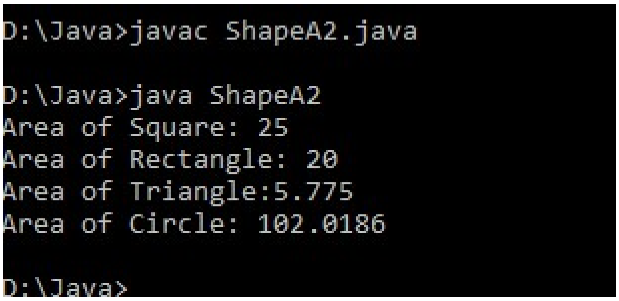
Program no:5

AIM: Area of different shapes using overload functions.

Program

```
public class ShapeA2
{
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(5));
System.out.println("Area of Rectangle: "+obj.area(5,4));
System.out.println("Area of Triangle:"+obj.area(5.5,2.1));
System.out.println("Area of Circle: "+obj.area(5.7));
}
}
```

Output



```
D:\Java>javac ShapeA2.java

D:\Java>java ShapeA2
Area of Square: 25
Area of Rectangle: 20
Area of Triangle:5.775
Area of Circle: 102.0186

D:\Java>
```

Program no:6

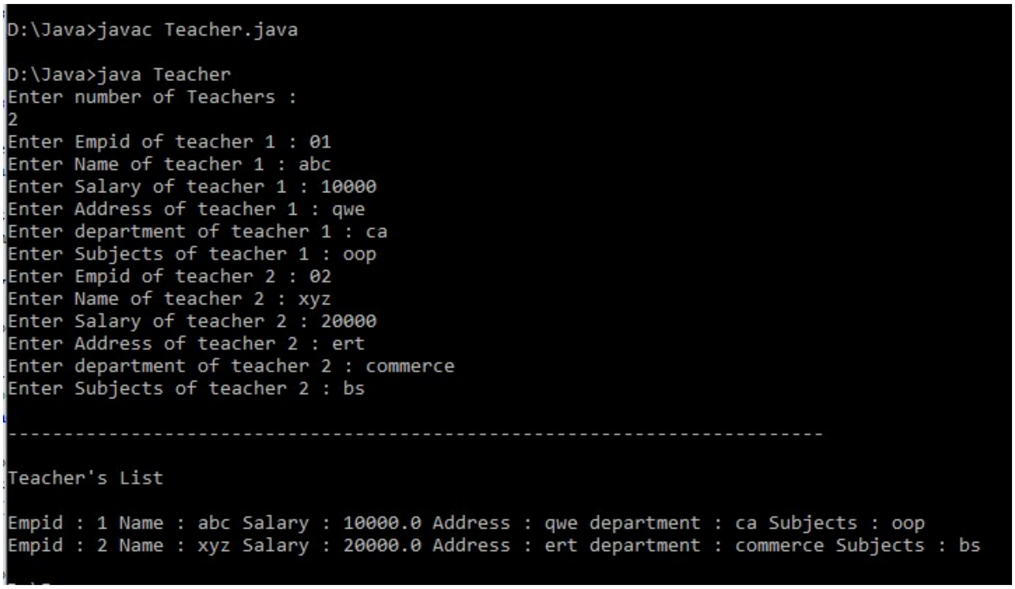
AIM:Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these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

Program

```
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, double salary,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) {
```

```
// TODO Auto-generated method stub
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



```
D:\Java>javac Teacher.java

D:\Java>java Teacher
Enter number of Teachers :
2
Enter Empid of teacher 1 : 01
Enter Name of teacher 1 : abc
Enter Salary of teacher 1 : 10000
Enter Address of teacher 1 : qwe
Enter department of teacher 1 : ca
Enter Subjects of teacher 1 : oop
Enter Empid of teacher 2 : 02
Enter Name of teacher 2 : xyz
Enter Salary of teacher 2 : 20000
Enter Address of teacher 2 : ert
Enter department of teacher 2 : commerce
Enter Subjects of teacher 2 : bs

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

Teacher's List

Empid : 1 Name : abc Salary : 10000.0 Address : qwe department : ca Subjects : oop
Empid : 2 Name : xyz Salary : 20000.0 Address : ert department : commerce Subjects : bs
```

Program no:7

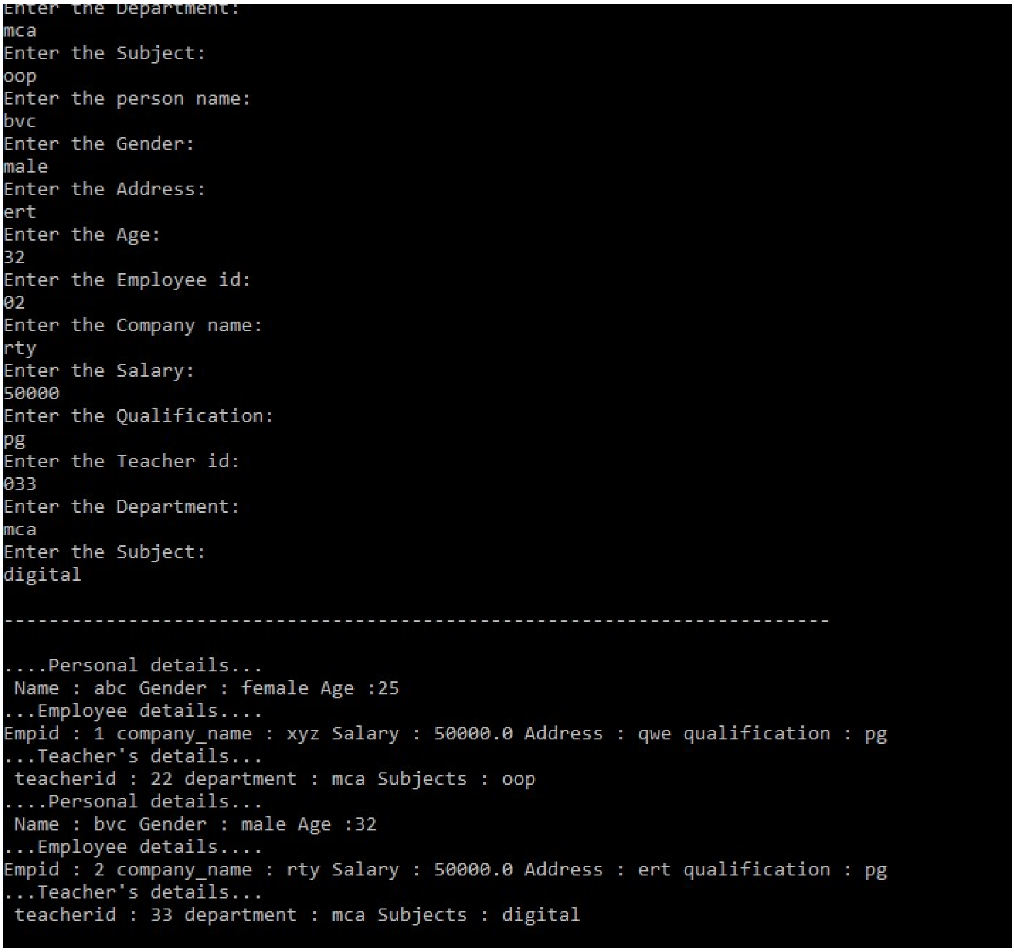
AIM: Create a class ‘Person’ with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class ‘Employee’ that inherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor.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

Program

```
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



```
Enter the Department:
mca
Enter the Subject:
oop
Enter the person name:
bvc
Enter the Gender:
male
Enter the Address:
ert
Enter the Age:
32
Enter the Employee id:
02
Enter the Company name:
rty
Enter the Salary:
50000
Enter the Qualification:
pg
Enter the Teacher id:
033
Enter the Department:
mca
Enter the Subject:
digital

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

....Personal details...
 Name : abc Gender : female Age :25
...Employee details....
Empid : 1 company_name : xyz Salary : 50000.0 Address : qwe qualification : pg
...Teacher's details...
 teacherid : 22 department : mca Subjects : oop
....Personal details...
 Name : bvc Gender : male Age :32
...Employee details....
Empid : 2 company_name : rty Salary : 50000.0 Address : ert qualification : pg
...Teacher's details...
 teacherid : 33 department : mca Subjects : digital
```

Program no:8

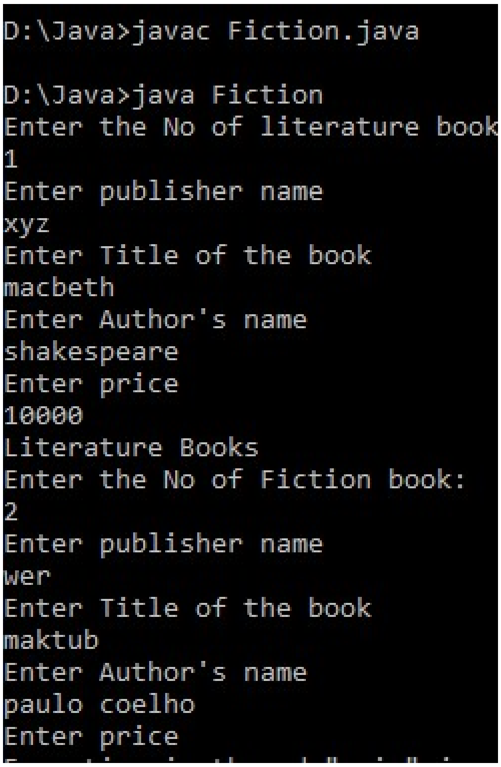
AIM: Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

