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.

Aim:

To define a class 'product' with data members, pcode, pname and price and create 3 objects of the class and find the product having the lowest price.

Source code:

Product.java

```
import java.io.*;
import java.lang.*;
public class Product
{
int pcode;
String pname;
int price;
BufferedReaderbr=newBufferedReader(new InputStreamReader(System.in);
public void getdata()
{
try
{
System.out.println("enter pcode, price,pname");
pcode=Integer.parseInt(br.readLine());
price=Integer.parseInt(br.readLine());
pname=br.readLine();
}
catch(IOException e)
{
System.out.println(e);
}
```

```
}
 public void show()
{
System.out.println("pcode:"+pcode);
System.out.println("pname:"+pname);
System.out.println("price:"+price);
}
static void compare(Product p1,Product p2,Product p3)
{
System.out.println(" The product with lowest price is:");
if(p1.price<p2.price && p1.price < p3.price)
{
System.out.println("pcode:"+p1.pcode);
System.out.println("pname:"+p1.pname);
System.out.println("price:"+p1.price);
}
else if(p2.price<p1.price && p2.price < p3.price)
{
System.out.println("pcode:"+p2.pcode);
System.out.println("pname:"+p2.pname);
System.out.println("price:"+p2.price);
}
else
{
System.out.println("pcode:"+p3.pcode);
System.out.println("pname:"+p3.pname);
System.out.println("price:"+p3.price);
```

```
}
}
public static void main(String[] args) throws IOException
{
Product p1,p2,p3;
p1= new Product();
p2= new Product();
p3= new Product();
p1.getdata();
p2.getdata();
p3.getdata();
p1.show();
p2.show();
p3.show();
compare(p1,p2,p3);
}
}
```

OUTPUT

```
enter pcode, price,pname
1001
200
powder
enter pcode, price,pname
1008
50
cream
enter pcode, price,pname
1007
100
box
pcode:1001
pname:powder
price:200
pcode:1008
pname:cream
price:50
pcode:1007
pname:box
price:100
 The product with lowest price is:
pcode:1008
pname:cream
price:50
```

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

Aim:

To read 2 matrices from the console and perform matrix addition.

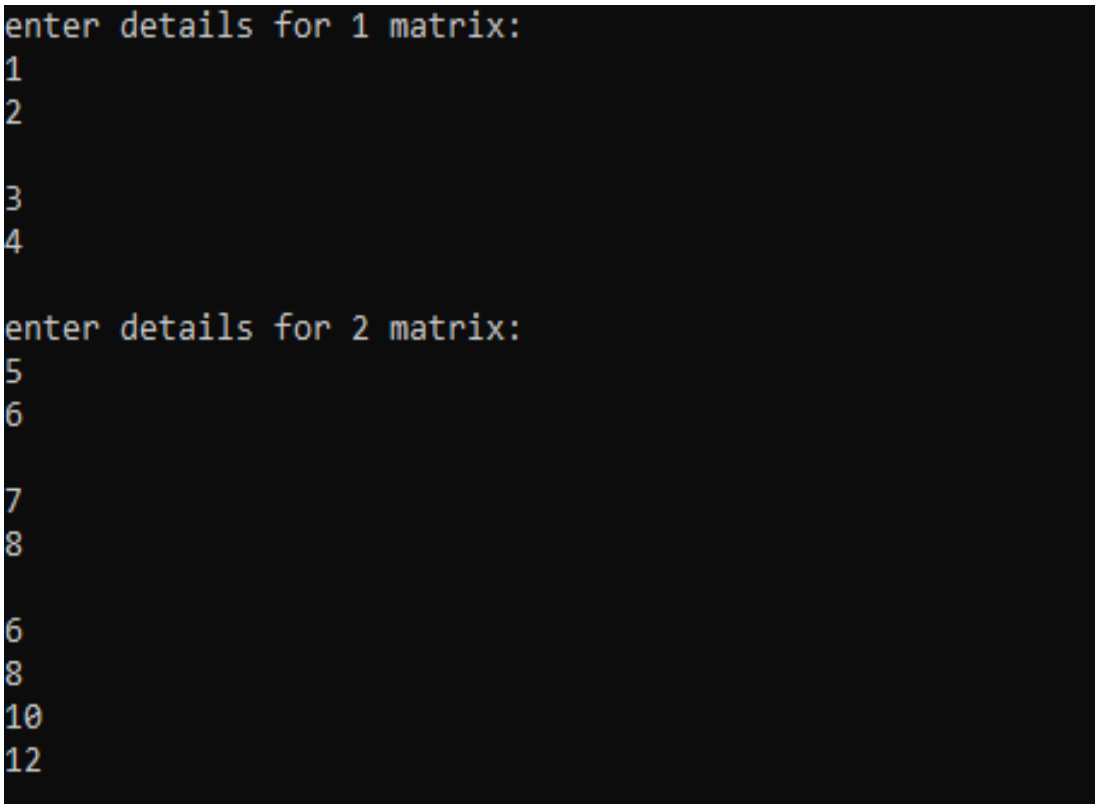
Source code:

matrix1.java

```
import java.io.*;
public class matrix1
{
public static void main(String[] args)
{
int a[][]=new int[2][2];
int b[][]=new int[2][2];
int c[][]=new int[2][2];
BufferedReaderbr=newBufferedReader(newInputStreamReader(System.in))
;
try
{
System.out.println("enter details for 1 matrix:");
for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
a[i][j]=Integer.parseInt(br.readLine());
}
System.out.println();
}
System.out.println("enter details for 2 matrix:");
for(int i=0;i<2;i++)
{
```

```
   for(int j=0;j<2;j++)
{
b[i][j]=Integer.parseInt(br.readLine());
}
System.out.println();
}
}
catch(IOException e)
{
System.out.println(e);
}
for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
c[i][j]=a[i][j]+b[i][j];
}
}
for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
System.out.println(c[i][j]);
}}
}
}
```

OUTPUT


enter details for 1 matrix:
1
2
3
4
enter details for 2 matrix:
5
6
7
8
6
8
10
12

3) Add complex numbers

Aim:

To add complex numbers

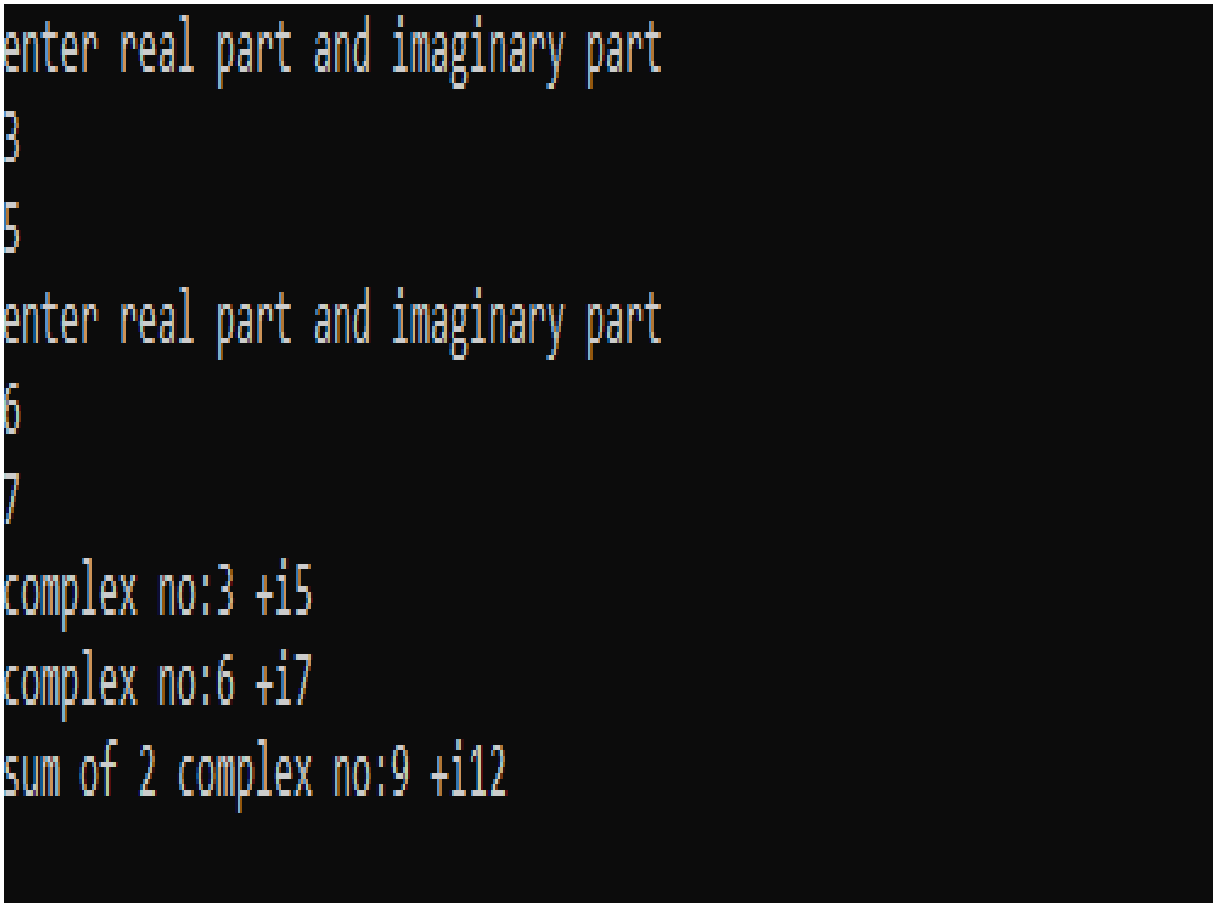
Source code:

complex.java

```
import java.io.*;
public class complex
{
int real;
int img;
BufferedReader br=newBufferedReader(new InputStreamReader(System.in));
public void getdata()
{
try
{
System.out.println("enter real part and imaginary part");
real=Integer.parseInt(br.readLine());
img=Integer.parseInt(br.readLine());
}
catch(IOException e)
{
System.out.println(e);
}
}
 public void show()
{
System.out.println("complex no:"+ real +" +i"+ img);
}
public static void main(String[] args)
{
```

```
complex c1,c2;
c1 = new complex();
c2 = new complex();
c1.getdata();
c2.getdata();
c1.show();
c2.show();
int x= c1.real + c2.real;
int y = c1.img + c2.img;
System.out.println("sum of 2 complex no:"+ x +" +i"+ y);
}
}
```

OUTPUT


enter real part and imaginary part
3
5
enter real part and imaginary part
6
7
complex no:3 +i5
complex no:6 +i7
sum of 2 complex no:9 +i12

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

Aim:

To read a matrix from console and check whether it is symmetric or not.

Source code:

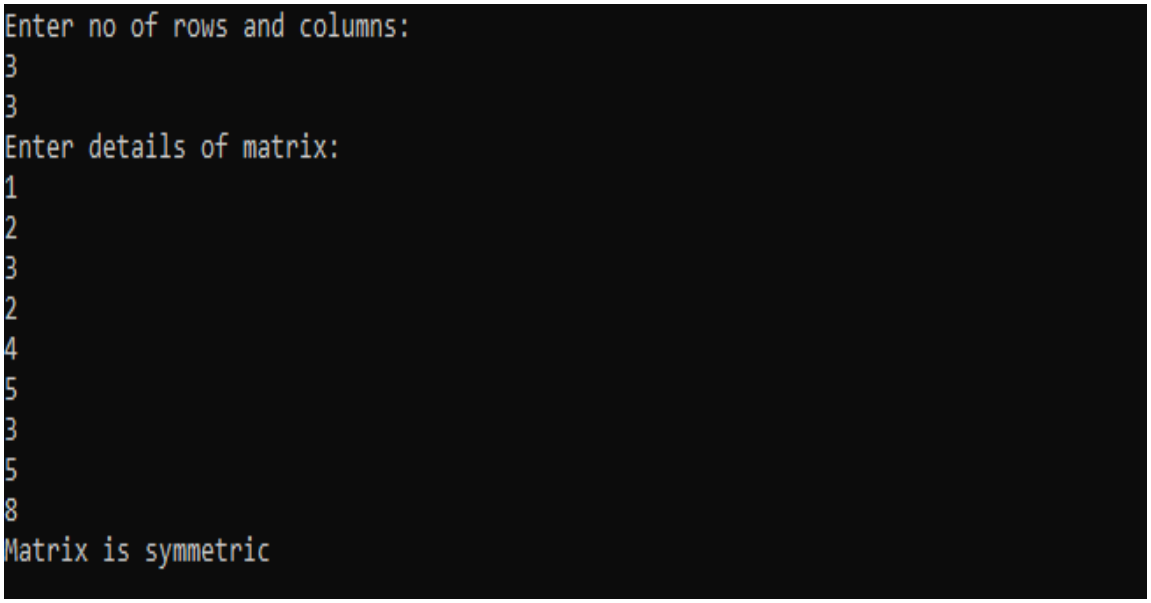
matrix2.java

```
import java.io.*;
 public class matrix2
{
public static void main(String[] args)
{
int a[][]= new int[3][3];
int b[][]= new int[3][3];
int r=0,c=0,flag=0;
BufferedReader br =  new BufferedReader(newInputStreamReader(System.in));
System.out.println("Enter no of rows and columns:");
try
{
r=Integer.parseInt(br.readLine());
c=Integer.parseInt(br.readLine());
System.out.println("Enter details of matrix:");
for(int i=0;i<r;i++)
{
for(int j=0;j<c;j++)
{
a[i][j]=Integer.parseInt(br.readLine());
}
   }
}
catch(IOException e)
```

```
{
 System.out.println(e);
}
for(int i=0;i<r;i++)
{
for(int j=0;j<c;j++)
{
b[j][i]=a[i][j];
}
}
if(r==c)
{
for(int i=0;i<r;i++)
{
for(int j=0;j<c;j++)
{
if(b[i][j] != a[i][j])
{
flag=1;
break;
}
}
}
if(flag==1)
System.out.println("Matrix is not symmetric");
else
System.out.println("Matrix is symmetric");
```

```
}
else
System.out.println("Since matrix is not square matrix, it is not symmetric");
}
}
```

OUTPUT


Enter no of rows and columns:
3
3
Enter details of matrix:
1
2
3
2
4
5
3
5
8
Matrix is symmetric

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.

Aim:

To create CPU with attribute price and to create inner class processor and static nested class RAM. Then create an object of CPU and print information of processor and RAM.

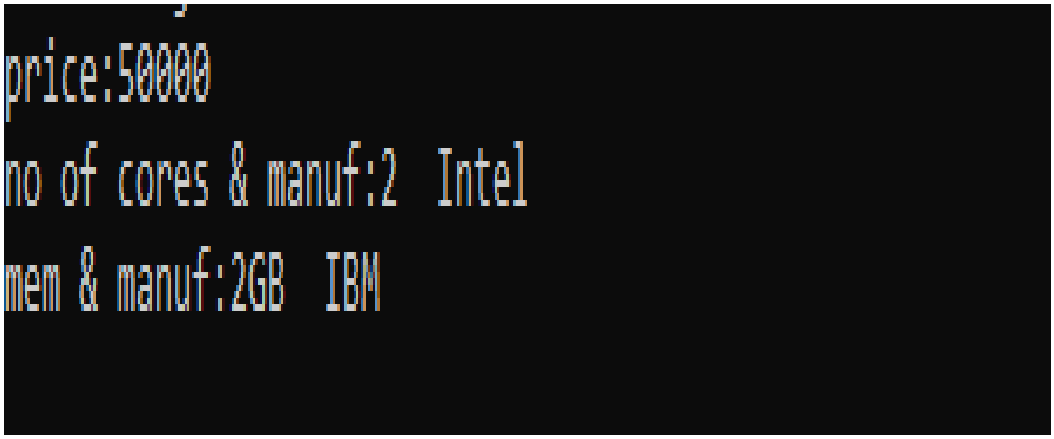
Source code:

CPU.java

```
import java.io.*;
class CPU
{
int price=50000;
 class processor
    {
    int noofcores=2;
    String manuf="Intel";
    }
    static class RAM
    {
    String mem="2GB";
    String manuf="IBM";
    }
    public static void main(String []args) throws IOException
    {
    CPU c= new CPU();
CPU.processor p= c.new processor();
CPU.RAM r= new CPU.RAM();
System.out.println("price:"+c.price);
```

```
System.out.println("no of cores & manuf:"+p.noofcores +"  " +p.manuf);
System.out.println("mem & manuf:"+r.mem+"  "+r.manuf);
}
}
```

OUTPUT



6) Program to Sort strings

Aim:

To write a program to sort string.

Source code:

arrstring.java

```
import java.io.*;
public class arrstring
{
public static void main(String[] args) throws IOException
{
String[] arr= {"Java","CPP","Visual Basic", "Python"};
int size=arr.length;
for(int i=0;i<size-1;i++)
{
for(int j=i+1;j<size;j++)
{
if(arr[i].compareTo(arr[j])>0))
{
String temp=arr[i];
arr[i]=arr[j];
arr[j]=temp;
}}}
for(int i=0;i<size;i++)
{
System.out.println(arr[i]);
}
}
}
```

OUTPUT

7) Search an element in an array.

Aim:

To write a program to search an element in an array.

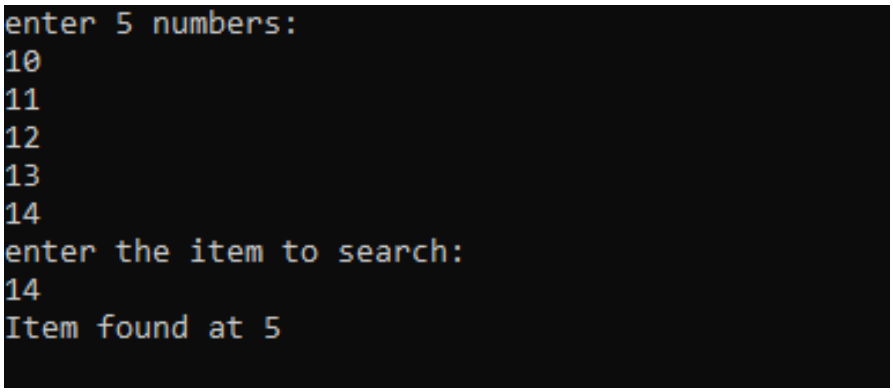
Source code:

array.java

```
import java.io.*;
class array
{
public static void main(String []args)
{
int []arr= new int[5];
int item=0,flag=0,i=0;
BufferedReader br=newBufferedReader(new InputStreamReader(System.in));
try
{
System.out.println("enter 5 numbers:");
for(i=0;i<5;i++)
{
arr[i]=Integer.parseInt(br.readLine());
}
System.out.println("enter the item to search:");
item=Integer.parseInt(br.readLine());
}
catch(IOException e)
{
System.out.println(e);
}
for(i=0;i<5;i++)
{
if(item==arr[i])
```

```
{
flag=1;
break;
}
}
if(flag==1)
System.out.println("Item found at "+ (i+1));
else
System.out.println("Item not found ");
}
}
```

OUTPUT


enter 5 numbers:
10
11
12
13
14
enter the item to search:
14
Item found at 5

8) Perform string manipulations.

Aim:

To perform string manipulation.

Source code:

strmanip.java

```
import java.io.*;
class strmanip
{
public static void main(String []args)
{
String s1,s2,s3;
int choice;
BufferedReader br =newBufferedReader(new InputStreamReader(System.in));
try
{
System.out.println("enter 2 strings:");
s1=br.readLine();
s2=br.readLine();
do
{
System.out.println("enterur(1copy,2append,3compare,4reverse,5exit)choice:);
choice=Integer.parseInt(br.readLine());
switch(choice)
{
case 1: s3=s1;
System.out.println("copied string:"+s3);
break;
case 2: s3=s1+ "  "+s2;
System.out.println("appended strings:"+s3);
break;
```

```
case 3: int x=s1.compareTo(s2);
if(x==0)
System.out.println("strings are equal");
else if(x<0)
System.out.println("II string is larger");
else
System.out.println("I string is larger");
break;
case 4: s3=new StringBuffer(s1).reverse().toString();
System.out.println("reversed string:"+s3);
break;
}
}while(choice<5);
}
catch(IOException e)
{
System.out.println(e);
}
}
}
```

OUTPUT

```
enter 2 strings:
Anila
Lawrence
enter ur(1-copy,2-append, 3-compare,4-reverse,5-exit) choice:
2
appended strings:Anila  Lawrence
enter ur(1-copy,2-append, 3-compare,4-reverse,5-exit) choice:
1
copied string:Anila
enter ur(1-copy,2-append, 3-compare,4-reverse,5-exit) choice:
3
II string is larger
enter ur(1-copy,2-append, 3-compare,4-reverse,5-exit) choice:
4
reversed string:alinA
enter ur(1-copy,2-append, 3-compare,4-reverse,5-exit) choice:
5
```

9) 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.

Aim:

To write a program to create a class employee with attributes eNo, eName, eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

Source code:

Employee.java

```
import java.io.*;
class emp
{
int empno,salary;
String ename;
void getdata(int empno, String ename, int salary)
{
this.empno=empno;
this.ename=ename;
this.salary=salary;
}
void display()
{
System.out.println("employee details:");
System.out.println("emp no:"+ empno);
System.out.println("emp name:"+ ename);
System.out.println("salary:"+ salary);
}
}
class Employee
{
public static void main(String[] args)
```

```
{
emp []e= new emp[5];
BufferedReader br=newBufferedReader(new InputStreamReader(System.in));
int eno1=0,eno2=0,sal=0,i,n=0;
String name;
try
{
System.out.println("enter no of employees:");
n=Integer.parseInt(br.readLine());
for(i=0;i<n;i++)
{
System.out.println("enter the empno,ename,salary:");
eno1=Integer.parseInt(br.readLine());
name=br.readLine();
sal=Integer.parseInt(br.readLine());
e[i]=new emp();
e[i].getdata(eno1,name,sal);
}
System.out.println("enter the emp no to search:");
eno2=Integer.parseInt(br.readLine());
for(i=0;i<n;i++)
{
if(e[i].empno==eno2)
{
e[i].display();
break;
}
}
}
```

```
catch(IOException e1)
{
System.out.println(e1);}
}
}
```

OUTPUT

```
enter no of employees:
4
enter the empno,ename,salary:
1
Anila
4000
enter the empno,ename,salary:
2
Ajay
5000
enter the empno,ename,salary:
3
Adarsh
6000
enter the empno,ename,salary:
4
Arya
7000
enter the emp no to search:
1
employee details:
emp no:1
emp name:Anila
salary:4000
```

10) Area of different shapes using overload functions.

Aim:

To find area of different shapes using overload function.

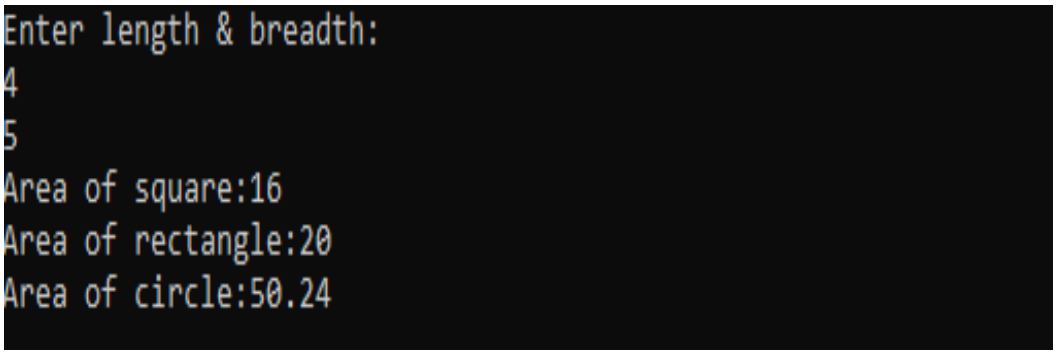
Source code:

over.java

```
import java.io.*;
 class over
{
void area(int l)
{
System.out.println("Area of square:"+l * l);
}
void area(int l, int b)
{
System.out.println("Area of rectangle:"+l * b);
}
void area(float l)
{
System.out.println("Area of circle:"+3.14 * l *l);
}
public static void main(String []args)
{
int l=0,b=0;
over o= new over();
BufferedReaderbr=newBufferedReader(new InputStreamReader(System.in));
try
{
System.out.println("Enter length & breadth:");
l= Integer.parseInt(br.readLine());
b= Integer.parseInt(br.readLine());
```

```
}
catch(IOException e)
{
System.out.println(e);
}
o.area(l);
o.area(l,b);
float x= (float)l;
o.area(x);
}
}
```

OUTPUT



11) 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.

Aim:

To create a class Employee with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create class 'Teacher' that inherit the properties of class employee and contain its own data members include display function to display all the data members. Use array of objects to display details of N teachers.

Source code:

inher.java

```
import java.io.*;
class emp
{
int empid,salary;
String ename,address;
emp(int empid, String ename, int salary,String address)
{
this.empid=empid;
this.ename=ename;
this.salary=salary;
this.address=address;
}
}
class teacher extends emp
{
String dept,subj;
teacher(int empid, String ename, int salary,Stringaddress,Stringdept,Stringsubj)
{
```

```
super(empid,ename,salary, address);
this.dept=dept;
this.subj=subj;
}
void display()
{
System.out.println("employee details:");
System.out.println("emp no:"+ empid);
System.out.println("emp name:"+ ename);
System.out.println("salary:"+ salary);
System.out.println("address:"+ address);
System.out.println("department:"+ dept);
System.out.println("subject:"+ subj);
}
}
class inher
{
public static void main(String[] args)
{
teacher []t= new teacher[5];
BufferedReaderbr=newBufferedReader(newInputStreamReader(System.i);
int eno1=0,sal=0,i,n=0;
String name,addr,dept,subj;
try
{
System.out.println("enter no ofemployees:");n=Integer.parseInt(br.readLin);
for(i=0;i<n;i++)
{
System.out.println("enter the empno,ename,salary,address,dept,subj:");
```

```
eno1=Integer.parseInt(br.readLine());
name=br.readLine();
sal=Integer.parseInt(br.readLine());
addr=br.readLine();
dept=br.readLine();
subj=br.readLine();
t[i]=new teacher(eno1,name,sal,addr,dept,subj);
}
}
catch(IOException e1)
{System.out.println(e1);}
for(i=0;i<n;i++)
{
t[i].display();
}
}
}
```

OUTPUT

```
enter no of employees:
2
enter the empno,ename,salary,address,dept,subj:
1
Anila
25000
Little Heaven,Trivandrum
Sales
Java
enter the empno,ename,salary,address,dept,subj:
2
Arya
45000
Arya villa,Trivandrum
Manager
Computer
employee details:
emp no:1
emp name:Anila
salary:25000
address:Little Heaven,Trivandrum
department:Sales
subject:Java
employee details:
emp no:2
emp name:Arya
salary:45000
address:Arya villa,Trivandrum
department:Manager
subject:Computer
```

12) 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.

Aim:

To 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. Create another class 'Teacher' that inherits the properties of class Employee and also contain constructors and methods to display the data members and display details of N teachers.

Source code:

TestInheritance1.java

```
import java.util.Scanner;
class person {
String memb_name;
String gender;
String address;
int age;
Scanner ps =new Scanner(System.in);
public person() {
System.out.println("enter member name:");
memb_name=ps.next();
System.out.println("enter gender:");
gender=ps.next();
System.out.println("enter address:");
address=ps.next();
System.out.println("enter age:");
```

```
age=ps.nextInt();
}
}
class Employee extends person {
int emp_id;
String company_name;
String qualification;
double salary;
Scanner em =new Scanner(System.in);
Employee() {
System.out.println("enter emp_id:");
emp_id=em.nextInt();
System.out.println("enter company_name:");
company_name=em.next();
System.out.println("enter qualification:");
qualification=em.next();
System.out.println("enter salary:");
salary=em.nextDouble();
}
}
class Teacher extends Employee {
String subject;
String department;
int Teacher_id;
Scanner th=new Scanner(System.in);
Teacher() {
System.out .println("enter subject:");
subject=th.next();
System.out.println("enter department:");
```

```
department=th.next();
System.out.println("enter Teacher_id:");
Teacher_id=th.nextInt();
}
public void display() {
System.out.println("----------****DETAILS*****------------");
System.out.println("Name: "+memb_name);
System.out.println("Gender: "+gender);
System.out.println("Address: "+address);
System.out.println("Age: "+age);
System.out.println("Employee id: "+emp_id);
System.out.println("Company name: "+company_name);
System.out.println("Qualification: "+qualification);
System.out.println("Salary: "+salary);
System.out.println("Subject: "+subject);
System.out.println("Department details: "+department);
System.out.println("Teacher id: "+Teacher_id);
}
}
public class TestInheritance1{
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
int i;
System.out.println("Enter number of persons: ");
int n=sc.nextInt();
Teacher k[]=new Teacher[n];
for(i=0;i<n;i++)
{
k[i]=new Teacher();
```

```
}
for(i=0;i<n;i++)
{
k[i].display();
}
}
}
```

OUTPUT

```
Enter number of persons:
2
enter member name:
anila
enter gender:
female
enter address:
uygiuyiufhryferjgth
enter age:
22
enter emp_id:
234
enter company_name:
tcs
enter qualification:
mca
enter salary:
30000
enter subject:
computer
enter department:
computer
enter Teacher_id:
245
enter member name:
ajay
enter gender:
male
enter address:
iueoriurihfhuhi
enter age:
22
enter emp_id:
235
enter company_name:
wipro
enter qualification:
mca
enter salary:
35000
enter subject:
science
enter department:
physics
```

```
enter Teacher_id:
246
----------****DETAILS*****------------
Name: anila
Gender: female
Address: uygiuyiufhryferjgth
Age: 22
Employee id: 234
Company name: tcs
Qualification: mca
Salary: 30000.0
Subject: computer
Department details: computer
Teacher id: 245
----------****DETAILS*****------------
Name: ajay
Gender: male
Address: iueoriurihfhuhi
Age: 22
Employee id: 235
Company name: wipro
Qualification: mca
Salary: 35000.0
Subject: science
Department details: physics
Teacher id: 246
```

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

Aim:

To write a program with class publisher, Book, Literature and Fiction and read information and print details of book from any of the categories.

Source code:

Dispbook.java

```
import java.util.*;
class Publisher
{
String name;
Scanner sc=new Scanner(System.in);
public Publisher()
{
System.out.println("ENTER THE DETAILS");
System.out.println("enter the name");
name = sc.next();
}
}
class book extends Publisher
{
int bid;
String company_name;
int price;
Scanner em= new Scanner(System.in);
public book()
{
System.out.println("enter the id");
bid = em.nextInt();
```

```
System.out.println("enter the company name");
company_name = em.next();
System.out.println("enter price");
price = em.nextInt();
}
}
class lict extends book
{
String type;
String discription;
String auother;
Scanner tr = new Scanner(System.in);
public  void get()
{
System.out.println("emter the details of literature books");
System.out.println("enter the type");
type = tr.next();
System.out.println("enter the description");
discription = tr.next();
System.out.println("enter author");
auother = tr.next();
}
public void display()
{
System.out.println("------* details*-------");
System.out.println("The name of book:"+name);
System.out.println("The book id:"+bid);
System.out.println("Published by:"+company_name);
System.out.println("Price:"+price);
```

```
System.out.println("Category:"+type);
System.out.println("About book:"+discription);
}
}
class fict extends book
{
String ftype;
String fdiscription;
String fauother;
Scanner fi = new Scanner(System.in);
public  void get()
{
System.out.println("enter the details of fiction books");
System.out.println("enter the type");
ftype = fi.next();
System.out.println("enter the description");
fdiscription = fi.next();
System.out.println("enter author");
fauother = fi.next();
}
public void display()
{
System.out.println("------* details*-------");
System.out.println("The name of book:"+name);
System.out.println("Book id :"+bid);
System.out.println("Published company:"+company_name);
System.out.println("Price:"+price);
System.out.println("Category:"+ftype);
System.out.println("About book:"+fdiscription);
```

```
System.out.println("Author name:"+fauother);
}
}
public class Dispbook
{
public static void main(String[] args)
{
 System.out.println("Enter type of book ");
System.out.println("1.literature:");
System.out.println("2.fiction:");
System.out.println("enter the choice:");
Scanner pc =new Scanner(System.in);
int choice=pc.nextInt();
switch(choice)
{
case 1: System.out.println("literature books");
lict a =new lict();
a.get();
a.display();
break;
case 2: System.out.println("fiction books");
fict b =new fict();
b.get();
b.display();
break;
default:
System.out.println("invalid choice");
break;
} } }
```

OUTPUT

```
Enter type of book
1.literature:
2.fiction:
enter the choice:
1
literature books
ENTER THE DETAILS
enter the name
RTYUI
enter the id
1002
enter the company name
JGGDGGHHUY
enter price
2345
emter the details of literature books
enter the type
LITERATURE
enter the description
GUGERYRYF YGRUEIJOR
enter author
------* details*-------
The name of book:RTYUI
The book id:1002
Published by:JGGDGGHHUY
Price:2345
Category:LITERATURE
About book:GUGERYRYF
```

```
Enter type of book
1.literature:
2.fiction:
enter the choice:
2
fiction books
ENTER THE DETAILS
enter the name
TYUIO
enter the id
1003
enter the company name
GTUOOKJK
enter price
2347
enter the details of fiction books
enter the type
FICTION
enter the description
JHFUERIFHH RFEGHR
enter author
------* details*-------
The name of book:TYUIO
Book id :1003
Published company:GTUOOKJK
Price:2347
Category:FICTION
About book:JHFUERIFHH
Author name:RFEGHR
```

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

Aim:

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

Source code:

inter1.java

```
import java.io.*;
class students
{
int rollno;
String name;
float marks;
void getdata(int rollno,String name,float marks)
{
this.rollno=rollno;
this.name=name;
this.marks=marks;
}
}
interface sports
{
final int score=10;
void show();
}
class result extends students implements sports
{
public void show()
{
```

```
System.out.println("roll no:"+ rollno);
System.out.println("name:"+ name);
System.out.println("marks:"+ marks);
System.out.println("score:"+ score);
}
}
class inter1
{
public static void main(String []args)
{
int rollno=0;
String name="";
float marks=0;
result r = new result();
BufferedReaderbr=newBufferedReader(newInputStreamReader(System.in));
try
{
System.out.println("enter rollno,name,marks:");
rollno=Integer.parseInt(br.readLine());
name=br.readLine();
marks=Float.parseFloat(br.readLine());
}
catch(IOException e)
{
System.out.println(e);
}
r.getdata(rollno,name,marks);
r.show();
} }
```

OUTPUT

```
enter rollno,name,marks:
1
Anila
50
roll no:1
name:Anila
marks:50.0
score:10
```

15) Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

Aim:

To create an interface with prototypes of functions area and perimeter and create another class which implement the above interfaces. Create a menu driven program to find area and perimeter of objects.