Program

```
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



```
D:\Java>javac Fiction.java

D:\Java>java Fiction
Enter the No of literature book
1
Enter publisher name
xyz
Enter Title of the book
macbeth
Enter Author's name
shakespeare
Enter price
10000
Literature Books
Enter the No of Fiction book:
2
Enter publisher name
wer
Enter Title of the book
maktub
Enter Author's name
paulo coelho
Enter price
```

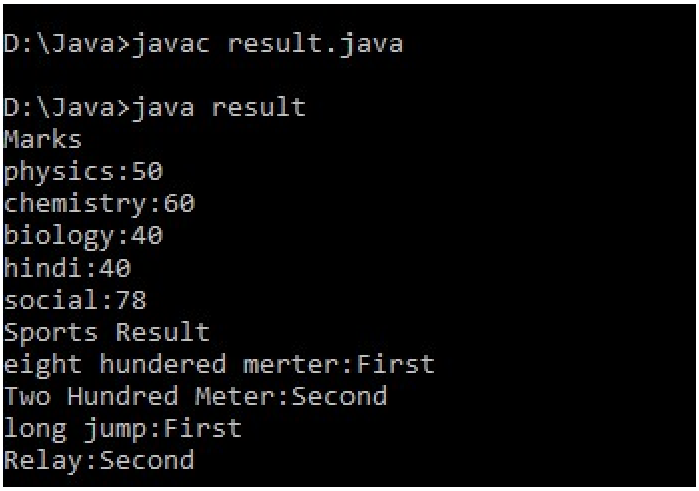
Program no:9

AIM: Create classes Student and sports.Create another class result inherited from student and sports.Display the academic and sports score of a student.

PROGRAM

```
interface student
{
void stresullt();
}
interface sports
{
void spresult();
}
class result implements student,sports
{
public void spresult()
{
String eighthundred="First";
String twohundred="Second";
String longjump="First";
String relay="Second";
System.out.println("Sports Result");
System.out.println("eight hundered merter:"+ eighthundred);
System.out.println("Two Hundred Meter:"+twohundred);
System.out.println("long jump:"+longjump);
System.out.println("Relay:"+relay);
}
public void stresullt()
{
int physics=50;
int chemistry=60;
int biology=40;
int hindi=40;
int social=78;
System.out.println("Marks");
System.out.println("physics:"+physics);
System.out.println("chemistry:"+chemistry);
System.out.println("biology:"+biology);
System.out.println("hindi:"+hindi);
System.out.println("social:"+social);
}
public static void main(String[] args)
{
result r = new result(); r.stresullt();
r.spresult();
}
}
```

Output



```
D:\Java>javac result.java

D:\Java>java result
Marks
physics:50
chemistry:60
biology:40
hindi:40
social:78
Sports Result
eight hundered merter:First
Two Hundred Meter:Second
long jump:First
Relay:Second
```

Program no:10

AIM: Create an interface having prototype 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.

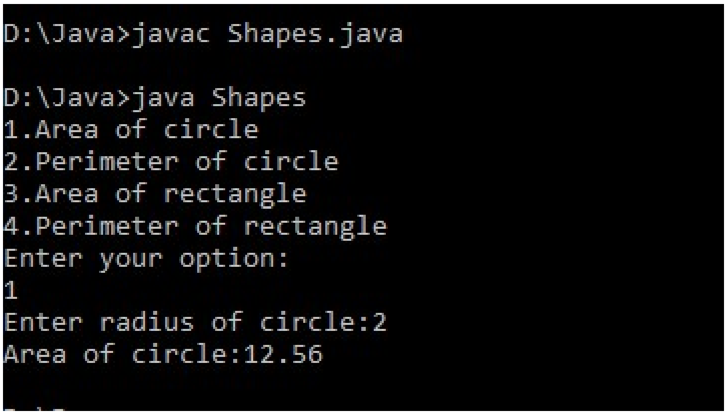
PROGRAM

```
import java.util.Scanner;
interface Shape
{
void input();
void area();
void perimeter();
}
class Circle implements Shape
{
int r = 0;
double pi = 3.14, ar = 0,per=0;
public void input()
{ Scanner s = new Scanner(System.in); System.out.print("Enter radius of circle:");
r= s.nextInt();
}
public void area()
{
ar = pi * r * r;
```

```
System.out.println("Area of circle:"+ar);
}
public void perimeter()
{
per = 2 * pi * r;
System.out.println("Perimeter of circle:"+per);
}
}
class Rectangle implements Shape
{
int l = 0, b = 0;
double ar,per;
public void input()
{ Scanner s = new Scanner(System.in);
System.out.print("Enter length of rectangle:");
l = s.nextInt();
System.out.print("Enter breadth of rectangle:");
b = s.nextInt();
}
public void area()
{
ar = l * b;
System.out.println("Area o}
public void perimeter()
```

```
{

per = 2 * (l + b);

System.out.println("Perimeter of rectangle:"+per);

}

}public class Shapes

{

public static void main(String[] args)

{ int n;

Scanner s = new Scanner(System.in);

Rectangle obj1 = new Rectangle();

Circle obj2 = new Circle(); System.out.println("1.Area of circle");

System.out.println("2.Perimeter of circle");

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

System.out.println("4.Perimeter of rectangle");

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

n= s.nextInt();

switch(n) {

case 1:

obj2.input();

obj2.area();

break;

case 2:

obj2.input();

obj2.perimeter();obj2.input();
obj2.area();

break;

case 4:

obj2.input();

obj2.perimeter();

break;

default:

System.out.println("Invalid option");

break;

case 3:

obj2.input();

obj2.area();

break;

case 4:

obj2.input();

obj2.perimeter();

break;

default:

System.out.println("Invalid option");

}

}

}
```

Output



```
D:\Java>javac Shapes.java

D:\Java>java Shapes
1.Area of circle
2.Perimeter of circle
3.Area of rectangle
4.Perimeter of rectangle
Enter your option:
1
Enter radius of circle:2
Area of circle:12.56
```

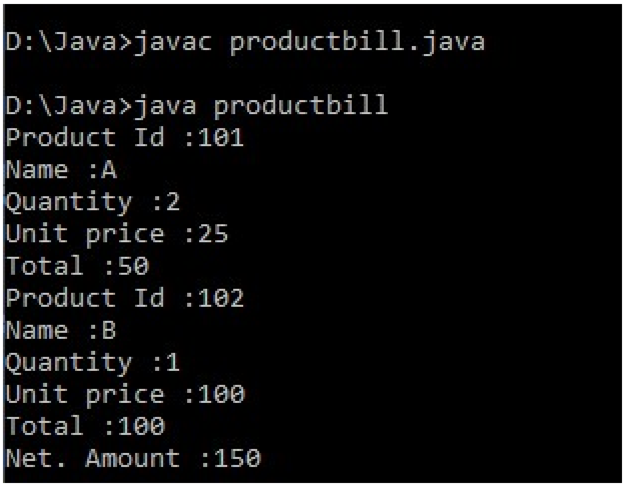
Program no:11

AIM:Prepare bill with the given format using calculate method from interface. Order No.Date Productid name quantity price total 101 A 2 25 50 102 B 1 100 100 Net.Amount 150

PROGRAM

```
interface bill
{
int productdetails();
}
class product1 implements bill
{
int id = 101,quantity= 2,unit=25,total=0; String name="A";
public int productdetails()
{
total = quantity * unit;
System.out.println("Product Id :"+id);
System.out.println("Name :"+name);
System.out.println("Quantity :"+quantity);
System.out.println("Unit price :"+unit);
System.out.println("Total :"+total);
return(total);
}
}
class product2 implements bill{
int id = 102,quantity= 1,unit=100,total=0;
String name="B";
public int productdetails()
{
total = quantity * unit;
System.out.println("Product Id :"+id);
System.out.println("Name :"+name);
System.out.println("Quantity :"+quantity);
System.out.println("Unit price :"+unit);
System.out.println("Total :"+total);
return(total);
}
}
public class productbill
{
public static void main(String[] args)
{
product1 p1 = new product1();
product2 p2 = new product2();
int t1= p1.productdetails();
int t2= p2.productdetails();
int t3=t1+t2;
System.out.println("Net. Amount :"+t3);
}
}
```

Output



```
D:\Java>javac productbill.java

D:\Java>java productbill
Product Id :101
Name :A
Quantity :2
Unit price :25
Total :50
Product Id :102
Name :B
Quantity :1
Unit price :100
Total :100
Net. Amount :150
```

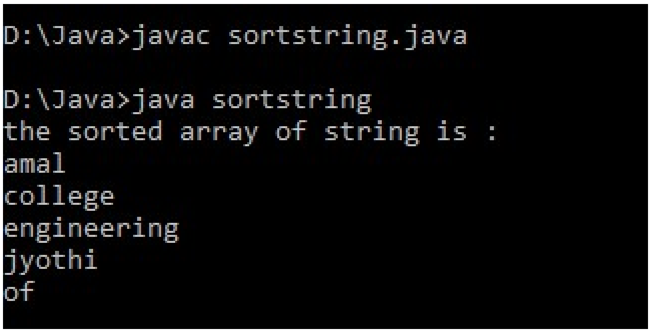
Program no:12

AIM: Program to sort string

PROGRAM