Source code:

inter2.java

```
import java.io.*;
interface areaperi
{
void area();
void perimeter();
}
class rectangle implements areaperi
{
int l=10,b=20;
public void area()
{
System.out.println("area of rectangle:"+ (l*b));
}
public void perimeter()
{
System.out.println("perimeter of rectangle:"+(2*(l+b)));
}
}
class circle implements areaperi
{
int r=10;
```

```
public void area()
{
System.out.println("area of circle:"+ (3.14*r*r));
}
public void perimeter()
{
System.out.println("perimeter of circle:"+ (2*3.14*r));
}
}
class inter2
{
public static void main(String[] args)
{
rectangle r = new rectangle();
circle c = new circle();
int ch=0;
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
try
{
System.out.println("enter 1 for circle, 2 for rectangle:");
ch=Integer.parseInt(br.readLine());
switch(ch)
{
case 1: c.area();
c.perimeter();
break;
case 2: r.area();
r.perimeter();
break;
```

```
}
}
catch(IOException e1)
{System.out.println(e1);}
}
    }
```

OUTPUT

```
enter 1 for circle, 2 for rectangle:
1
area of circle:314.0
perimeter of circle:62.800000000000004
```

```
enter 1 for circle, 2 for rectangle:
2
area of rectangle:200
perimeter of rectangle:60
```

16) 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

Aim:

To create a bill using calculate method with the given details

Source code:

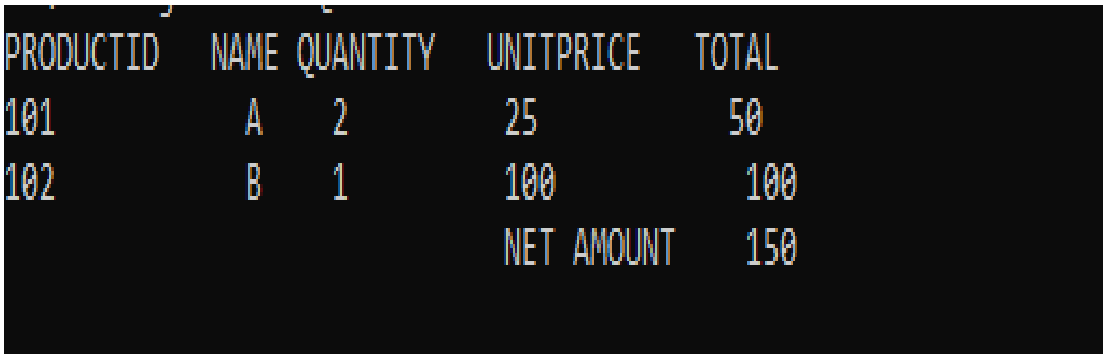
BillQ16.java

```
import java.io.*;
interface bill
{
int calculate();
}
class product implements bill
{
   int pid;
   String pname;
   int qty;
   int unitprice;
   int total;
   void getdata(int pid,String pname,int qty,int price)
{
this.pid=pid;
this.pname=pname;
this.qty=qty;
```

```
this.unitprice=price;
}
    public int calculate()
    {
        total=unitprice*qty;
        return total;
    }
    void show()
    {
     System.out.println(pid+" "+pname+" "+qty+""+unitprice+""+calculate());
    }
}
class BillQ16
{
    public static void main(String[] args)
{
  product p1=new product();
  product p2=new product();
  int pid = 0,price = 0,qty = 0;
  String pname = null;
  BufferedReader br +newBufferedReader(newInputStreamReader(System.in));
   p1.getdata(101,"A",2,25);
   p2.getdata(102,"B",1,100);
System.out.println("PRODUCTID"+"   "+"NAME"+" "+"QUANTITY"+"
"+"UNITPRICE"+"   "+"TOTAL");
                p1.show();
                p2.show();
                int t1=p1.calculate();
```

```
        int t2=p2.calculate();
        System.out.println("NET AMOUNT    "+(t1+t2));
        }
    }
```

OUTPUT



17) 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.

Aim:

To create a graphics package that has classes and interfaces for some figures and test the package by finding area of those figures.

Source code:

Package – Graphics

Dimension.java

```
package Graphics;
import java.util.Scanner;
interface calculation
{
public void rectangle();
public void triangle();
public void circle();
public void square();
}
public class Dimension implements calculation {
public void main(String args[])
{
double area;
Scanner obj1=new Scanner(System.in);
public void rectangle()
{
int l,b;
System.out.println("Enter the length of the
rectangle:");
l=obj1.nextInt();
System.out.println("Enter the Breath of the rectangle:");
b=obj1.nextInt();
```

```
area=l*b;
System.out.println("Area:"+area+"\n");
}
public void triangle() {
int h,b;
System.out.println("Enter the base of the Triangle:");
b=obj1.nextInt();
System.out.println("Enter the height of the Triangle:");
h=obj1.nextInt();
area1=(h*b)/2;
System.out.println("Area:"+area1+"\n");
}
public void circle() {
float r;
System.out.println("Enter the Radius of the Circle:");
r=obj1.nextInt();
area2=3.14*r*r;
System.out.println("Area:"+area2+"\n");
}
public void square() {
int s;
System.out.println("Enter the Side of the Square:");
s=obj1.nextInt();
area3=s*s;
System.out.println("Area:"+area3+"\n");
}
}
interface calculation{
public void rectangle();
```

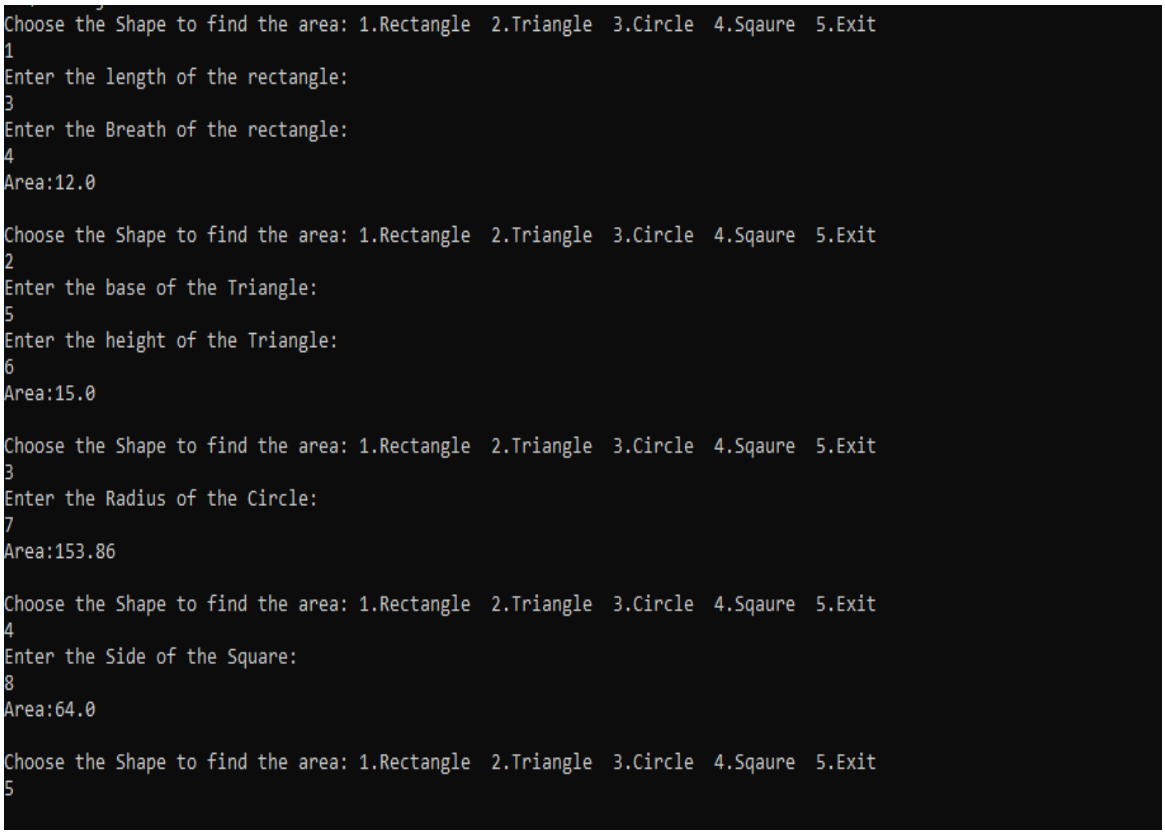
```
public void triangle();
public void circle();
public void square();
}
```

Test.java

```
import java.io.*;
import java.util.Scanner;
import Graphics.Dimension;
public class Test
{
public static void main(String[] args)
{
Scanner obk=new Scanner(System.in);
Dimension obj=new Dimension();
int ch=0;
//l=obj1.nextInt();
while(ch<5)
{
System.out.println("Choose the Shape to find the area:"+" 1.Rectangle 2.Triangle  3.Circle  4.Sqaure  5.Exit");
ch=obk.nextInt();
switch(ch)
{
case 1:
obj.rectangle();
break;
case 2:
obj.triangle();
break;
```

```
case 3:
obj.circle();
break;
case 4:
obj.square();
break;
case 5: break;
default:
System.out.println("invalid choice");
break;
}
}
}
}
```

OUTPUT



18) 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.

Aim:

To create an arithmetic package that has classes and interfaces for 4 basic arithmetic operations and test the package by implementing the operations on two numbers

Source code:

Package – Operations

arithop.java

```
package operations;
import java.util.Scanner;
public class arithop implements calculation {
double sum;
Scanner obj1=new Scanner(System.in);
public void addition() {
int l,b;
System.out.println("Enter first number");
l=obj1.nextInt();
System.out.println("Enter second number");
b=obj1.nextInt();
sum=l+b;
System.out.println("sum:"+sum+"\n");
}
public void multiplication() {
int h,b;
int mul;
System.out.println("Enter first number:");
b=obj1.nextInt();
System.out.println("Enter second number");
h=obj1.nextInt();
```

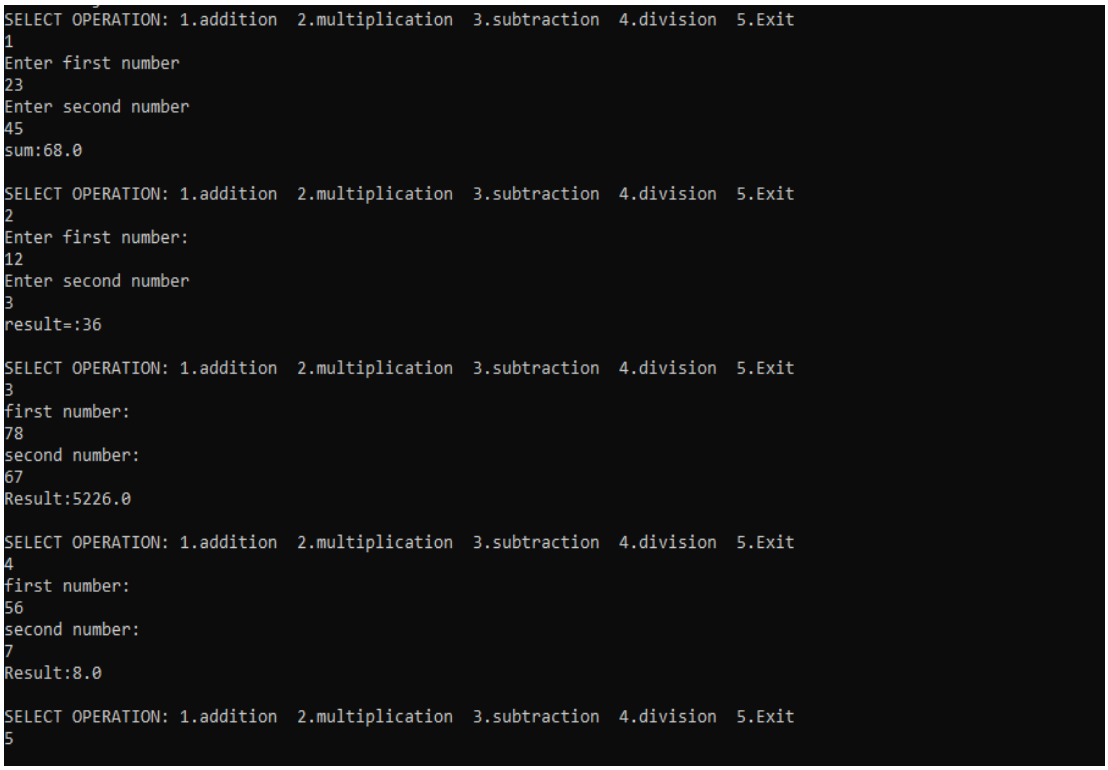
```
mul=h*b;
System.out.println("result=:"+mul+"\n");
}
public void subtraction() {
float r,z,sub;
System.out.println("first number:");
r=obj1.nextInt();
System.out.println("second number:");
z=obj1.nextInt();
sub=r*z;
System.out.println("Result:"+sub+"\n");
}
public void division() {
float r,z,div;
System.out.println("first number:");
r=obj1.nextInt();
System.out.println("second number:");
z=obj1.nextInt();
div=r/z;
System.out.println("Result:"+div+"\n");
}
}
interface calculation{
public void addition();
public void multiplication();
public void subtraction();
public void division();
}
```

Test1.java

```
import java.io.*;
import java.util.Scanner;
import operations.arithop;
public class Test1
{
public static void main(String[] args) {
Scanner obk=new Scanner(System.in);
arithop obj=new arithop();
int ch=0;
//l=obj1.nextInt();
while(ch<5) {
System.out.println("SELECT OPERATION:"+" 1.addition  2.multiplication  3.subtraction  4.division  5.Exit");
ch=obk.nextInt();
switch(ch) {
case 1:
obj.addition();
break;
case 2:
obj.multiplication();
break;
case 3:
obj.subtraction();
break;
case 4:
obj.division();
break;
case 5: break;
```

```
default:
System.out.println("Invalid");
break;
}
}
}
}
```

OUTPUT



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

Aim:

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

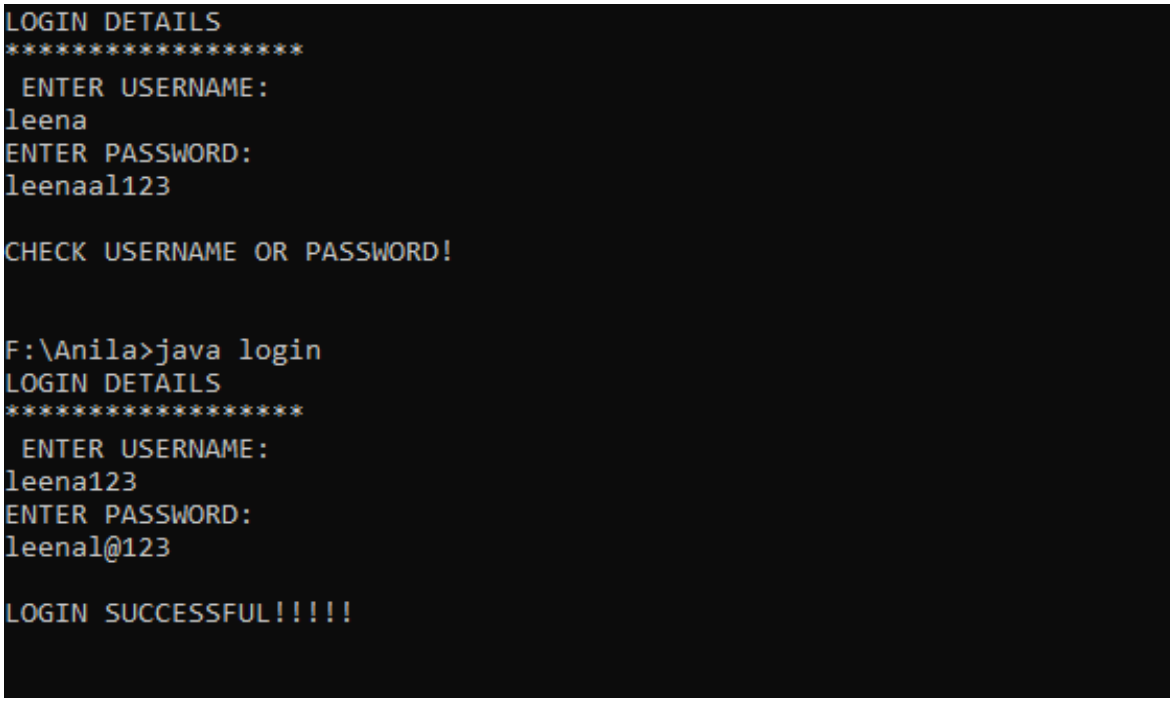
Source code:

login.java

```
import java.util.Scanner;
public class login{
public static void main(String[] args)
{
try {
int n, n2;
String username,password;
Scanner sc = new Scanner(System.in);
System.out.println("LOGIN DETAILS");
System.out.println("******************");
System.out.println(" ENTER USERNAME:");
username=sc.nextLine();
System.out.println("ENTER PASSWORD:");
password= sc.nextLine();
n=password.length();
n2=username.length();
if(n<6 || n2<6){
throw new ArithmeticException("\nCHECK
USERNAME OR PASSWORD!\n");
}else{
System.out.println("\nLOGIN SUCCESSFUL!!!!!\n");
}
```

```
}
catch(Exception e) {
System.out.println(e);
}
}
}
```

OUTPUT



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

Aim:

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

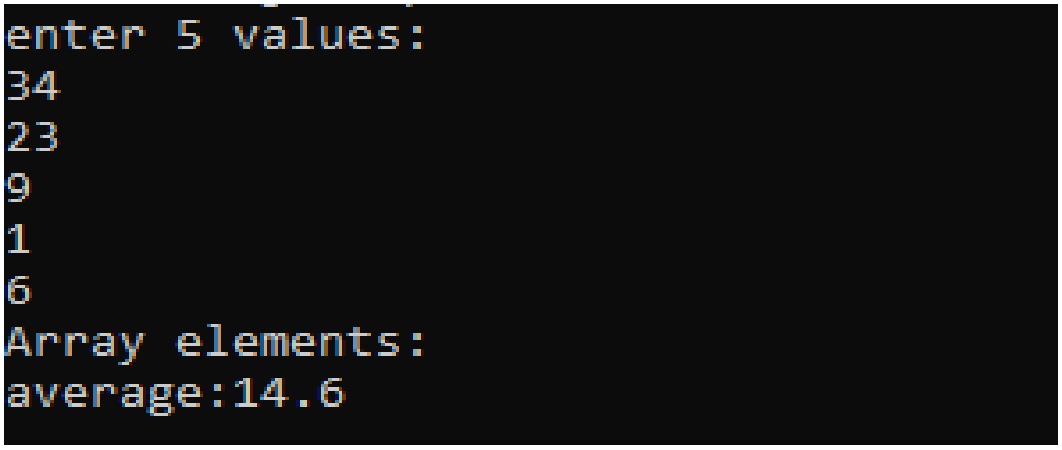
Source code:

positive.java

```
import java.io.*;
class myexcep extends Exception
{
myexcep(String msg)
{
super(msg);
}
}
class positive
{
public static void main(String []args) throws IOException
{
int a[]= new int [5];
int l=0,i,s=0,av=0;
BufferedReader br = new BufferedReader(new InputStreamReader (System.in));
try
{
System.out.println("enter 5 values:");
for(i=0;i<5;i++)
{
l=Integer.parseInt(br.readLine());
if(l<0)
throw new myexcep("enter only positive value");
```

```
else
a[i]=l;
av=i+1;
}
}
catch(myexcep e)
{
System.out.println("caught my exception");
System.out.println(e.getMessage());
}
System.out.println("Array elements:");
for(int j=0;j<5;j++)
{
s+=a[j];
}
System.out.println("average:"+(float)((float)s/av));
}
}
```

OUTPUT



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

Aim:

To define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads.

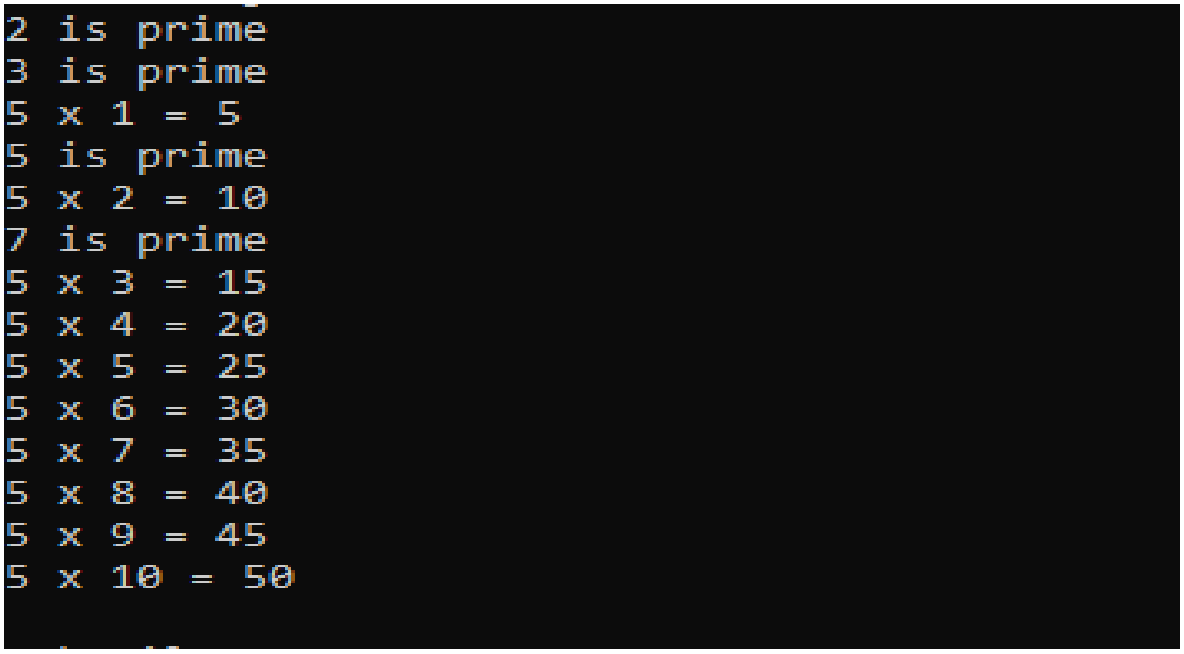
Source code:

thread1.java

```
import java.io.*;
class table extends Thread
{
public void run()
{
for(int i=1;i<=10;i++)
System.out.println("5 x "+i+" = "+(5*i));
}
}
class prime extends Thread
{
public void run()
{
int flag=0;
for(int i=2;i<=10;i++)
{
flag=0;
for(int j=2;j<=i/2;j++)
{
if(i%j==0)
{
```

```
flag=1;
break;
}
}
if(flag==0)
System.out.println(i +" is prime");
}
}
}
class thread1
{
public static void main(String []args)
{
table t = new table();
prime p = new prime();
t.start();
p.start();
}
}
```

OUTPUT


2 is prime
3 is prime
5 x 1 = 5
5 is prime
5 x 2 = 10
7 is prime
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

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

Aim:

To define 2 classes for generating Fibonacci numbers and for displaying even numbers in a given range. Implement using threads.

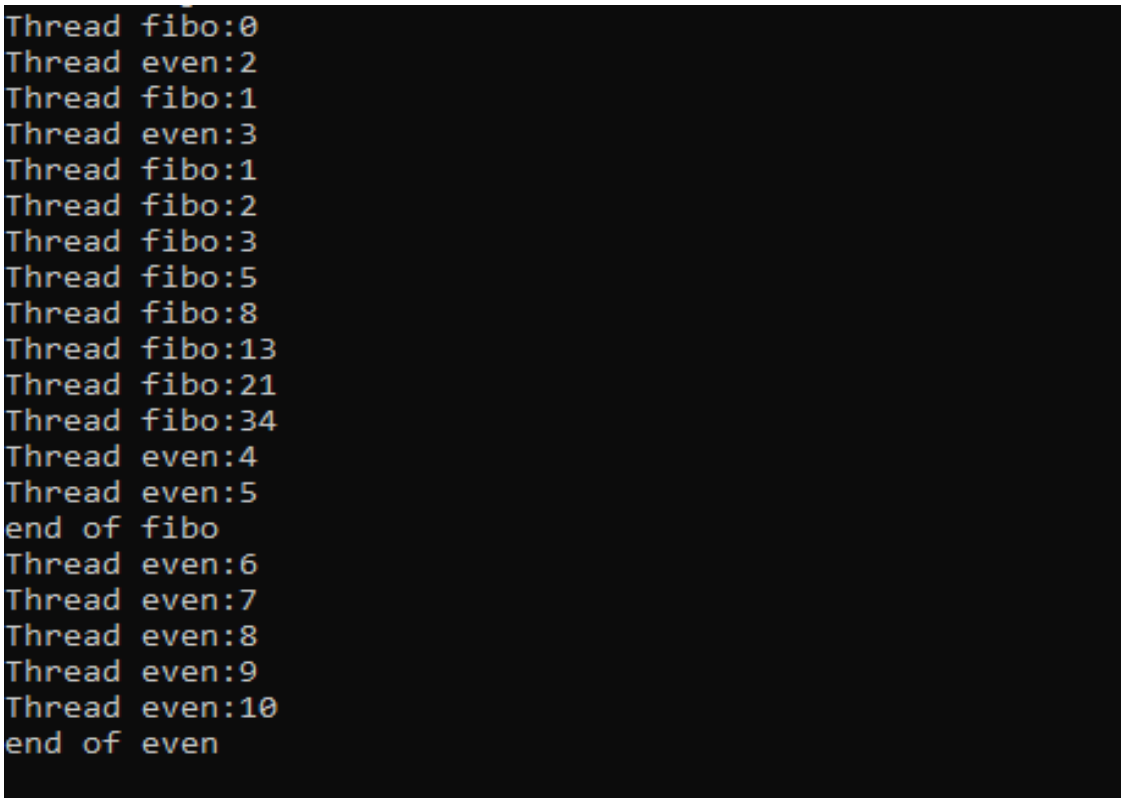
Source code:

thread2.java

```
import java.io.*;
class fibo implements Runnable
{
int a=0,b=1,c;
public void run()
{
System.out.println("Thread fibo:"+a);
System.out.println("Thread fibo:"+b);
for(int i=3;i<=10;i++)
{
c=a+b;
a=b;
b=c;
System.out.println("Thread fibo:"+c);
}
System.out.println("end of fibo");
}
}
class even implements Runnable
{
public void run()
```

```
{
for(int i=2;i<=10;i++)
System.out.println("Thread even:"+i);
System.out.println("end of even");
}
}
class thread2
{
public static void main(String []args)
{
fibo f= new fibo();
even e =new even();
Thread t1= new Thread(f);
Thread t2= new Thread(e);
t1.start();
t2.start();
}
     }
```

OUTPUT


Thread fibo:0
Thread even:2
Thread fibo:1
Thread even:3
Thread fibo:1
Thread fibo:2
Thread fibo:3
Thread fibo:5
Thread fibo:8
Thread fibo:13
Thread fibo:21
Thread fibo:34
Thread even:4
Thread even:5
end of fibo
Thread even:6
Thread even:7
Thread even:8
Thread even:9
Thread even:10
end of even

23) Producer/Consumer using ITC

Aim:

To write a program for producer/ consumer using ITC

Source code:

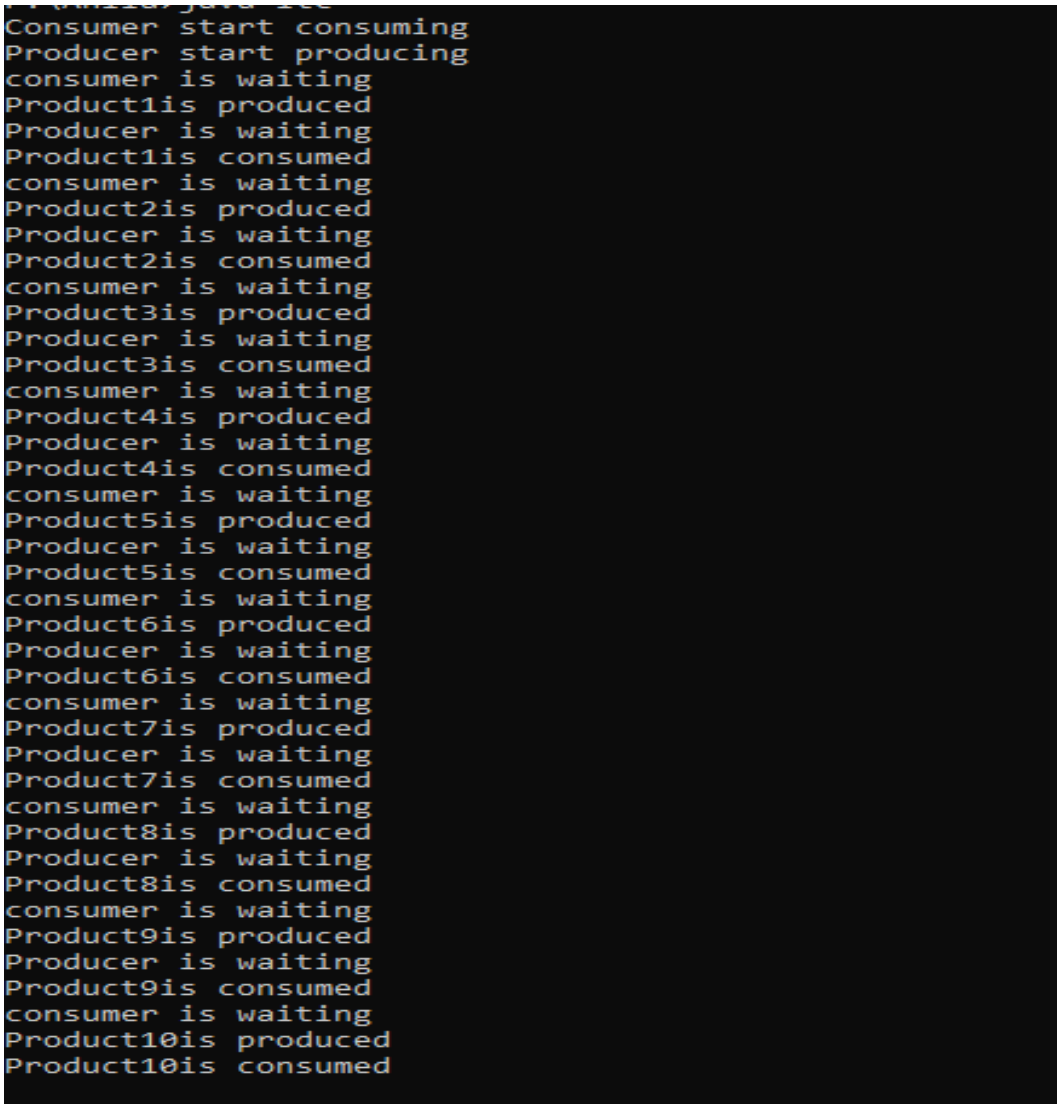
itc.java

```
import java.io.*;
class Buffer
{
int a;
boolean produced=false;
public synchronized void produce(int x)
{
if(produced)
{
System.out.println("Producer is waiting");
try
{wait();}
catch(Exception e)
{System.out.println(e);}
}
a=x;
System.out.println("Product"+a+ "is produced");
produced=true;
notify();
}
public synchronized void consume()
{
if(!produced)
```

```
{
System.out.println("consumer is waiting");
try
{wait();}
 catch(Exception e)
{System.out.println(e);}
}
System.out.println("Product"+a+ "is consumed");
produced=false;
notify();
}
}
class producer extends Thread
{
Buffer b;
public producer(Buffer b)
{ this.b=b;}
public void run()
{
System.out.println("Producer start producing");
for(int i =1;i<=10;i++)
b.produce(i);
}
}
class consumer extends Thread
{
Buffer b;
```

```
public consumer(Buffer b)
{ this.b=b;}
public void run()
{
System.out.println("Consumer start consuming");
for(int i =1;i<=10;i++)
b.consume();
}
}
public class itc
{
public static void main(String []args)
{
Buffer b = new Buffer();
producer p = new producer(b);
consumer c= new consumer(b);
p.start();
c.start();
}
}
```

OUTPUT


Consumer start consuming
Producer start producing
consumer is waiting
Product1is produced
Producer is waiting
Product1is consumed
consumer is waiting
Product2is produced
Producer is waiting
Product2is consumed
consumer is waiting
Product3is produced
Producer is waiting
Product3is consumed
consumer is waiting
Product4is produced
Producer is waiting
Product4is consumed
consumer is waiting
Product5is produced
Producer is waiting
Product5is consumed
consumer is waiting
Product6is produced
Producer is waiting
Product6is consumed
consumer is waiting
Product7is produced
Producer is waiting
Product7is consumed
consumer is waiting
Product8is produced
Producer is waiting
Product8is consumed
consumer is waiting
Product9is produced
Producer is waiting
Product9is consumed
consumer is waiting
Product10is produced
Product10is consumed

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

Aim:

To write a program to create generic stack and do push and pop operations

Source code:

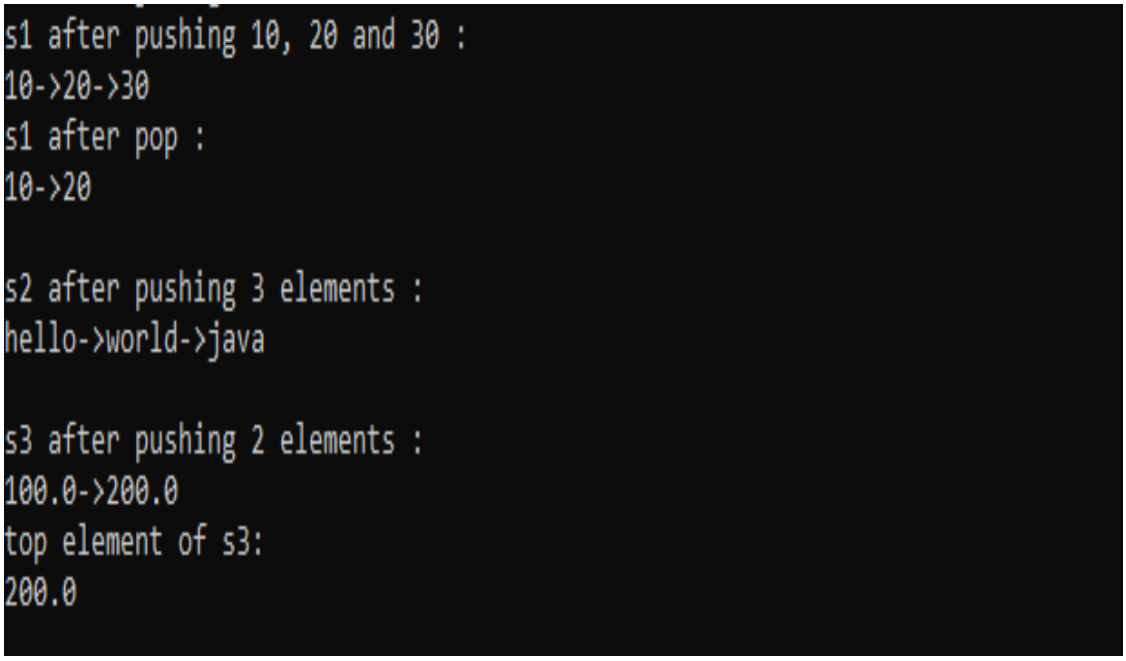
genstack.java

```
import java.io.*;
import java.util.*;
class stack<T>
{
ArrayList<T> A;
int top = -1;
int size;
stack(int size)
{
this.size = size;
this.A = new ArrayList<T>(size);
 }
 void push(T X)
 {
         if (top + 1 == size)
{
        System.out.println("Stack Overflow");
  }
     else
 {
   top = top + 1;
  // Over-writing existing element
   /*  if (A.size() > top)
   A.set(top, X);
    else
```

```
   // Creating new element*/
  A.add(X);
   }
}
T top()
{
if (top == -1)
{
System.out.println("Stack Underflow");
return null;
}
else
return A.get(top);
}
void pop()
{
if (top == -1)
System.out.println("Stack Underflow");
 else
 top--;
}
boolean empty() { return top == -1; }
 public String toString()
{
String Ans = "";
for (int i = 0; i < top; i++)
Ans += String.valueOf(A.get(i)) + "->";
Ans += String.valueOf(A.get(top));
return Ans;
```

```
}
}
// Main Class
public class genstack
{
 public static void main(String[] args)
   {
          stack<Integer> s1 = new stack<Integer>(3);
          s1.push(10);
          s1.push(20);
          s1.push(30);
          System.out.println("s1 after pushing 10, 20 and 30 :\n" + s1);
          s1.pop();
          System.out.println("s1 after pop :\n" + s1);
           stack<String> s2 = new stack<String>(3);
          s2.push("hello");
          s2.push("world");
          s2.push("java");
          System.out.println("\ns2 after pushing 3 elements :\n" + s2);
         stack<Float> s3 = new stack<Float>(2);
         s3.push(100.0f);
          s3.push(200.0f);
        System.out.println("\ns3 after pushing 2 elements :\n" + s3);
        System.out.println("top element of s3:\n"+ s3.top());
          }
}
```

OUTPUT


s1 after pushing 10, 20 and 30 :
10->20->30
s1 after pop :
10->20
s2 after pushing 3 elements :
hello->world->java
s3 after pushing 2 elements :
100.0->200.0
top element of s3:
200.0

25) Using generic method perform Bubble sort.