```
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



```
D:\Java>javac sortstring.java

D:\Java>java sortstring
the sorted array of string is :
amal
college
engineering
jyothi
of
```

Program no:13

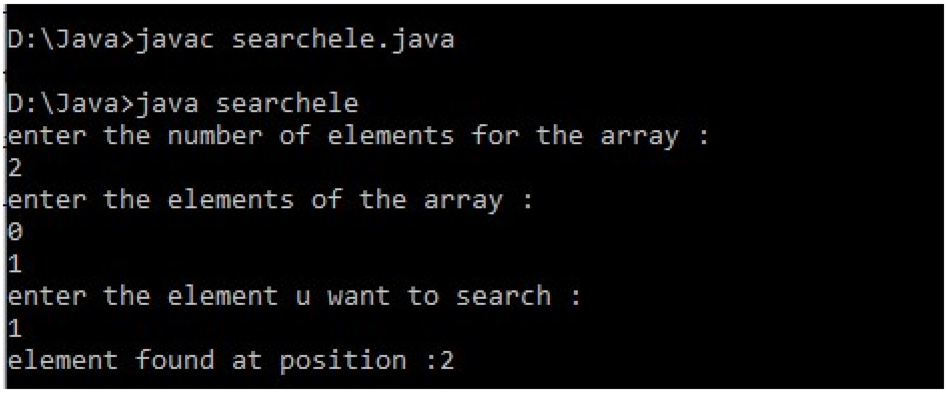
AIM: Search an element in an array

PROGRAM

```
  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



```
D:\Java>javac searchele.java

D:\Java>java searchele
enter the number of elements for the array :
2
enter the elements of the array :
0
1
enter the element u want to search :
1
element found at position :2
```

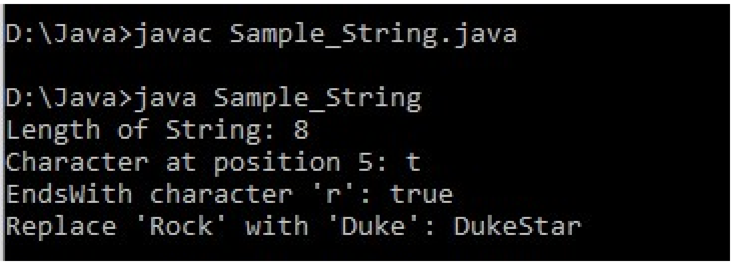
Program no:14

AIM: Perform string manipulation

PROGRAM

```
public class Sample_String

{

  public static void main(String[] args)

  {

  String str_Sample = "RockStar";

  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 'r': " + str_Sample.endsWith("r"));

  System.out.println("Replace 'Rock' with 'Duke': " + str_Sample.replace("Rock",
"Duke"));

}

}
```

Output



```
D:\Java>javac Sample_String.java

D:\Java>java Sample_String
Length of String: 8
Character at position 5: t
EndsWith character 'r': true
Replace 'Rock' with 'Duke': DukeStar
```

Program no:15

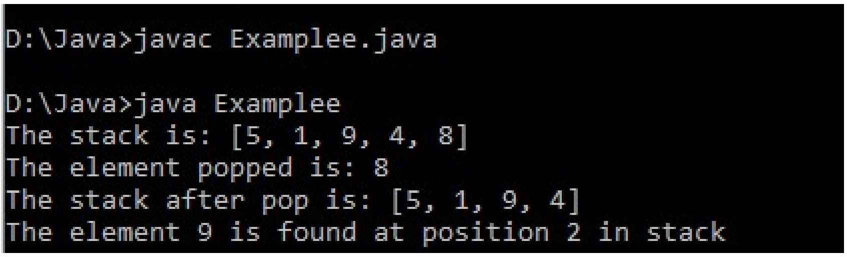
AIM: Java program to create generic stack and do the push and pop operation A stack class is provided by the Java collection framework and it implements the Stack data structure. The stack implements LIFO i.e. Last In First Out. This means that the elements pushed last are the ones that are popped first.

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

```
import java.io.*;
import java.util.*;
public class Examplee
{
 public static void main (String[] args)
 {
 Stack<Integer> s = new Stack<Integer>();
 s.push(5);
 s.push(1);
 s.push(9);
 s.push(4);
 s.push(8);
 System.out.print("The stack is: " + s);
 System.out.print("\nThe element popped is: ");
 Integer num1 = (Integer) s.pop();
 System.out.print(num1);
 System.out.print("\nThe stack after pop is: " + s);
 Integer pos = (Integer) s.search(9);
 if(pos == -1)
 System.out.print("\nThe element 9 not found in stack");
 else
 System.out.print("\nThe element 9 is found at position " + pos + " in stack");
 }
}
```

Output



```
D:\Java>javac Examplee.java

D:\Java>java Examplee
The stack is: [5, 1, 9, 4, 8]
The element popped is: 8
The stack after pop is: [5, 1, 9, 4]
The element 9 is found at position 2 in stack
```

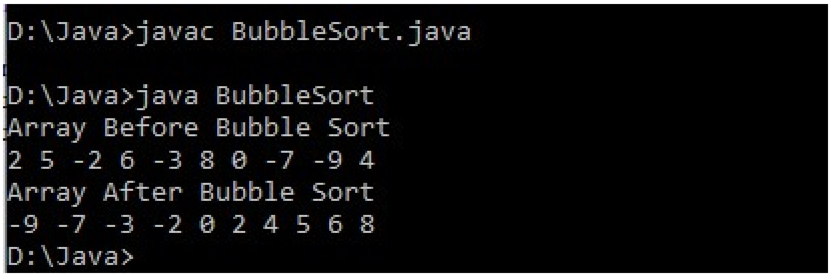
Program no:16

AIM: Generic method implement 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

```
public class BubbleSort

 {

static void bubbleSort(int[] arr)

 {

int n = arr.length;

int temp = 0;

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

for(int j=1; j < (n-i); j++) {

if(arr[j-1] > arr[j]) {

temp = arr[j-1];

arr[j-1] = arr[j];

arr[j] = temp;

}

}

}

}

public static void main(String[] args) {

int arr[] = { 2, 5, -2, 6, -3, 8, 0, -7, -9, 4 };

System.out.println("Array Before Bubble Sort");
for(int i = 0; i < arr.length; i++) {

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

}

System.out.println();

bubbleSort(arr);

System.out.println("Array After Bubble Sort");

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

{

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

}

}

}
```

Output



```
D:\Java>javac BubbleSort.java

D:\Java>java BubbleSort
Array Before Bubble Sort
2 5 -2 6 -3 8 0 -7 -9 4
Array After Bubble Sort
-9 -7 -3 -2 0 2 4 5 6 8
D:\Java>
```

Program no:17

AIM: Maintain a list of string using arraylist from a collection of framework, perform builtin operation 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.

<u>PROGRAM</u>

```
import java.util.*;

public class ArrayListDemo

{

 public static void main(String args[])

 {

 // create an array list

 ArrayList al = new ArrayList();

 System.out.println("Initial size of al: " + al.size());

 // add elements to the array list

 al.add("C");

 al.add("A");

 al.add("E");

 al.add("B");

 al.add("D");

 al.add("F");

 al.add(1, "A2");

 System.out.println("Size of al after additions: " + al.size());
 // display the array list
  System.out.println("Contents of al: " + al);
 // Remove elements from the array list
 al.remove("F");
 al.remove(2);
 System.out.println("Size of al after deletions: " + al.size());
 System.out.println("Contents of al: " + al);
 }
}
```

Output

```
Initial  size of al:0
Size of  al  after  addition:7
Contents of al:[C, A2, A,E, B,D,F]
Size of  al after  deletion:5
Contents  of al:[C,A2,E,B,D]
```

Program no:18

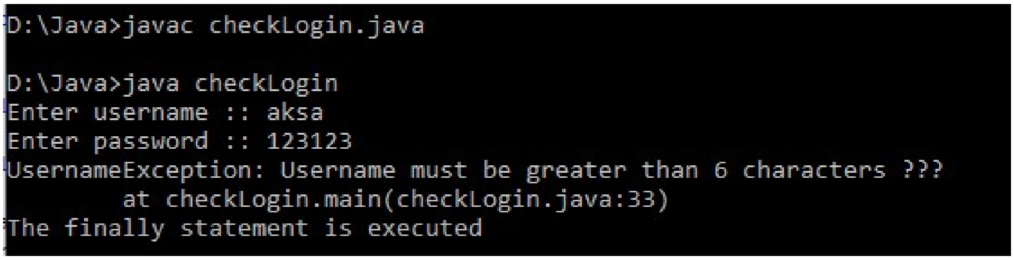
AIM: Write a user defined exception class to authentication the username and password.

PROGRAM

```
import java.util.Scanner;
class UsernameException extends Exception
 {
public UsernameException(String msg)
 {
 super(msg);
}
}
class PasswordException extends Exception
{
public PasswordException(String msg)
 {
 super(msg);
}
}
public class checkLogin
 {
public static void main(String[] args)
 {
 Scanner s = new Scanner(System.in);
 String username, password;
 System.out.print("Enter username :: ");
 username = s.nextLine();