Aim:

To write a program to perform bubble sort using generic method

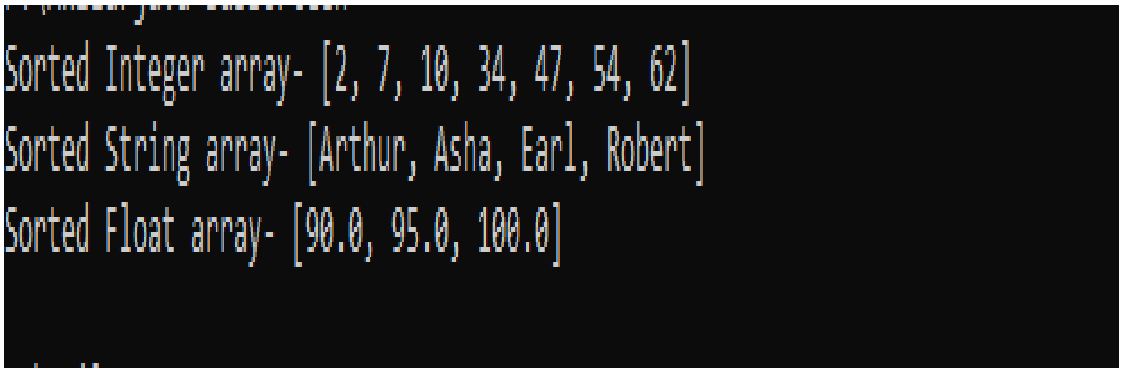
Source code:

BubSortGen.java

```
import java.util.Arrays;
public class BubSortGen<T extends Comparable<? super T>>
{
T[] array;
BubSortGen(T[] array)
{
this.array = array;
}
private T[] bubbleSort()
{
for(int i = array.length; i > 1; i--)
{
for(int j = 0; j < i - 1; j++)
{
  if(array[j].compareTo(array[j+1]) > 0)
  swapElements(j, array);
}
}
return array;
}
private void swapElements(int index, T[] arr)
{
T temp = arr[index];
arr[index] = arr[index+1];
arr[index+1] = temp;
```

```
}
public static void main(String[] args)
{
Integer[] intArr = {47, 62, 34, 7, 10,  2, 54};
BubSortGen<Integer> bsg1 = new BubSortGen<Integer>(intArr);
Integer[] sa1 = bsg1.bubbleSort();
System.out.println("Sorted Integer array- " + Arrays.toString(sa1));
String[] strArr = {"Earl", "Robert", "Asha", "Arthur"};
BubSortGen<String> bsg2 = new BubSortGen<String>(strArr);
String[] sa2 = bsg2.bubbleSort();
System.out.println("Sorted String array- " + Arrays.toString(sa2));
Float[] fArr = {100.0f, 90.0f,95.0f};
BubSortGen<Float> bsg3 = new BubSortGen<Float>(fArr);
Float[] sa3 = bsg3.bubbleSort();
System.out.println("Sorted Float array- " + Arrays.toString(sa3));
}
}
```

OUTPUT


Sorted Integer array- [2, 7, 10, 34, 47, 54, 62]
Sorted String array- [Arthur, Asha, Earl, Robert]
Sorted Float array- [90.0, 95.0, 100.0]

26) Maintain a list of Strings using Array List from collection framework, perform built-in operations.

Aim:

To write a program to maintain a list of string using Array list from collection frame work, perform built-in operations

Source code:

arrlist.java

```
import java.io.*;
import java.util.*;
class arrlist
{
public static void main(String[] args)
{
int n = 5;
ArrayList<Integer> arrli  = new ArrayList<Integer>(n);
for (int i = 1; i <= n; i++)
arrli.add(i);
System.out.println(arrli);
arrli.remove(3);
System.out.println(arrli);
arrli.add(2,33);
arrli.set(3,44);
for (int i = 0; i < arrli.size(); i++)
System.out.print(arrli.get(i) + " ");
}
}
```

OUTPUT

27) Program to remove all the elements from a linked list.

Aim:

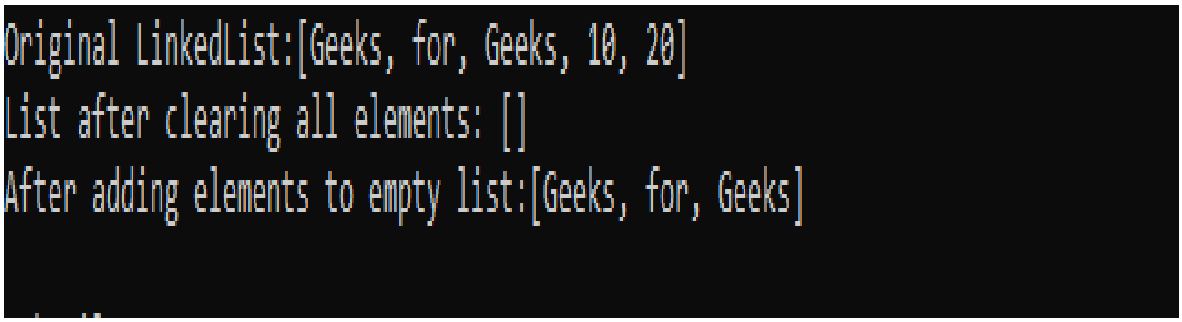
To write a program to remove all elements from linked list

Source code:

LLDemo.java

```
import java.io.*;
import java.util.LinkedList;
public class LLDemo {
public static void main(String args[])
   {
 // Creating an empty LinkedList
      LinkedList<String> list = new LinkedList<String>();
// Using add() method to add elements in the list
      list.add("Geeks");
      list.add("for");
      list.add("Geeks");
      list.add("10");
      list.add("20");
 // Displaying the List
      System.out.println("Original LinkedList:" + list);
// Clearing the list
      list.clear();
// Accessing the List after clearing it
      System.out.println("List after clearing all elements: " + list);
// Adding elements after clearing the list
      list.add("Geeks");
      list.add("for");
      list.add("Geeks");
 // Displaying the List
      System.out.println("After adding elements to empty list:" + list); }}
```

OUTPUT


Original LinkedList:[Geeks, for, Geeks, 10, 20]
List after clearing all elements: []
After adding elements to empty list:[Geeks, for, Geeks]

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

Aim:

To write a Program to remove an object from the Stack when the position is passed as parameter.

Source code:

stack.java

```
import java.io.*;
import java.util.*;
public class stack
{
public static void main(String []args)
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
Stack <String> s= new Stack<String>();
try
{
System.out.println("enter 5 string values");
for(int i=0;i<5;i++)
s.add(br.readLine());
System.out.println("Stack elements are:"+s);
int pos=0;
System.out.println("enter the position of element to remove object:");
pos=Integer.parseInt(br.readLine());
String st= s.remove(pos);
System.out.println("Stack element removed:"+st);
System.out.println("Remaining Stack elements are:"+s);
}
catch(IOException e)
{System.out.println(e);}
} }
```

OUTPUT

```
enter 5 string values
leena
anu
ann
rose
achu
Stack elements are:[leena, anu, ann, rose, achu]
enter the position of element to remove object:
3
Stack element removed:rose
Remaining Stack elements are:[leena, anu, ann, achu]
```

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

Aim:

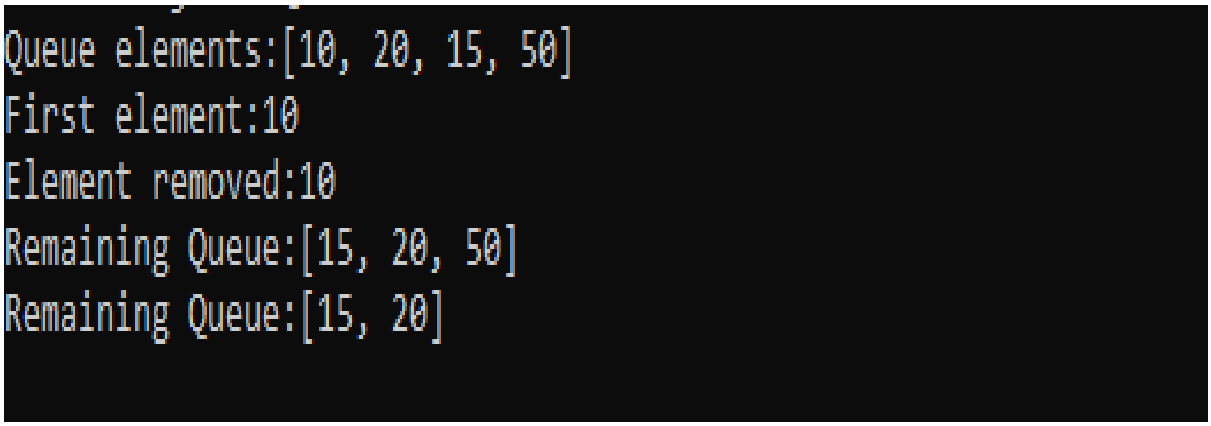
To write a program to demonstrate the creation of queue object using the PriorityQueue class.

Source code:

PQueue.java

```
import java.util.*;
class PQueue {
public static void main(String args[])
   {
PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>();
// Adding items to the pQueue using add()
      pQueue.add(10);
      pQueue.add(20);
      pQueue.add(15);
      pQueue.add(50);
      System.out.println("Queue elements:"+pQueue);
      System.out.println("First element:"+pQueue.peek());
      System.out.println("Element removed:"+pQueue.poll());
      System.out.println("Remaining Queue:"+pQueue); //printing queue”);
      pQueue.remove(50);
      System.out.println("Remaining Queue:"+pQueue); //printing queue”);
   }
}
```

OUTPUT


Queue elements:[10, 20, 15, 50]
First element:10
Element removed:10
Remaining Queue:[15, 20, 50]
Remaining Queue:[15, 20]

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

Aim:

To write a program to demonstrate the addition and deletion of elements in deque.

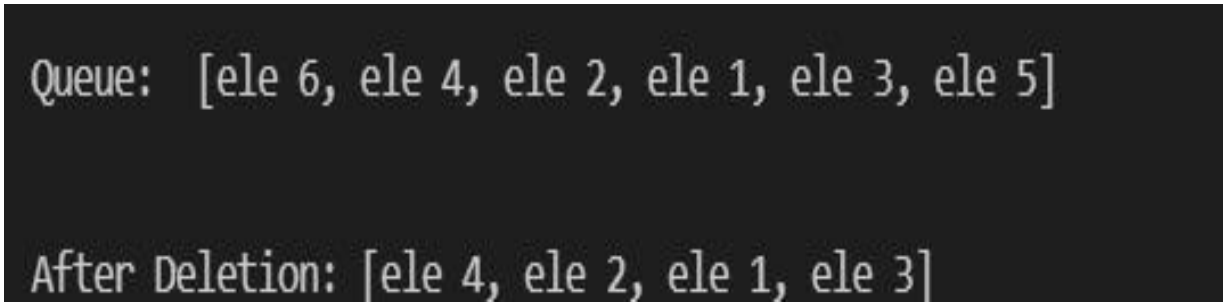
Source code:

deque.java

```
import java.util.*;
public class deque
   {
public static void main(String[] args)
        {
Deque<String> dq= new LinkedList<String>();
// Add at the last
      dq.add("Element 1 (Tail)");
// Add at the first
      dq.addFirst("Element 2 (Head)");
 // Add at the last
      dq.addLast("Element 3 (Tail)");
// Add at the first
      dq.push("Element 4 (Head)");
// Add at the last
      dq.offer("Element 5 (Tail)");
// Add at the first
      dq.offerFirst("Element 6 (Head)");
      System.out.println(dq + "\n");
// We can remove the first element
// or the last element.
    dq.removeFirst();
     dq.removeLast();
```

```
        System.out.println("Deque after removing " + "first and last: " + dq);}
}
```

OUTPUT



31) Program to demonstrate the creation of Set object using the LinkedHashset class.

Aim:

To write a program to demonstrate the creation of set object using the linkedhashset class.

Source code:

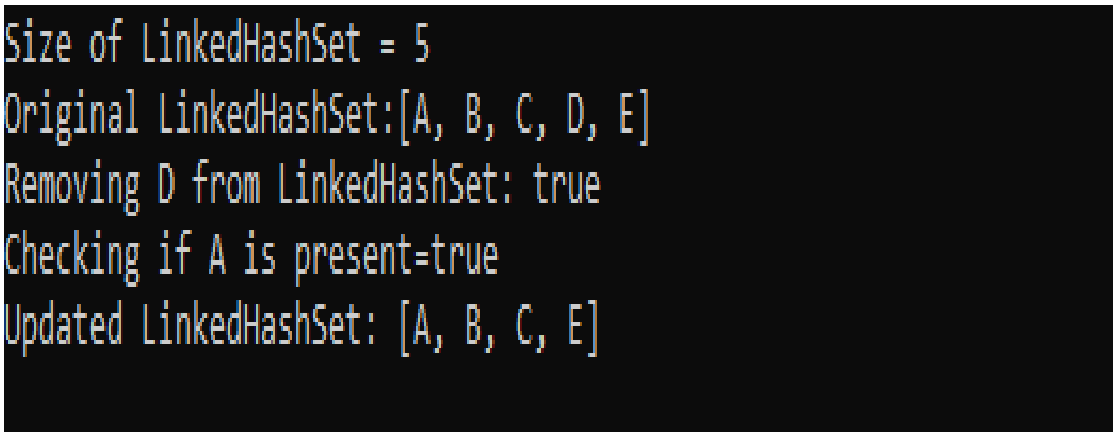
LHSet.java

```
import java.util.LinkedHashSet;

public class LHSet
{
// Main Method
   public static void main(String[] args)
   {
      LinkedHashSet<String> ls = new LinkedHashSet<String>();
  // Adding element to LinkedHashSet
      ls.add("A");
      ls.add("B");
      ls.add("C");
      ls.add("D");
      // This will not add new element as A already exists
      ls.add("A");
      ls.add("E");
      System.out.println("Size of LinkedHashSet = " + ls.size());
      System.out.println("Original LinkedHashSet:" + ls);
      System.out.println("Removing D from LinkedHashSet: "
      +ls.remove("D"));
      System.out.println("Checking if A is present=" + ls.contains("A"));
      System.out.println("Updated LinkedHashSet: " + ls);
```

```
    }
}
```

OUTPUT


Size of LinkedHashSet = 5
Original LinkedHashSet:[A, B, C, D, E]
Removing D from LinkedHashSet: true
Checking if A is present=true
Updated LinkedHashSet: [A, B, C, E]

32) Write a Java program to compare two hash set.