 System.out.print("Enter password :: ");
 password = s.nextLine();

 int length = username.length();

 try
 {
 if(length < 6)
 throw new UsernameException("Username must be greater than 6 characters
???");
```

```
 else if(!password.equals("hello"))
 throw new PasswordException("Incorrect password\nType correct password
???");
 else
 System.out.println("Login Successful !!!");
 }
 catch (UsernameException u)
 {
 u.printStackTrace();
 }
 catch (PasswordException p)
 {
 p.printStackTrace();
 }
 finally {
 System.out.println("The finally statement is executed");
 }
}
}
```

Output



```
D:\Java>javac checkLogin.java

D:\Java>java checkLogin
Enter username :: aksa
Enter password :: 123123
UsernameException: Username must be greater than 6 characters ???
        at checkLogin.main(checkLogin.java:33)
The finally statement is executed
```

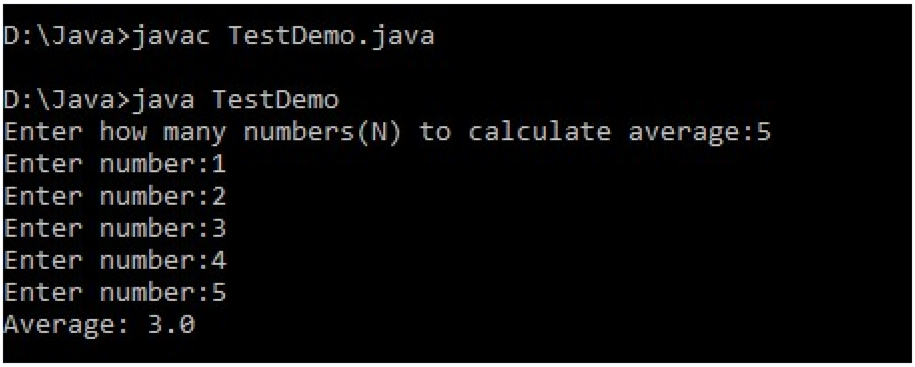
Program no:19

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

PROGRAM

```
import java.util.Scanner;
import java.util.InputMismatchException;
public class TestDemo
{
public static void main(String args[])
{
double total = 0, N, userInput;
Scanner input = new Scanner(System.in);
while (true)
{
System.out.print("Enter how many numbers(N) to calculate average:");
userInput = input.nextDouble();
if (userInput > 0)
{
N = userInput;
break;
}
else
System.out.println("N must be positive.");
}
for (int i = 0; i < N; i++)
{
while (true)
{
System.out.print("Enter number:");
try
{
userInput = input.nextDouble();
total += userInput;
break;
}
catch (InputMismatchException e)
{
input.nextLine();
System.out.println("Input must bea number. Try again");
}
}
}
System.out.println("Average: "+ total / N);
}
}
```

Output



```
D:\Java>javac TestDemo.java

D:\Java>java TestDemo
Enter how many numbers(N) to calculate average:5
Enter number:1
Enter number:2
Enter number:3
Enter number:4
Enter number:5
Average: 3.0
```

Program no:20

AIM: Define 2 classes one for generating multiplication table of 5 and other for displaying first N prime numbers implement using threads(thread class)

PROGRAM

```
class ThreadA extends Thread
{
 public void run()
 {
 int n = 5;
 for (int i = 1; i <= 10; ++i)
 System.out.println(n + " * " + i +
 " = " + n * i);
 System.out.println("Exiting from Thread A ...");
 }
}
class ThreadB extends Thread
{
 public void run( )
 {
 Scanner sc = new Scanner(System.in);
int i,n,p,count,flag;
System.out.println("Enter the number of prime terms you want!");
 n=sc.nextInt();
 System.out.println("First "+n+" prime numbers are :-");
p=2;
 i=1;
 while(i<=n)
 {
 flag=1;
 for(count=2;count<=p-1;count++)
 {
 if(p%count==0) //Will be true if p is not prime
 {
 flag=0;
 break; //Loop will terminate if p is not prime
 }
 }
  if(flag==1)
 {
 System.out.print(p+" ") ;
 i++;
 }
 p++;
 }
 }
 System.out.println("Exiting from Thread B ...");
}
  }
public class Demonstration_111
 {
 public static void main(String args[])
 {
 ThreadA a = new ThreadA();
 ThreadB b = new ThreadB();
 a.start();
 b.start();
 System.out.println("... Multithreading is over ");
 }
}
```

Output

```
5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
5*10=50
Enter the number of prime terms you want
4
First  4 prime numbers are :-
2 3 5 7
```

}

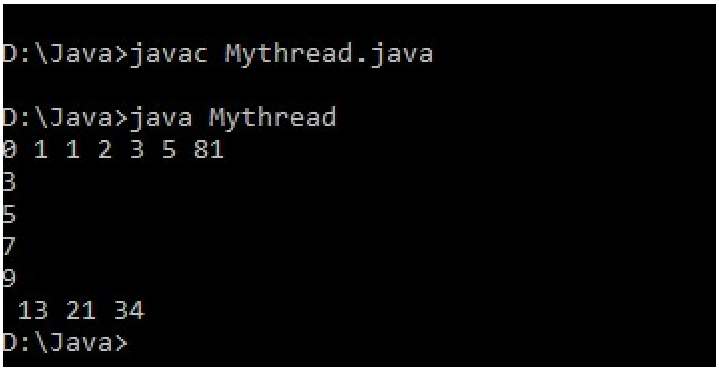
Program no:21

AIM: Define 2 classes one for generating fibanocci numbers and other for displaying even numers in a given range.implement using threads(runnable interface)

PROGRAM

```
public class Mythread

{

 public static void main(String[] args)

 {

 Runnable r = new Runnable1();

 Thread t = new Thread(r);

 t.start();

 Runnable r2 = new Runnable2();

 Thread t2 = new Thread(r2);

 t2.start();

  }

}

class Runnable2 implements Runnable

{

 public void run(){

 for(int i=0;i<11;i++){

 if(i%2 == 1)

 System.out.println(i);

 }

 }
}

class Runnable1 implements Runnable

{

 public void run(){

 int n1=0,n2=1,n3,i,count=10;

System.out.print(n1+" "+n2);//printing 0 and 1


for(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are already printed

{

 n3=n1+n2;

 System.out.print(" "+n3);

 n1=n2;

 n2=n3;

}

 }

 }
```

Output



```
D:\Java>javac Mythread.java

D:\Java>java Mythread
0 1 1 2 3 5 81
3
5
7
9
 13 21 34
D:\Java>
```

Program no:22

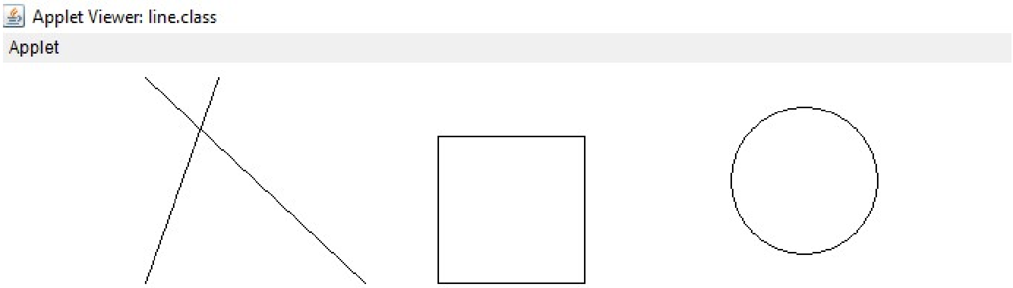
AIM: Program to draw circle,rectangle,line in applet

PROGRAM

```
import java.awt.*;
import java.applet.*;
public class line extends Applet
{
public void paint(Graphics g)
{
 g.drawLine(100,10,250, 150);
 g.drawLine(100,150,150,10);
 g.setColor(Color.black);
 g.drawRect(300, 50, 100, 100);
 g.setColor(Color.black);
 g.drawOval(500,30,100,100);
}
}
<html>
 <head>
 </head>
 <body>
 <applet code = "line.class" width = "420" height = "320"></applet>
 </body>
```

```
</html>
```

Output



Program no:23

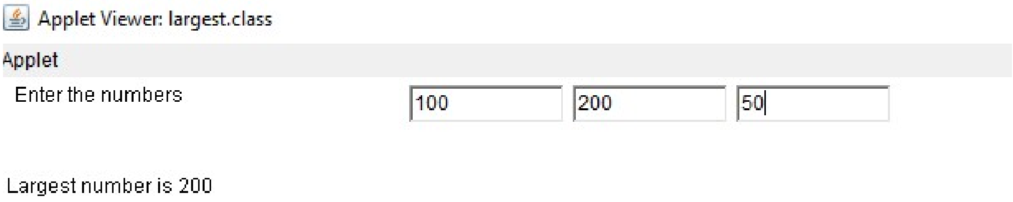
AIM: Program to find maximum of three numbers using AWT

PROGRAM

```
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;