Aim:

To write a program to compare two hash set

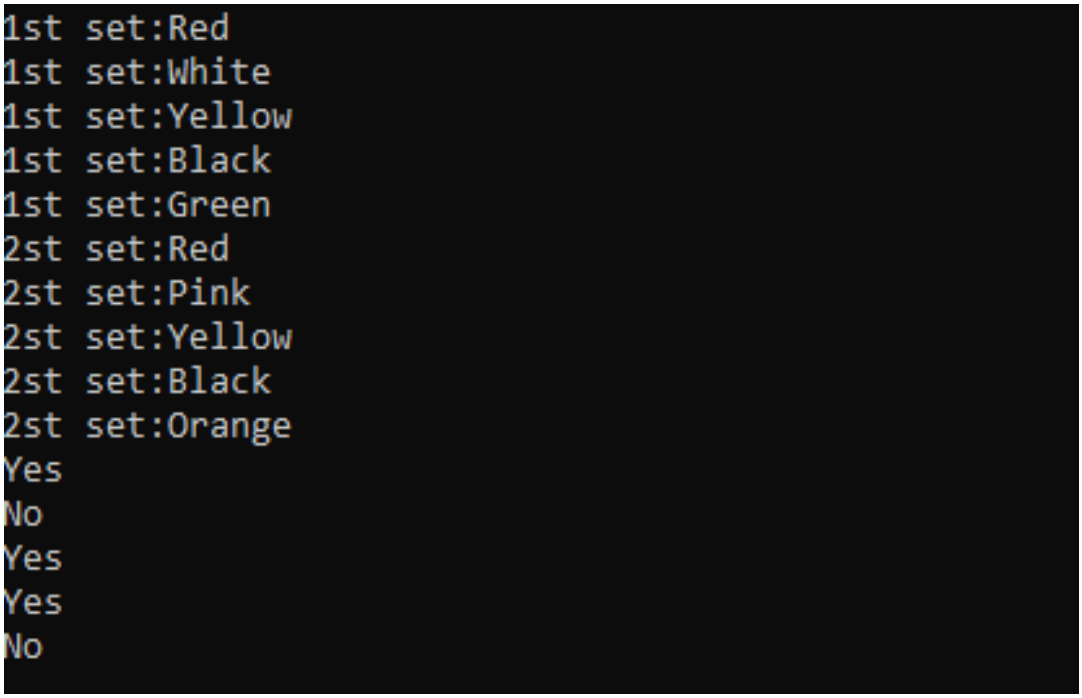
Source code:

hset.java

```
import java.util.*;
public class hset
{
        public static void main(String[] args)
{
// Create a empty hash set
        HashSet<String> h_set = new HashSet<String>();
// use add() method to add values in the hash set
         h_set.add("Red");
         h_set.add("Green");
         h_set.add("Black");
         h_set.add("White");
        h_set.add("Yellow");
        for (String element : h_set)
        System.out.println("1st set:"+element);
        HashSet<String> h_set2 = new HashSet<String>();
         h_set2.add("Red");
         h_set2.add("Pink");
         h_set2.add("Black");
         h_set2.add("Orange");
          h_set2.add("Yellow");
          for (String element : h_set2)
          System.out.println("2st set:"+element);
```

```
//comparison output in hash set
        for (String element : h_set)
         System.out.println(h_set2.contains(element) ? "Yes" : "No");
         }
       }
```

OUTPUT



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

Aim:

To write a program to demonstrate the working of Map interface by adding, changing and removing elements.

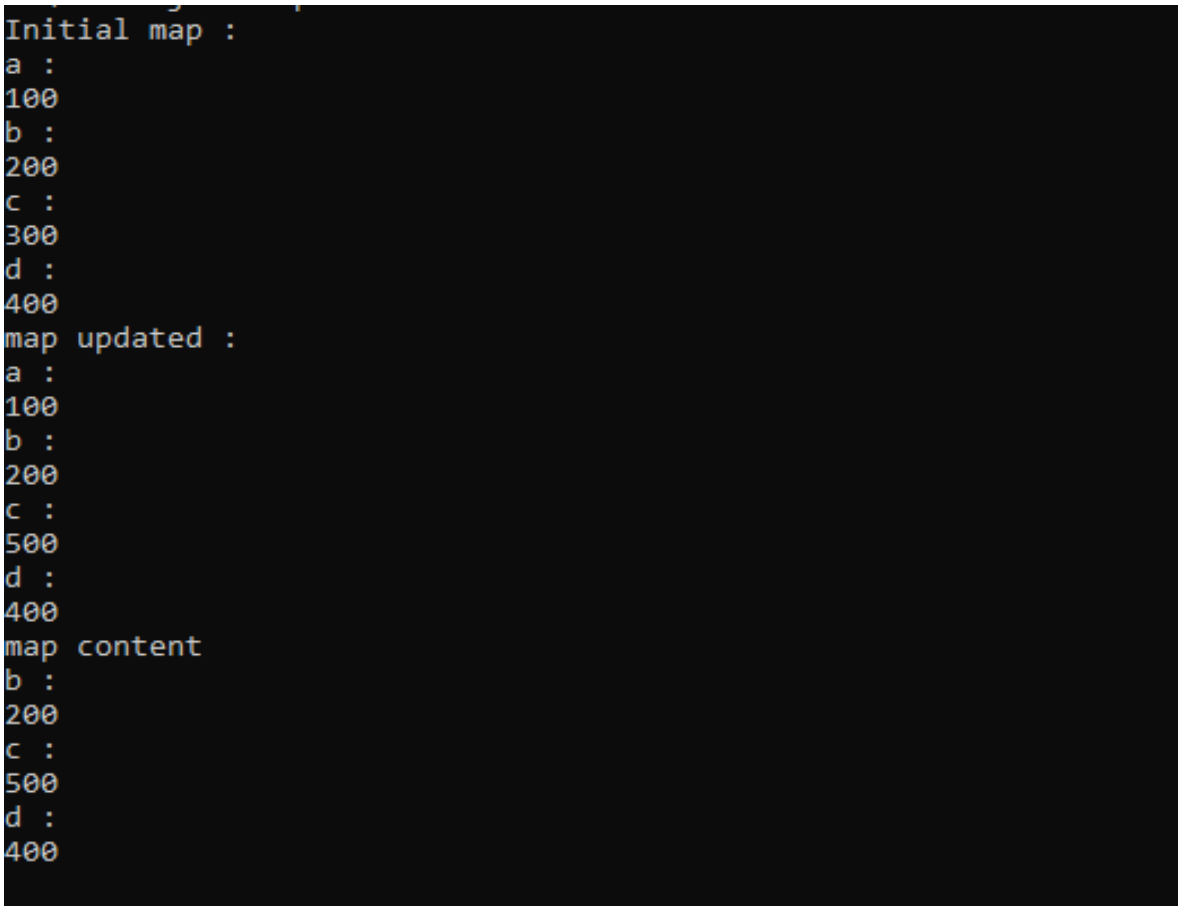
Source code:

map.java

```
import java.util.*;
class map
{
public static void main(String args[])
        {
        HashMap<String, Integer> hm = new HashMap<String, Integer>();
        hm.put("a",Integer(100));
        hm.put("b", Integer(200));
        hm.put("c", Integer(300));
        hm.put("d", Integer(400));
// Traversing through the map, the insertion order is not retained in the
   hashmap.
System.out.print("initial map contents:");
for (HashMap.Entry<String, Integer> me : hm.entrySet())
{
System.out.print(me.getKey() + ":");
System.out.println(me.getValue());
}
System.out.print("map contents after updation:");
hm.put("c",new Integer(500));
for (HashMap.Entry<String, Integer> me : hm.entrySet())
{
System.out.print(me.getKey() + ":");
System.out.println(me.getValue());
```

```
}
hm.remove("a");
System.out.print("map contents after removal:");
for (HashMap.Entry<String, Integer> me : hm.entrySet())
{
System.out.print(me.getKey() + ":");
System.out.println(me.getValue());
}
}
}
```

OUTPUT



34) Program to Convert HashMap to TreeMap.

Aim:

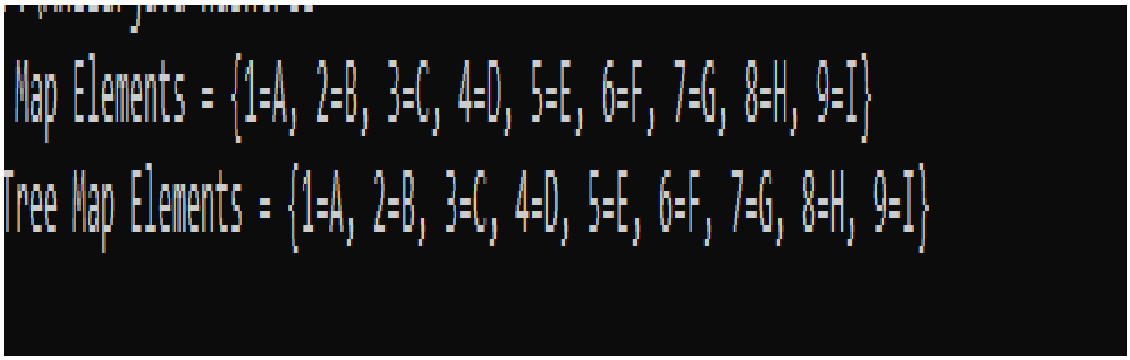
To write a program to convert hashmap to treemap.

Source code:

hashtree.java

```
import java.util.HashMap;
import java.util.TreeMap;
import java.util.Map;
public class hashtree
{
        public static void main(String[] a)
{
                Map<String, String> map = new HashMap<String, String>();
                map.put("1", "A");
                map.put("2", "B");
                map.put("4", "D");
                map.put("3", "C");
                map.put("5", "E");
                map.put("6", "F");
                map.put("8", "H");
                map.put("7", "G");
                map.put("9", "I");
System.out.println(" Map Elements = "+map);
Map<String, String>sorted = new TreeMap<String, String>(map);
//shows sorted form only
System.out.println("Tree Map Elements = "+sorted);
}
}
```

OUTPUT


Map Elements = {1=A, 2=B, 3=C, 4=D, 5=E, 6=F, 7=G, 8=H, 9=I}
Tree Map Elements = {1=A, 2=B, 3=C, 4=D, 5=E, 6=F, 7=G, 8=H, 9=I}

35) Program to draw Circle, Rectangle, Line in Applet.

Aim:

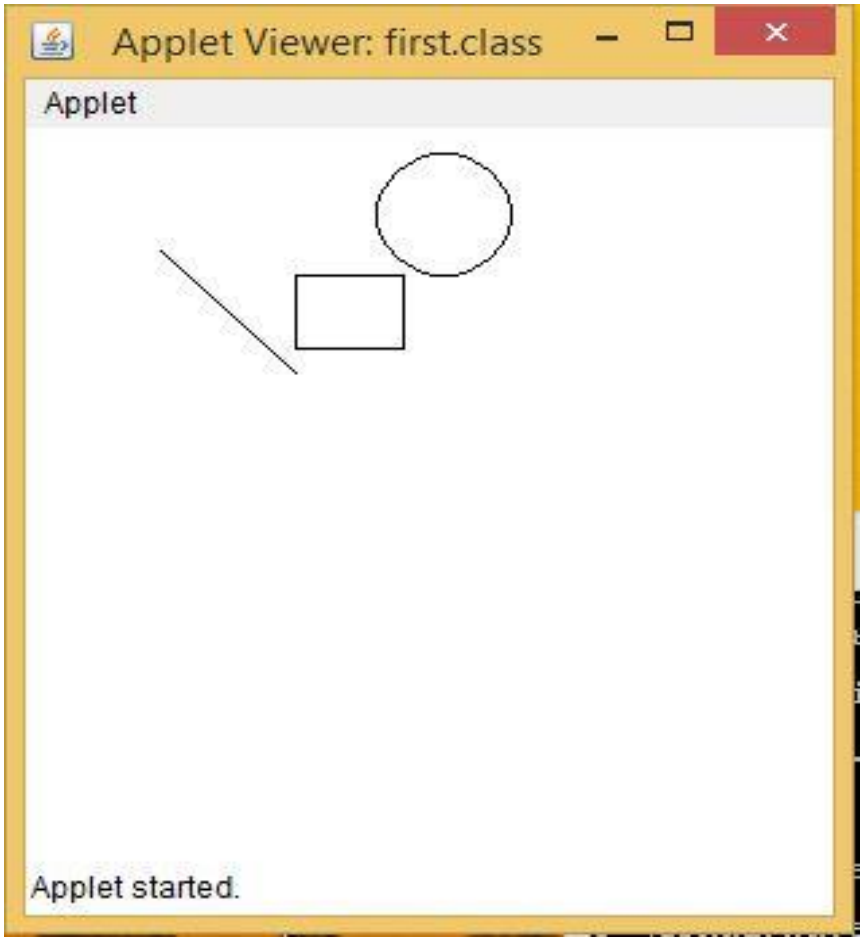
To write a program to draw Circle, Rectangle, Line in Applet.

Source code:

first.java

```
/* applet program to draw line, circle, rectangle*/
 import java.applet.Applet;
import java.awt.Graphics;
public class first extends Applet
{
        public void paint(Graphics g)
{
g.drawLine(100,100,50,50);
g.drawRect(100,60,40,30);
g.drawOval(130,10,50,50);
}
}
/*
<applet code="first.class" width="300" height="300">
</applet>
*/
```

OUTPUT



36) Program to find maximum of three numbers using AWT.

Aim:

To write a program to find maximum of three numbers using AWT

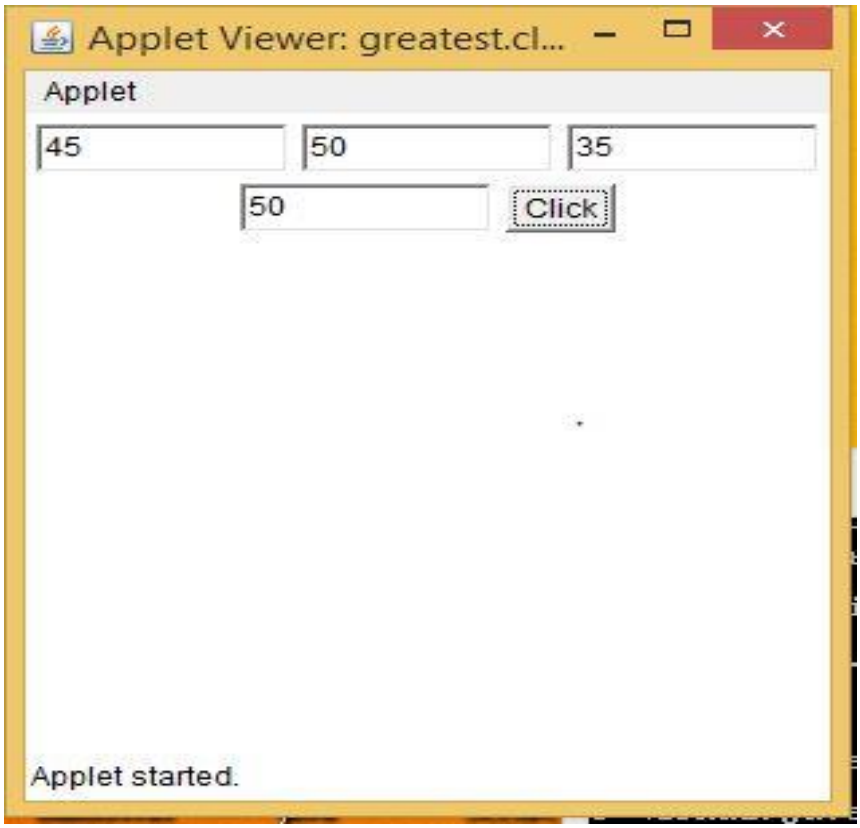
Source code:

greatest.java

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class greatest extends Applet implements ActionListener
{
TextField T1,T2,T3,T4;
Button B1;
public void init()
{
        T1 = new TextField(10);
        T2 = new TextField(10);
        T3 = new TextField(10);
        T4 = new TextField(10);
        B1 = new Button("Click");
add(T1);
add(T2);
add(T3);
add(T4);
add(B1);
T1.setText("");
T2.setText("");
T3.setText("");
T4.setText("");
B1.addActionListener(this);
          }
```