}

}

<html>
```

```
<head>

</head>

<body>

<div align="center">

<applet code="largest.class"width="800"height="500">

</applet>

</div>

</body>

</html>
```

Output



Applet Viewer: largest.class

Enter the numbers 100 200 50

Largest number is 200

Program no:24

AIM: Find the percentage of marks obtained by a student in 5 subject. Display a Happy face if he secures above 50% or sad face if otherwise.

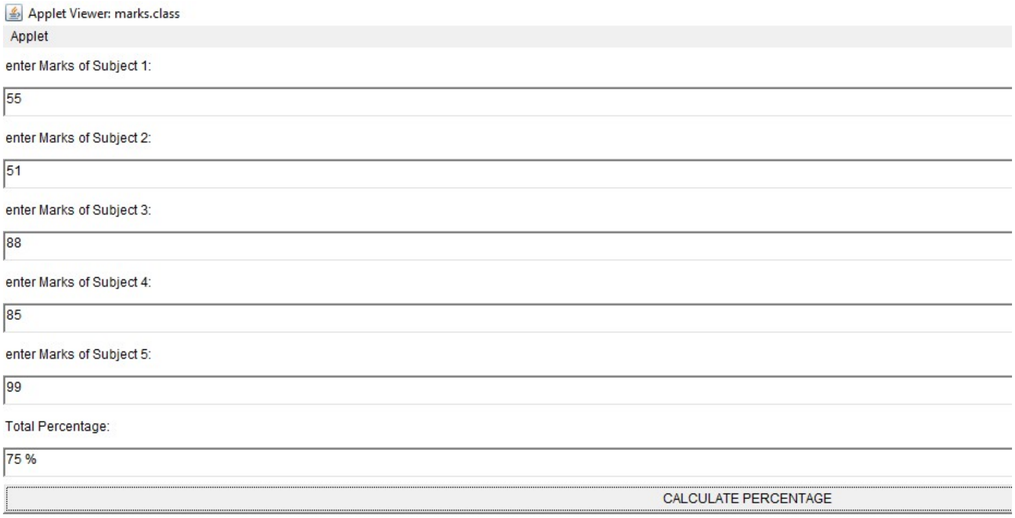
PROGRAM

```
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)

{

// TODO Auto-generated method stub

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.fillOval(120, 740, 15, 15);
g.fillOval(170, 740, 15, 15);
g.drawArc(130, 800, 50, 20, 180, 180);
int x[] = {95,85,106,95};
int y[] = {85,104,104,85};


}
else if(per>0 && per<50)
{
g.setColor(Color.yellow);
g.drawOval(80, 700, 150, 150);
g.fillOval(80, 700, 150, 150);


g.fillOval(120, 740, 15, 15);
g.fillOval(170, 740, 15, 15);
g.drawArc(130,820,50,20,0,180);
int x[] = {95,85,106,95};

int y[] = {85,104,104,85};


}

}

public static void main(String args[])

{

new marks();

}

}

<html>

<head>

</head>

<body>

<div align="center">

<applet code="marks.class"width="800"height="500">

</applet>

</div>

</body>

</html>
```

Output



enter Marks of Subject 1: 55

enter Marks of Subject 2: 51

enter Marks of Subject 3: 88

enter Marks of Subject 4: 85

enter Marks of Subject 5: 99

Total Percentage: 75 %

Program no:25

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

PROGRAM

```
import java.awt.*;

import java.applet.*;

import java.awt.event.*;

public class house extends Applet implements MouseListener, Runnable

 {

private Color doorColor = Color.WHITE;

public void paint(Graphics gp)

 {

int[] i = { 150, 300, 225 };

int[] j = { 150, 150, 25 };

gp.drawRect(150, 150, 150, 200);

gp.drawOval(200, 75, 50, 50);

gp.drawPolygon(i, j, 3);

gp.setColor(doorColor);

gp.fillRect(200, 200, 50, 150);

gp.setColor(Color.BLACK);

gp.drawRect(200, 200, 50, 150);

}

public void init()

 {
```

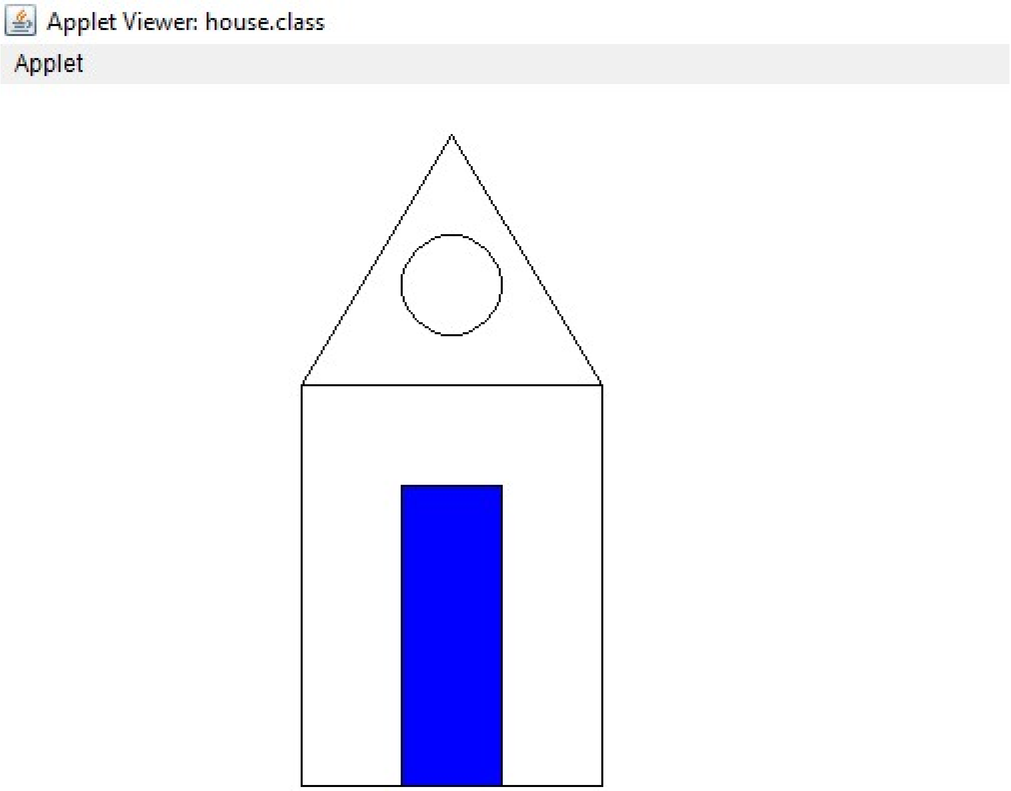
```
 this.setSize(200, 200);

addMouseListener(this);

}

public void run()

 {

while (true)

 {

repaint();

try {

Thread.sleep(17);

} catch (InterruptedException e)

{

e.printStackTrace();

}

}

}

public void mouseClicked(MouseEvent e)

 {

int x = e.getX(), y = e.getY();

if (x >= 200 && x <= 250 && y >= 200 && y <= 350)

doorColor = Color.RED;

else

doorColor = Color.BLUE;

repaint();
```

```
System.out.println("Mouse Position : X=" + x + " Y=" + y + "");
}
public void mousePressed(MouseEvent e)
 {
}
public void mouseReleased(MouseEvent e)
 {
}
public void mouseEntered(MouseEvent e)
{
}
public void mouseExited(MouseEvent e)
 {
}
}
```

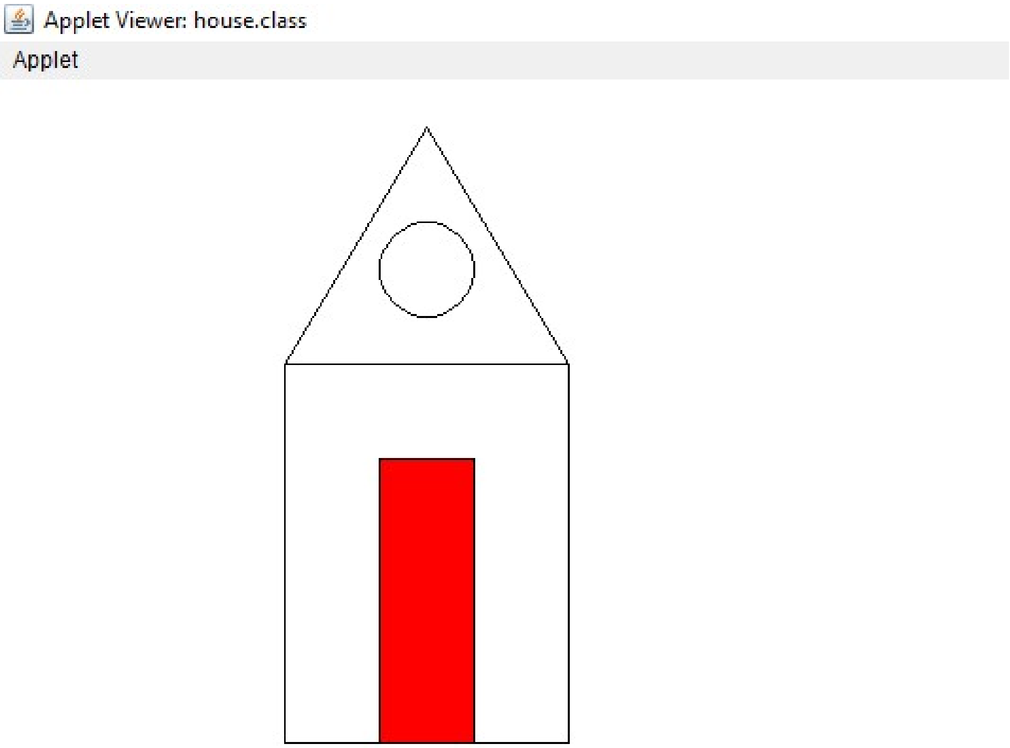
```
<html>
<head>
</head>
<body>
<div align="center">
<applet code="house.class" width="800" height="500">
</applet>
</div>
</body>
```