```
public void actionPerformed(ActionEvent e)
{
int a,b,c,big=0;
a=Integer.parseInt(T1.getText());
b=Integer.parseInt(T2.getText());
c=Integer.parseInt(T3.getText());
if (a>b)
{
if (a>c)
big=a;
else
big=c;
}
else
{
if (b>c)
big=b;
else
big=c;
}
T4.setText(""+big);
}
}
/*
<applet code="greatest.class" width="300" height="300">
</applet>
*/
```

OUTPUT


Applet Viewer: greatest.cl...
Applet
45
50
35
50
Click
Applet started.

37) 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.

Aim:

To write a program to find the percentage of marks obtained by a student in 5 subjects. Display a happy face if it is above 50% and sad face if not.

Source code:

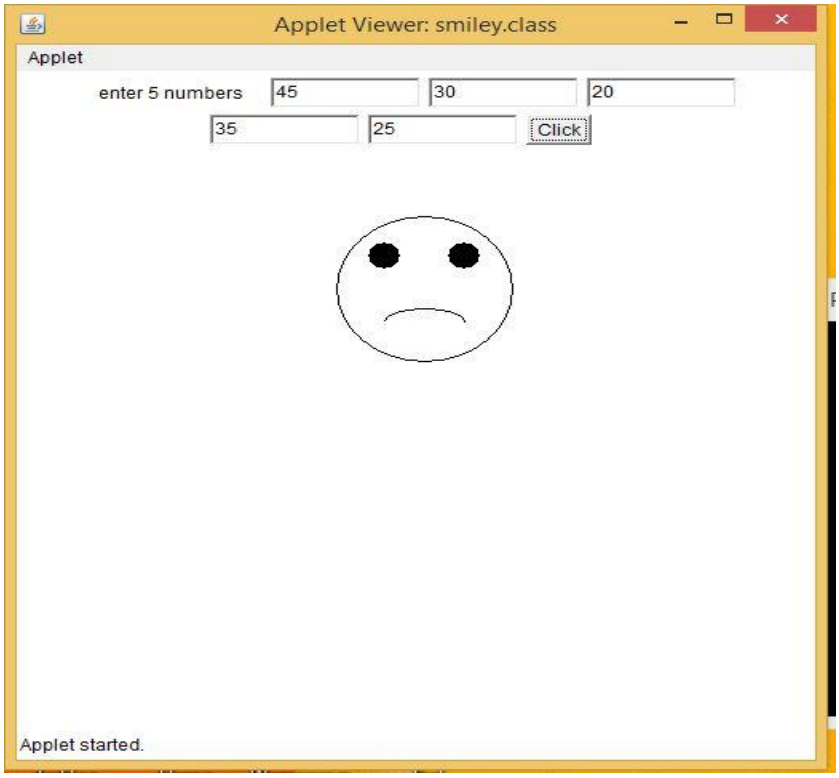
smiley.java

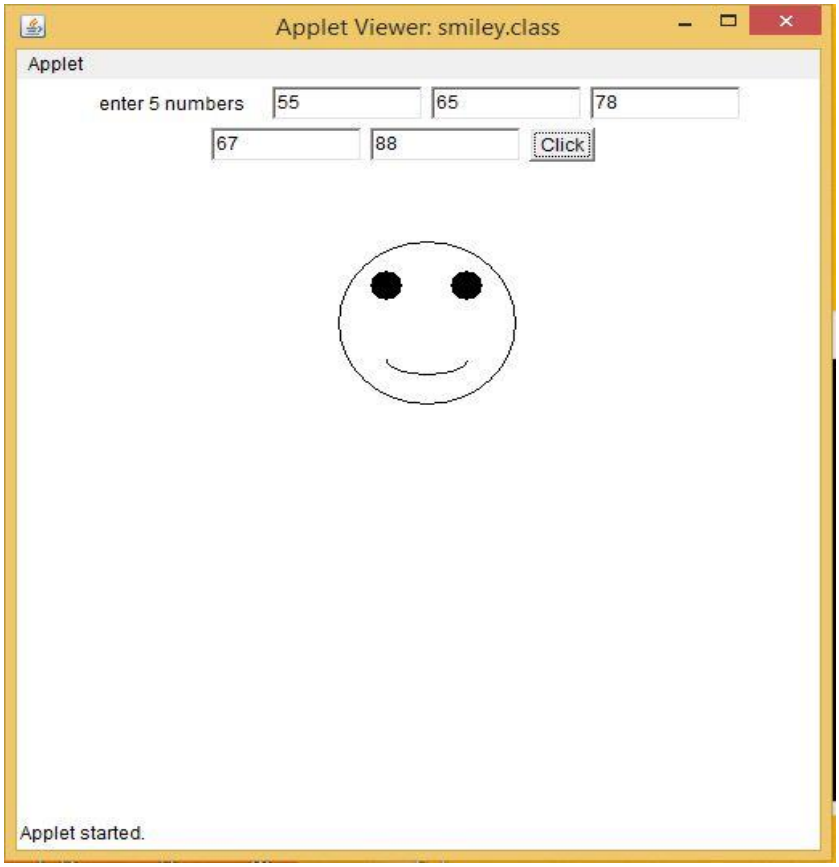
```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class smiley extends Applet implements ActionListener
{
TextField T1,T2,T3,T4,T5;
Label l,l2;
Button B1;
String str;
public smiley()
{
T1 = new TextField(10);
T2 = new TextField(10);
T3 = new TextField(10);
T4 = new TextField(10);
T5 = new TextField(10);
l = new Label("enter 5 numbers");
l2 = new Label();
B1 = new Button("Click");
add(l);
add(T1);
add(T2);
add(T3);
```

```
add(T4);
add(T5);
add(B1);
add(l2);
T1.setText("");
T2.setText("");
T3.setText("");
T4.setText("");
T5.setText("");
B1.addActionListener(this);
str="";
}
public void actionPerformed(ActionEvent e1)
{
int a,b,c,d,e,av=0;
a=Integer.parseInt(T1.getText());
b=Integer.parseInt(T2.getText());
c=Integer.parseInt(T3.getText());
d=Integer.parseInt(T2.getText());
e=Integer.parseInt(T3.getText());
av=(a+b+c+d+e)/5;
if(av>50)
str="yes";
else
str="no";
repaint();
}
public void paint(Graphics g)
{
```

```
super.paint(g);
g.drawString(str,550,550);
if(str=="yes")
{
g.drawOval(200,110,110,110);
g.setColor(Color.black);
g.fillOval(220,130,20,20);
g.fillOval(270,130,20,20);
g.drawArc(230,180,50,20,180,180);
}
else if(str=="no")
{
g.drawOval(200,110,110,110);
g.setColor(Color.black);
g.fillOval(220,130,20,20);
g.fillOval(270,130,20,20);
g.drawArc(230,180,50,20,-180,-180);
}
}
public static void main(String[] args)
{
new smiley();
}
}
/*
<applet code="smiley.class" width="500" height="500">
</applet>
*/
```

OUTPUT


Applet Viewer: smiley.class
Applet
enter 5 numbers
45
30
20
35
25
Click
Applet started.


Applet Viewer: smiley.class
Applet
enter 5 numbers
55
65
78
67
88
Click
Applet started.

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

Aim:

To write a program in 2D graphics command in an applet to construct a house. On mouse click event change the door colour.

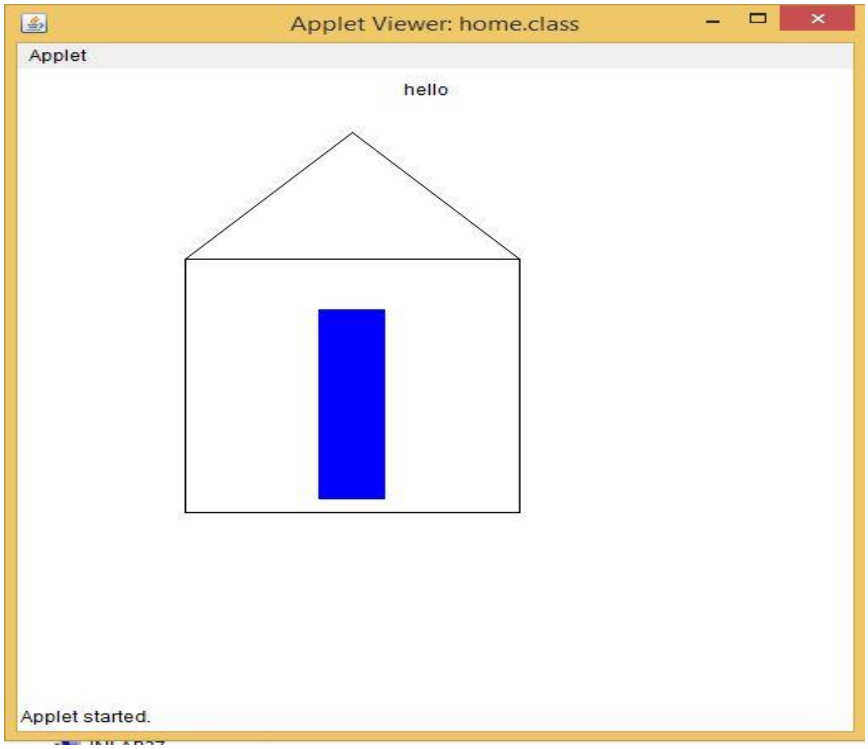
Source code:

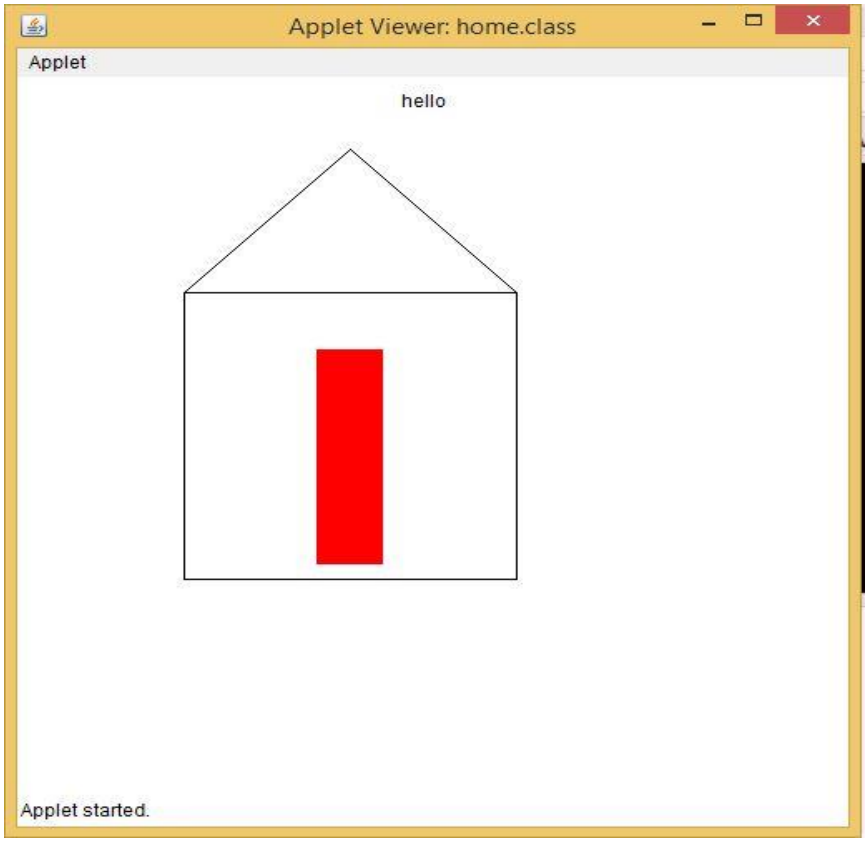
home.java

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class home extends Applet implements MouseListener
{
Color c =Color.blue;
public void init()
{
Label l;
l = new Label("hello");
l.setBounds(180,190,40,150);
add (l);
addMouseListener(this);
}
public void paint(Graphics g)
{
g.drawRect (100,150,200,200);
g.drawLine(100,150,200,50);
g.drawLine(300,150,200,50);
g.setColor(c);
g.fillRect (180,190,40,150);
}
public void mouseClicked(MouseEvent me)
```

```
{
c=Color.red;
// g.setColor(Color.red);
// g.fillRect (180,190,40,150);
repaint();
}
public void mouseExited(MouseEvent me){}
public void mouseMoved(MouseEvent me){}
public void mouseDragged(MouseEvent me){}
public void mouseEntered(MouseEvent me){}
public void mouseReleased(MouseEvent me){}
public void mousePressed(MouseEvent me){}
}
/*
<applet code="home.class" width="500" height="500">
</applet>
*/
```

OUTPUT


Applet Viewer: home.class
Applet
hello
Applet started.


Applet Viewer: home.class
Applet
hello
Applet started.

39) Implement a simple calculator using AWT components.

Aim:

To write a program to implement a simple calculator using AWT components

Source code:

calc.java

```
import java.awt.*;
import java.awt.event.*;
public class calc implements ActionListener
{
   int c,n;
   String s1,s2,s3,s4,s5;
   Frame f;
   Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;
   Panel p;
   TextField tf;
   GridLayout g;
   public calc()
   {
      f = new Frame("My calculator");
      p = new Panel();
      f.setLayout(new FlowLayout());
      b1 = new Button("0");
      b1.addActionListener(this);
      b2 = new Button("1");
      b2.addActionListener(this);
      b3 = new Button("2");
      b3.addActionListener(this);
      b4 = new Button("3");
      b4.addActionListener(this);
```

```
b5 = new Button("4");
b5.addActionListener(this);
b6 = new Button("5");
b6.addActionListener(this);
b7 = new Button("6");
b7.addActionListener(this);
b8 = new Button("7");
b8.addActionListener(this);
b9 = new Button("8");
b9.addActionListener(this);
b10 = new Button("9");
b10.addActionListener(this);
b11 = new Button("+");
b11.addActionListener(this);
b12 = new Button("-");
b12.addActionListener(this);
b13 = new Button("*");
b13.addActionListener(this);
b14 = new Button("/");
b14.addActionListener(this);
b15 = new Button("%");
b15.addActionListener(this);
b16 = new Button("=");
b16.addActionListener(this);
b17 = new Button("C");
b17.addActionListener(this);
tf = new TextField(20);
f.add(tf);
g = new GridLayout(4,4,10,20);
```

```
        p.setLayout(g);
p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);p.add(b6);p.add(b7);p.add(b8);p.add(b9);
p.add(b10);p.add(b11);p.add(b12);p.add(b13);p.add(b14);p.add(b15);p.add(b16);p.add(b17);
        f.add(p);
        f.setSize(300,300);
        f.setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
     if(e.getSource()==b1)
    {
            s3 = tf.getText();
            s4 = "0";
            s5 = s3+s4;
            tf.setText(s5);
     }
     if(e.getSource()==b2)
     {
            s3 = tf.getText();
            s4 = "1";
            s5 = s3+s4;
      tf.setText(s5);
     }
     if(e.getSource()==b3)
     {
            s3 = tf.getText();
            s4 = "2";
            s5 = s3+s4;
```

```
  tf.setText(s5);
}if(e.getSource()==b4)
{
    s3 = tf.getText();
    s4 = "3";
    s5 = s3+s4;
    tf.setText(s5);
}
if(e.getSource()==b5)
{
    s3 = tf.getText();
    s4 = "4";
    s5 = s3+s4;
    tf.setText(s5);
}
if(e.getSource()==b6)
{
    s3 = tf.getText();
    s4 = "5";
    s5 = s3+s4;
   tf.setText(s5);
}
if(e.getSource()==b7)
{
    s3 = tf.getText();
    s4 = "6";
    s5 = s3+s4;
    tf.setText(s5);
}
```

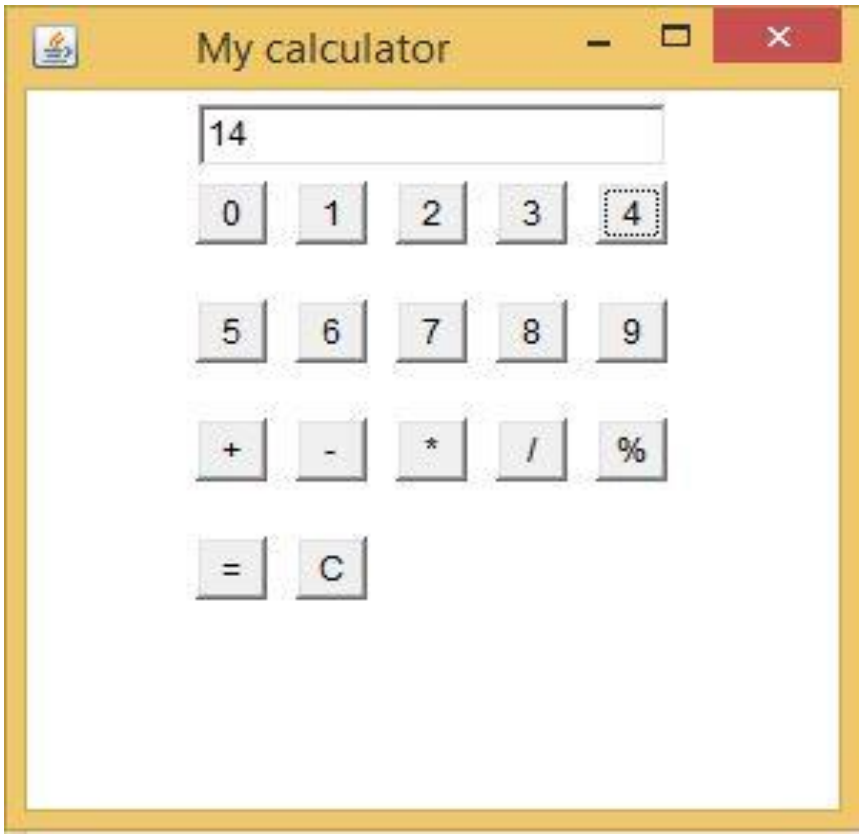
```
if(e.getSource()==b8)
{
   s3 = tf.getText();
   s4 = "7";
   s5 = s3+s4;
   tf.setText(s5);
}
if(e.getSource()==b9)
{
   s3 = tf.getText();
   s4 = "8";
   s5 = s3+s4;
   tf.setText(s5);
}
if(e.getSource()==b10)
{
   s3 = tf.getText();
   s4 = "9";
   s5 = s3+s4;
   tf.setText(s5);
}
if(e.getSource()==b11)
{
   s1 = tf.getText();
   tf.setText("");
   c=1;
}
if(e.getSource()==b12)
{
```

```
    s1 = tf.getText();
    tf.setText("");
    c=2;
 }
 if(e.getSource()==b13)
 {
    s1 = tf.getText();
    tf.setText("");
    c=3;
 }
 if(e.getSource()==b14)
 {
    s1 = tf.getText();
    tf.setText("");
    c=4;
 }
 if(e.getSource()==b15)
 {
    s1 = tf.getText();
    tf.setText("");
    c=5;
 }
 if(e.getSource()==b16)
 {
    s2 = tf.getText();
    if(c==1)
{
  n = Integer.parseInt(s1)+Integer.parseInt(s2);
  tf.setText(String.valueOf(n));
```

```
}
else
if(c==2)
{
    n = Integer.parseInt(s1)-Integer.parseInt(s2);
    tf.setText(String.valueOf(n));
}
else
if(c==3)
{
    n = Integer.parseInt(s1)*Integer.parseInt(s2);
    tf.setText(String.valueOf(n));
}
if(c==4)
{
 try
{
 int p=Integer.parseInt(s2);
  if(p!=0)
 {
  n = Integer.parseInt(s1)/Integer.parseInt(s2);
  tf.setText(String.valueOf(n));
  }
  else
  tf.setText("infinite");
    }
  catch(Exception i){}
  }
  if(c==5)
```

```
            {
                n = Integer.parseInt(s1)%Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
        }
        if(e.getSource()==b17)
        {
         tf.setText("");
        }
    }
    public static void main(String[] abc)
    {
     calc v = new calc();
    }
}
/*
<applet code="calc.class" width="500" height="500">
</applet>
*/
```

OUTPUT



40) 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.