```
</html>
```

Output



Applet Viewer: house.class
Applet


Applet Viewer: house.class
Applet

Program no:26

AIM:Implement a simple calculator using AWT components

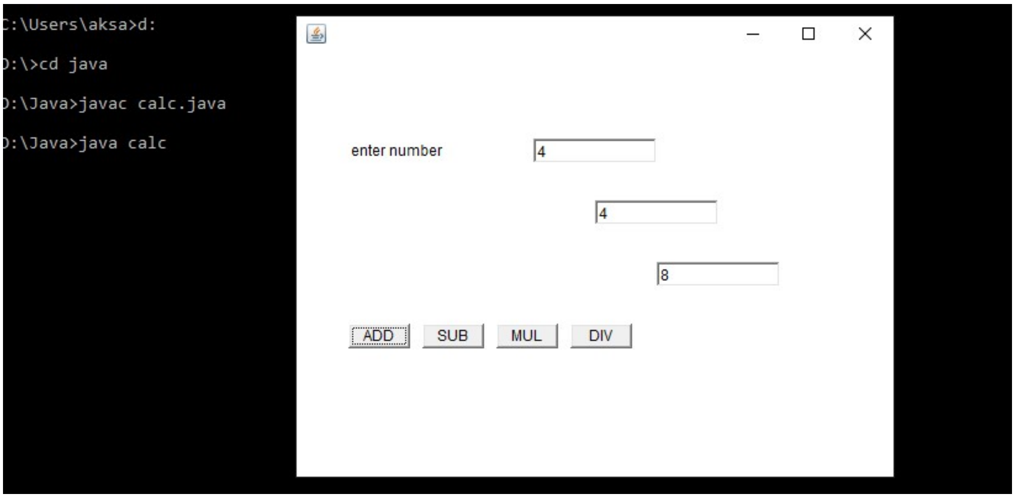
PROGRAM

```
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, 100, 100, 20);
 l3.setBounds(50, 100, 100, 20);
 t1.setBounds(200, 100, 100, 20);
t2.setBounds(250, 150, 100, 20);
t3.setBounds(300, 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();
```

```
 }
}
<html>
<head>
</head>
<body>
<div align="center">
<applet code="calc.class"width="800"height="500">
</applet>
</div>
</body>
</html>
```

Output



```
C:\Users\aksa>d:

D:\>cd java

D:\Java>javac calc.java

D:\Java>java calc
```

enter number 4

4

8

Program no: 27

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.

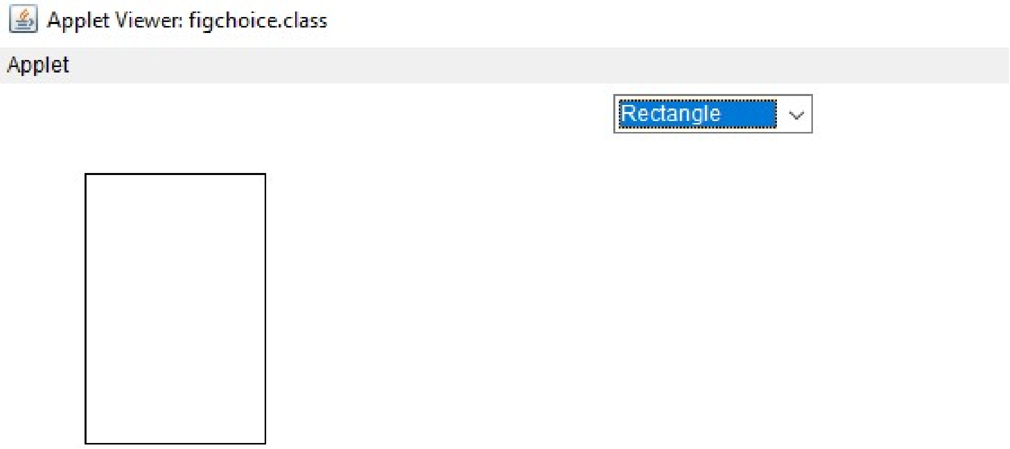
PROGRAM

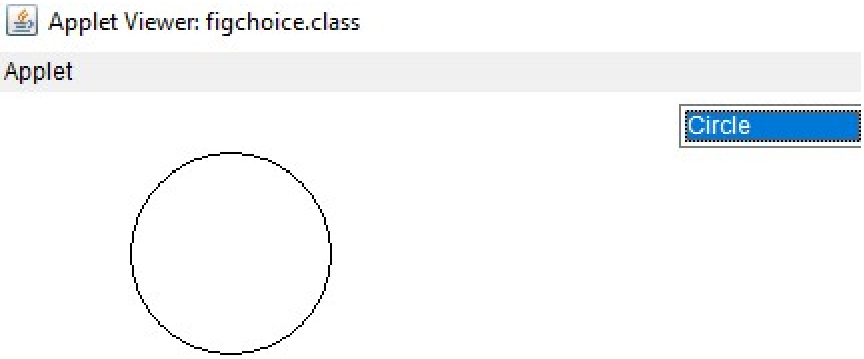
```
import java.applet.Applet;

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);

}

<html>

<head>

</head>

<body>

<div align="center">

<applet code="figchoice.class"width="800"height="500">

</applet>

</div>

</body>

</html>
```

Output




Applet Viewer: figchoice.class
Applet
Circle

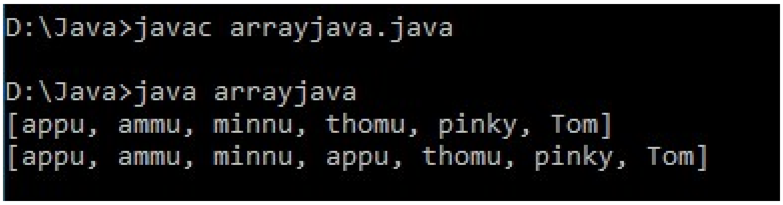
Program no:28

AIM: Maintain a list of Strings using ArrayList from collection framework, perform built-inimport java.util.

PROGRAM

```
import java.util.*;

 public class arrayjava

{

 public static void main(String args[])

 {

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

 list.add("appu");

 list.add("ammu");

 list.add("minnu");

 list.add("thomu");

 list.add("pinky");

 list.add("Tom");

 //displaying elements

 System.out.println(list);

 //Adding "appu" at the fourth position

 list.add(3, "appu");

 //displaying elements

 System.out.println(list);

 }

}
```

Output



```
D:\Java>javac arrayjava.java

D:\Java>java arrayjava
[appu, ammu, minnu, thomu, pinky, Tom]
[appu, ammu, minnu, appu, thomu, pinky, Tom]
```

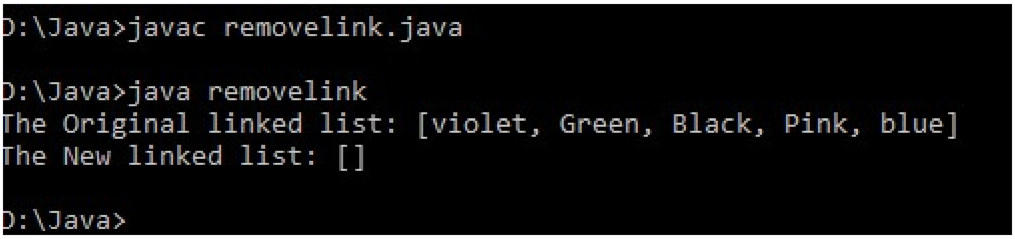
Program no:29

AIM: Program to remove all the elements from a linked list

PROGRAM

```
import java.util.*;
 public class removelink {
 public static void main(String[] args) {
 // create an empty linked list
 LinkedList<String> l_list = new LinkedList<String>();
 // use add() method to add values in the linked list
 l_list.add("violet");
 l_list.add("Green");
 l_list.add("Black");
 l_list.add("Pink");
 l_list.add("blue");

 // print the list
 System.out.println("The Original linked list: " + l_list);
 // Removing all the elements from the linked list
 l_list.clear();
 System.out.println("The New linked list: " + l_list);
}
}
```

Output



```
D:\Java>javac removelink.java

D:\Java>java removelink
The Original linked list: [violet, Green, Black, Pink, blue]
The New linked list: []

D:\Java>
```

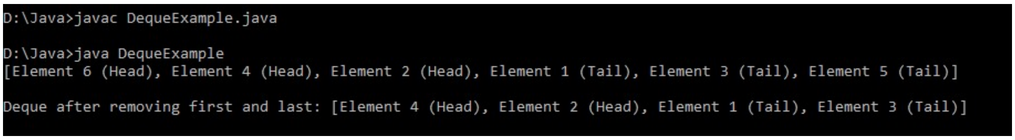
Program no:30

AIM: program to demonstrate the addition and deletion of elements in dequeue

PROGRAM

```
import java.util.*;

public class DequeExample {

        public static void main(String[] args)

        {

                Deque<String> deque

                        = new LinkedList<String>();

                deque.add("Element 1 (Tail)");deque.addFirst("Element 2 (Head)");

                deque.addLast("Element 3 (Tail)");

                deque.push("Element 4 (Head)");

                deque.offer("Element 5 (Tail)");

                deque.offerFirst("Element 6 (Head)")

                System.out.println(deque + "\n");

                deque.removeFirst();

                deque.removeLast();

                System.out.println("Deque after removing "

                                                + "first and last: "

                                                + deque);

        }

}
```

Output



```
D:\Java>javac DequeExample.java

D:\Java>java DequeExample
[Element 6 (Head), Element 4 (Head), Element 2 (Head), Element 1 (Tail), Element 3 (Tail), Element 5 (Tail)]

Deque after removing first and last: [Element 4 (Head), Element 2 (Head), Element 1 (Tail), Element 3 (Tail)]
```

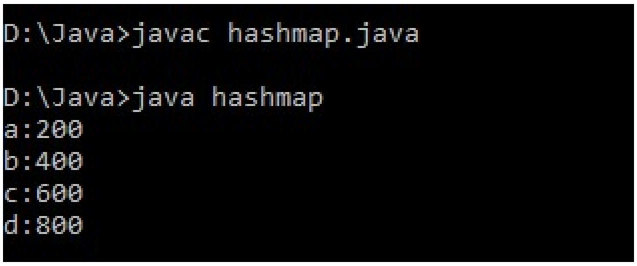
Program no:31

AIM: program to demonstrate the working of map interface byadding,removing,changing

PROGRAM

```
import java.util.*;

class hashmap

{

 public static void main(String args[])

 {

 Map<String, Integer> hm

 = new HashMap<String, Integer>();


 hm.put("a", new Integer(200));

 hm.put("b", new Integer(400));

hm.put("c", new Integer(600));

 hm.put("d", new Integer(800));


 // Traversing through the map

 for (Map.Entry<String, Integer> me : hm.entrySet())

 {

 System.out.print(me.getKey() + ":");

 System.out.println(me.getValue());

 }

 }

}
```

Output



```
D:\Java>javac hashmap.java

D:\Java>java hashmap
a:200
b:400
c:600
d:800
```

```
}
```

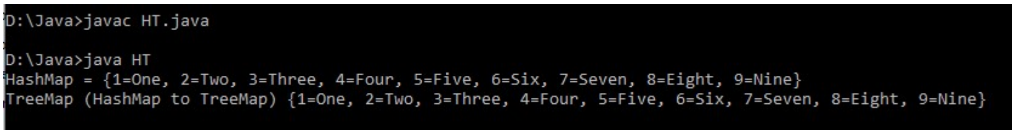
Program no:32

AIM: program to convert hash map to tree map

PROGRAM

```
import java.util.*;
import java.util.stream.*;
public class HT {
 public static void main(String args[])
 {
 Map<String, String> map = new HashMap<>();
 map.put("1", "One");
 map.put("2", "Two");
 map.put("3", "Three");
 map.put("4", "Four");
 map.put("5", "Five");
 map.put("6", "Six");
 map.put("7", "Seven");
 map.put("8", "Eight");
 map.put("9", "Nine");
 System.out.println("HashMap = " + map);
 Map<String, String> treeMap = new TreeMap<>();
 treeMap.putAll(map);
 System.out.println("TreeMap (HashMap to TreeMap) " + treeMap);
 }
```

}   **output**



```
D:\Java>javac HT.java

D:\Java>java HT
HashMap = {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}
TreeMap (HashMap to TreeMap) {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}
```

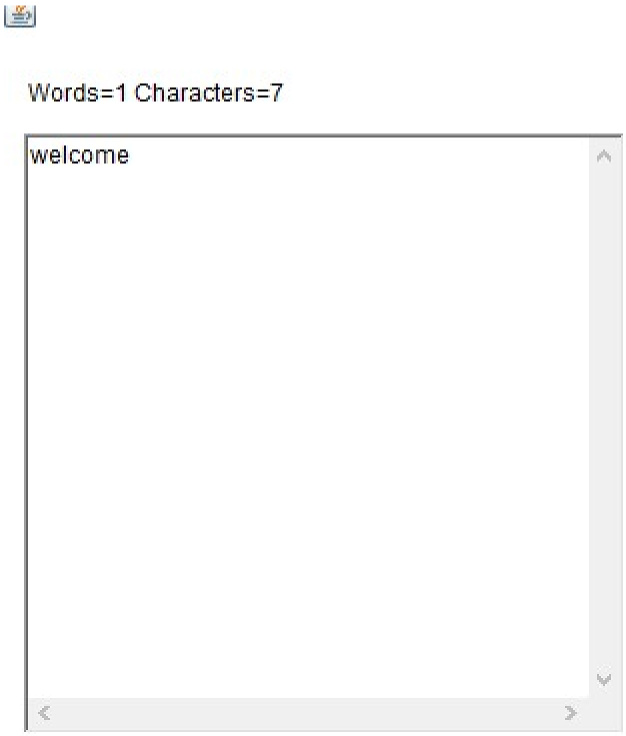
Program no:33

AIM: Develop a program to handle Key events.

PROGRAM

```
import java.awt.*;

import java.awt.event.*;

public class keyexamp extends Frame implements KeyListener

{

Label l;

TextArea a;

keyexamp()

{

l=new Label();

l.setBounds(20,50,200,20);

a=new TextArea();

a.setBounds(20,80,300,300);

a.addKeyListener(this);

add(l);

add(a);

setSize(400,400);

setLayout(null);

setVisible(true);

}

public void keyPressed(KeyEvent e)
{

}

public void keyReleased(KeyEvent e)

{

String t=a.getText();

String w[]=t.split("\\s");

l.setText("Words="+w.length+" Characters="+t.length());

}

public void keyTyped(KeyEvent e)

{}

public static void main(String args[])

{

new keyexamp();

}

}
```

Output



```
Words=1 Characters=7

welcome
```

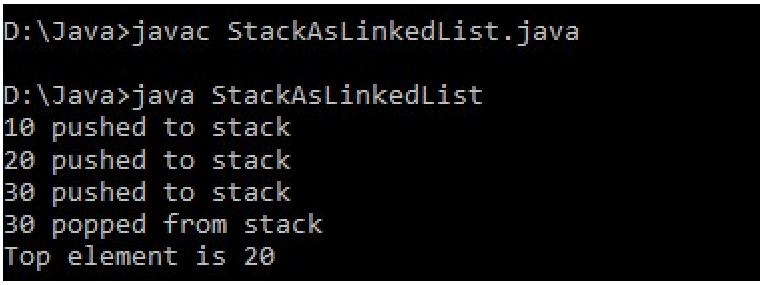
Program no:34

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

```
public class StackAsLinkedList
{

    StackNode root;

    static class StackNode {
             int data;
        StackNode next;

        StackNode(int data) { this.data = data; }
    }

    public boolean isEmpty()
    {
        if (root == null) {
            return true;
        }
    else
        return false;
}


public void push(int data)
{
    StackNode newNode = new StackNode(data);


    if (root == null)
                {
                  root = newNode;
                  }
    else {
        StackNode temp = root;
        root = newNode;
        newNode.next = temp;
    }
    System.out.println(data + " pushed to stack");
}


public int pop()
```

```
    {
      int popped = Integer.MIN_VALUE;
      if (root == null) {
        System.out.println("Stack is Empty");
      }
      else {
        popped = root.data;
        root = root.next;
      }
      return popped;
    }
         public int peek()
    {
      if (root == null) {
        System.out.println("Stack is empty");
        return Integer.MIN_VALUE;
      }
      else {
        return root.data;
```

output



```
D:\Java>javac StackAsLinkedList.java

D:\Java>java StackAsLinkedList
10 pushed to stack
20 pushed to stack
30 pushed to stack
30 popped from stack
Top element is 20
```

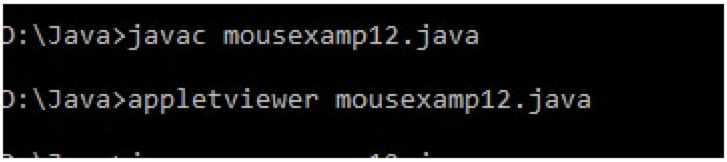
Program no:35

AIM: Develop a program to handle all mouse events

PROGRAM

```
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>javac mousexamp12.java

D:\Java>appletviewer mousexamp12.java
```

Program no:36

AIM:Develop a program to handle all window events

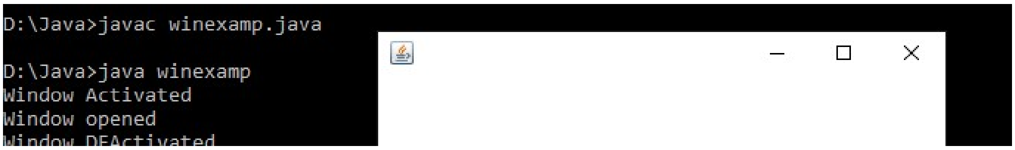
PROGRAM

```
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



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

D:\Java>java winexamp
Window Activated
Window opened
Window DEActivated
```

Program no:37

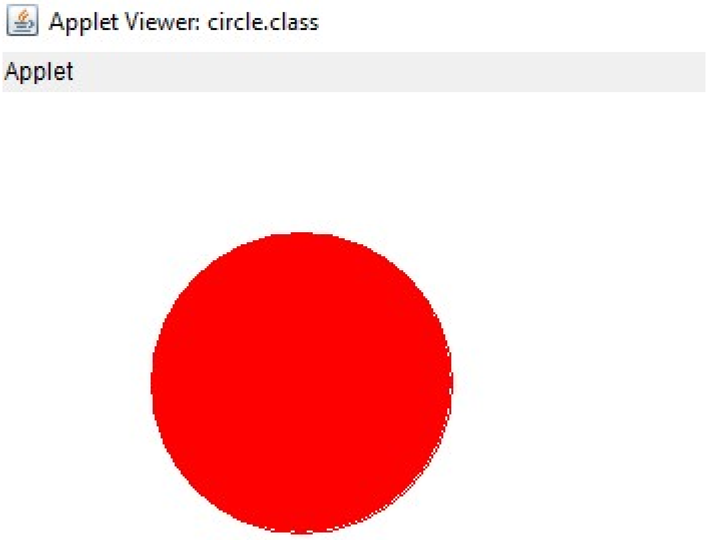
AIM:Program to draw Circle, Rectangle, Line in Applet.

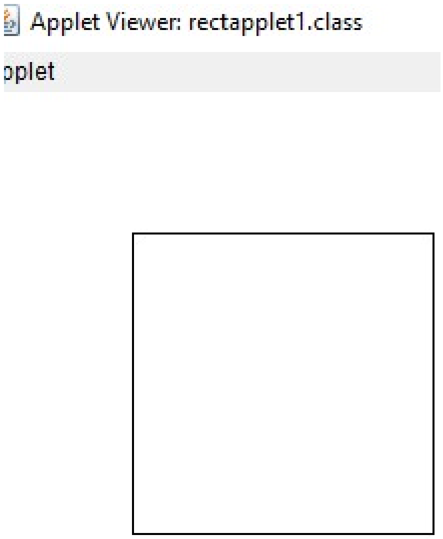
PROGRAM

```
import java.awt.*;

import java.applet.*;

public class circle extends Applet

{

public void paint(Graphics g)

{   g.setColor(Color.red);

   g.fillOval(80,70,150,150);

            g.drawOval(80,70,150,150);

            g.setColor(Color.BLACK);

}

}

<html>

<head>

</head>

<body>

<div align="center">

<applet code="circle.class"width="800"height="500">

*


</applet>

</div>

</body>
```

```
</html>
```

Output


Applet Viewer: circle.class
Applet


Applet Viewer: rectapplet1.class
pplet

Program no:38

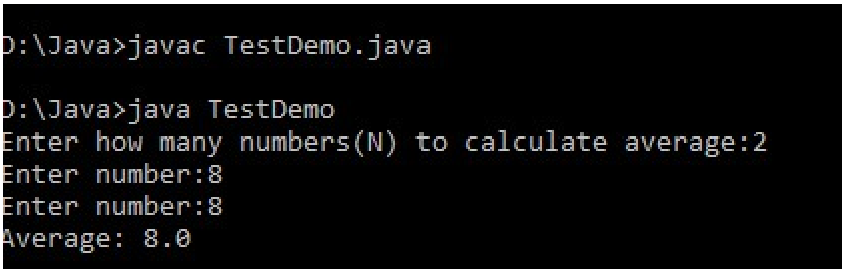
AIM:Find the average of N positive integers, raising a user defined exception for each negative input.

PROGRAM

```
import java.util.Scanner;

import java.util.InputMismatchException;

public class TestDemo

{

            public static void main(String args[])

            {

                        double total = 0, N, userInput;

                        Scanner input = new Scanner(System.in);

                        while (true)

                        {

                         System.out.print("Enter how many numbers(N) to calculate

                        average:");

                         userInput = input.nextDouble();

                          if (userInput > 0)

                          {

                                    N = userInput;

                                    break;

                                      }

                          else

                                              System.out.println("N must be positive.");
```

```
        }
        for (int i = 0; i < N; i++)
        {
            while (true)
            {
                System.out.print("Enter number:");
                try
                {
                    userInput = input.nextDouble();
                    total += userInput;
                    break;

                }
                catch (InputMismatchException e)
                {
                    input.nextLine();
                    System.out.println("Input must be a
                   number. Try again");
                }
            }
            }
        System.out.println("Average: "+ total / N);
    }
}
```

Output



```
D:\Java>javac TestDemo.java

D:\Java>java TestDemo
Enter how many numbers(N) to calculate average:2
Enter number:8
Enter number:8
Average: 8.0
```

Program no:39

AIM: Create an Arithmetic package that has classes and interfaces for the 4 basicarithmetic operations. Test the package by implementing all operations ontwo given numbers

PROGRAM

```
package Aarithmetic;



interface operations

{

public void input();

 public void add();

 public void

substract();

public void multiply();

public void division();

}

package Aarithmetic;

import java.util.*;

public class basic implements operations

{

double a,b,ad,dif,mult,div; public void input()

{
```

```
Scanner ab=new Scanner(System.in); System.out.println("Enter two
numbers"); a=ab.nextInt();

b=ab.nextInt();

}

public void add()

{

ad=a+b;

System.out.println("Sum is "+ad);

}

public void substract()

{



dif=a-b;

System.out.println("Difference is "+dif);

}

public void multiply()

{

mult=a*b;

System.out.println("Product is "+mult);

}

public void division()
```

```
{

div=a/b;

System.out.println("Quotient is "+div);

}

public static void main(String[] args)

{

basic o=new basic(); o.input();

o.add(); o.substract(); o.multiply();

o.division();

}

}
```

Output

```
 Javac –d . operations.java

Java –d . basic.java

Java Aarithmetic.basic

Enter two numbers

5

2

Sum is 7.0

Difference is 3.0

Product is 10.0

Quotient is 2.5
```

Program no:40

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

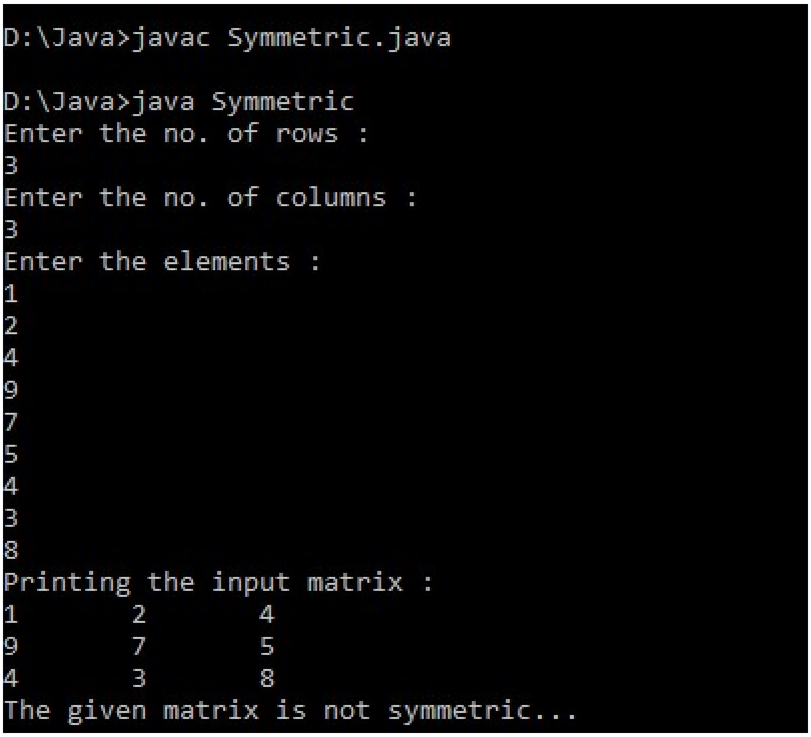
PROGRAM

```
import java.util.Scanner;

public class Symmetric

{

   public static void main(String[] args)

   {

            Scanner sc = new Scanner(System.in);

      System.out.println("Enter the no. of rows : ");

      int rows = sc.nextInt();

      System.out.println("Enter the no. of columns : ");

      int cols = sc.nextInt();

      int matrix[][] = new int[rows][cols];

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

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

      {

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

         {

            matrix[i][j] = sc.nextInt();

         }

      }

      System.out.println("Printing the input matrix :");

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

      {
```

```
      for (int j = 0; j < cols; j++)

      {

        System.out.print(matrix[i][j]+"\t");

                                }

                                System.out.println();

    }

    if(rows != cols)

    {

      System.out.println("The given matrix is not a square matrix, so it can't be
symmetric.");

    }

    else

    {

      boolean symmetric = true;

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

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

          if(matrix[i][j] != matrix[j][i])   {

            symmetric = false;

            break;      }

        }   }

      if(symmetric)

                         {

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

      }

      else

                         {
```

```
            System.out.println("The given matrix is not symmetric...");
         }   }   sc.close();
   }   }
```

Output



```
D:\Java>javac Symmetric.java

D:\Java>java Symmetric
Enter the no. of rows :
3
Enter the no. of columns :
3
Enter the elements :
1
2
4
9
7
5
4
3
8
Printing the input matrix :
1       2       4
9       7       5
4       3       8
The given matrix is not symmetric...
```