Aim:

To develop a program that has choice components which contains names of the given shapes and draw the corresponding shapes as per users choice.

Source code:

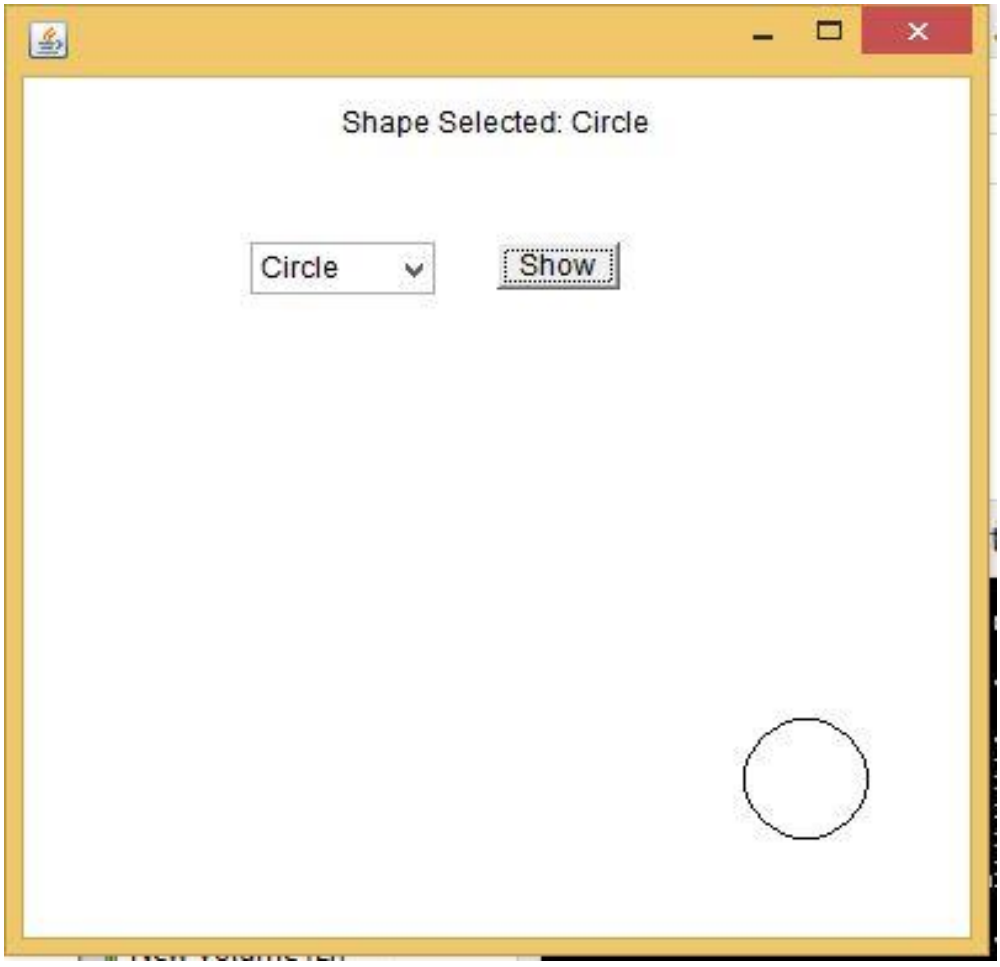
Drawchoice.java

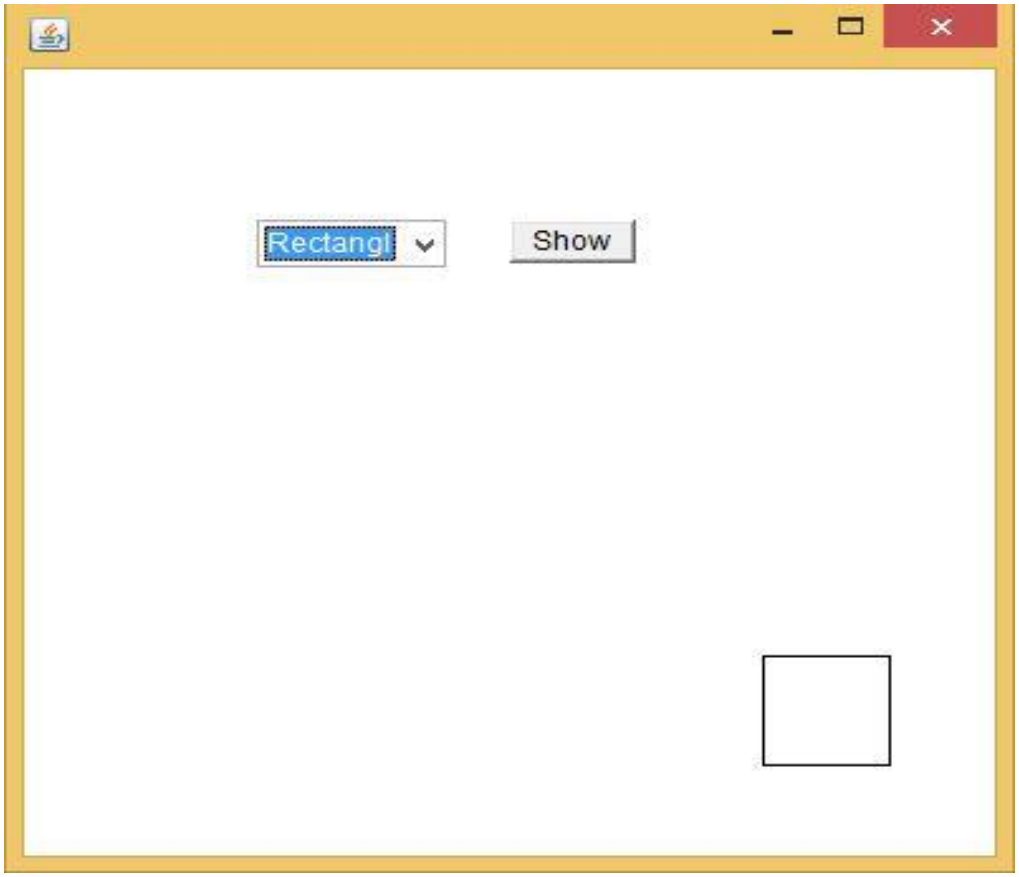
```
import java.awt.*;
import java.awt.event.*;
public class Drawchoice extends Frame implements ActionListener{
String data="";
String temp="";
Button b;
final Label label;
final Choice c;
public Drawchoice(){
b = new Button("Show");
b.addActionListener(this);
c = new Choice();
label = new Label();
c.setBounds(100, 100, 75, 75);
b.setBounds(200, 100, 50, 20);
label.setSize(400, 100);
label.setAlignment(Label.CENTER);
        c.add("Circle");
        c.add("Rectangle");
        c.add("Line");
        add(label);
        add(c);
```

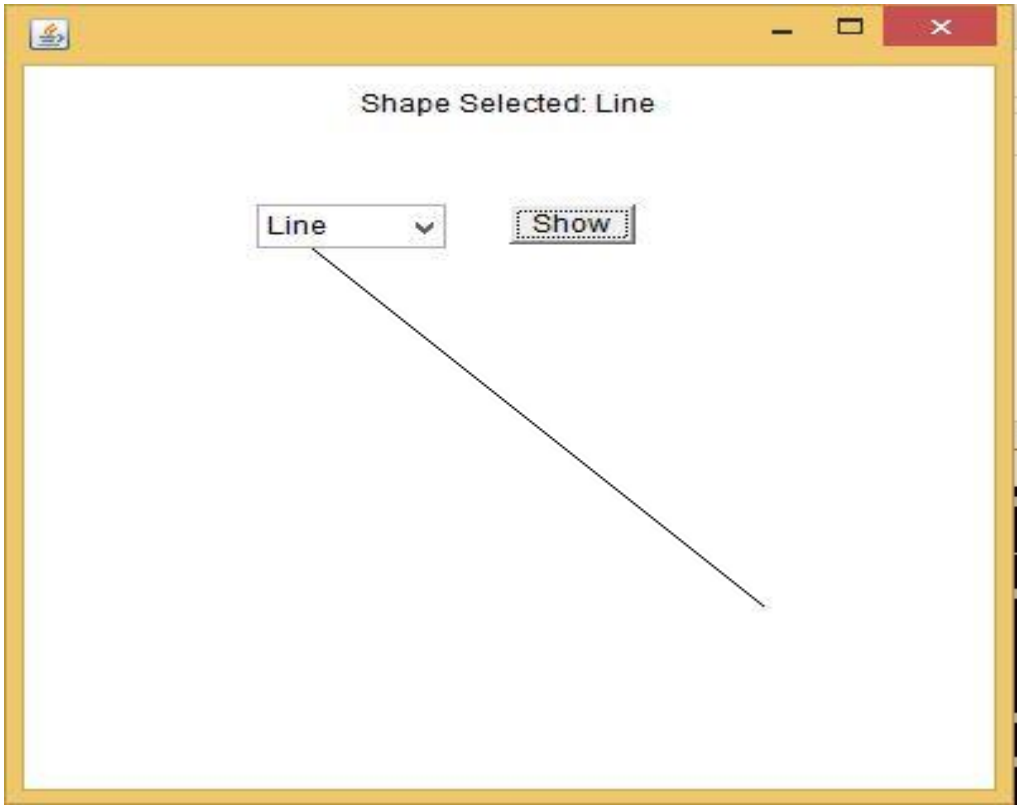
```
        add(b);
        setSize(400, 400);
        setLayout(null);
        setVisible(true);
  }
public void paint(Graphics g)
 {
super.paint(g);
if(data.equals("Line")){
g.drawLine(300,300,50,50);
}
else if(data.equals("Circle")){
g.drawOval(300,300,50,50);
}
else{
g.drawRect(300,300,50,50);
}
}
public void actionPerformed(ActionEvent e) {
if(e.getSource()==b){
temp = "Shape Selected: "+ c.getItem(c.getSelectedIndex());
data=c.getItem(c.getSelectedIndex());
label.setText(temp);
}
repaint();
}
public static void main(String[] args) {
 new Drawchoice();
}
```

```
}
/*
<applet code="Drawchoice.class" width="300" height="300">
</applet>*/
```

OUTPUT




Rectangl
Show


Shape Selected: Line
Line
Show

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

Aim:

To develop a program to handle all mouse events and window events.

Source code:

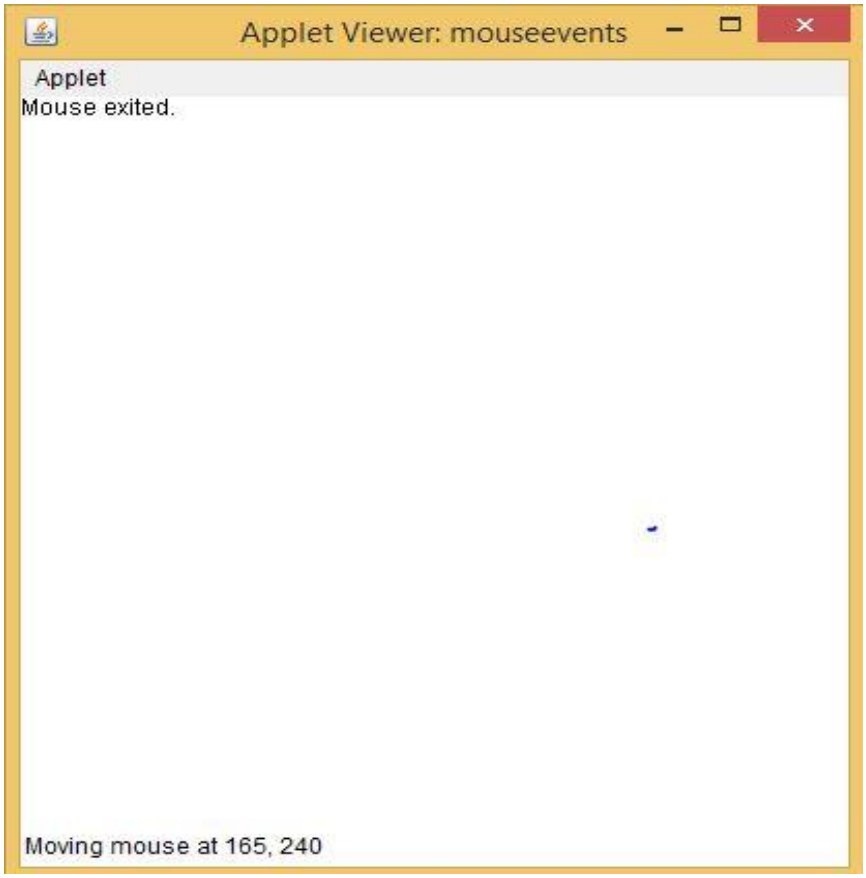
mouseevents.java

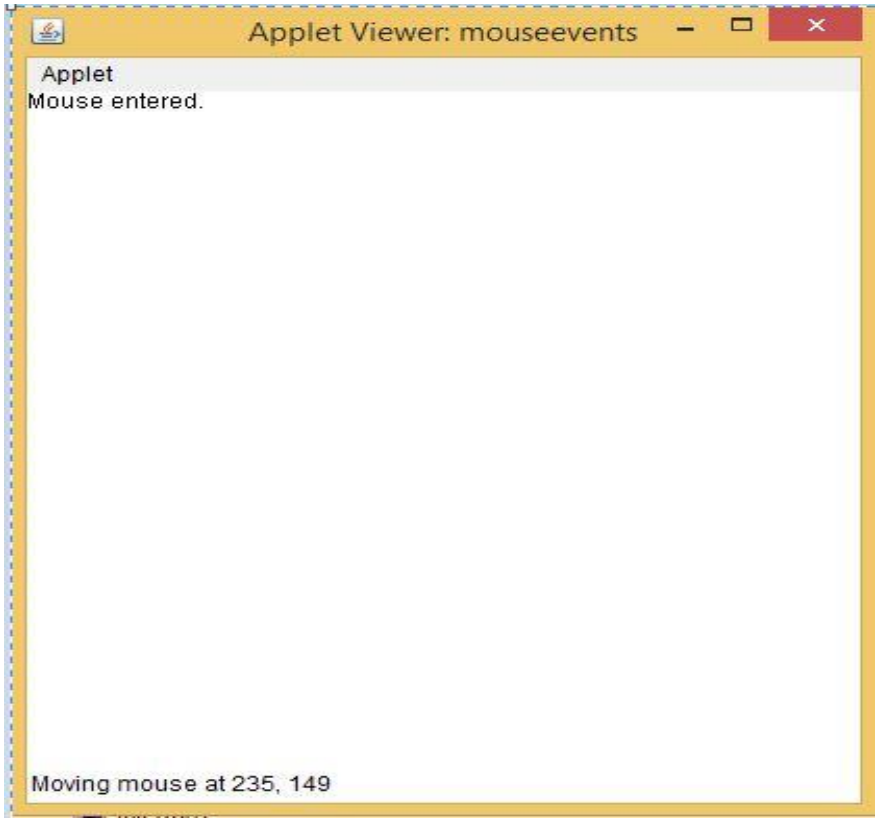
```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
 /*<applet code="mouseevents" width=400 height=400> </applet> */
public class mouseevents extends Applet implements MouseListener, MouseMotionListener
{
String msg = "";
int mouseX = 0, mouseY = 0; // coordinates of mouse
 public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}
// Handle mouse clicked.
public void mouseClicked(MouseEvent me)
{
// save coordinates
mouseX = 0;
mouseY = 10;
msg = "Mouse clicked.";
repaint();
}
// Handle mouse entered.
public void mouseEntered(MouseEvent me)
```

```
{
// save coordinates
mouseX = 0;
mouseY = 10;
msg = "Mouse entered.";
repaint();
}
// Handle mouse exited.
 public void mouseExited(MouseEvent me)
{
// save coordinates
mouseX = 0;
mouseY = 10;
msg = "Mouse exited.";
repaint();
}
// Handle button pressed.
 public void mousePressed(MouseEvent me)
{
// save coordinates
mouseX = me.getX();
mouseY = me.getY();
msg = "Down";
repaint();
}
// Handle button released.
 public void mouseReleased(MouseEvent me)
{
```

```
// save coordinates
mouseX = me.getX();
mouseY = me.getY();
msg = "Up";
repaint();
}
 // Handle mouse dragged.
 public void mouseDragged(MouseEvent me)
{
// save coordinates
mouseX = me.getX();
mouseY = me.getY();
msg = "*";
showStatus("Dragging mouse at " + mouseX + ", " + mouseY);
repaint();
}
 // Handle mouse moved.
 public void mouseMoved(MouseEvent me)
{
// show status
showStatus("Moving mouse at " + me.getX() + ", " + me.getY());
}
 // Display msg in applet window at current X,Y location.
 public void paint(Graphics g)
{
g.drawString(msg, mouseX, mouseY);
}
}
```

OUTPUT


Applet Viewer: mouseevents
Applet
Mouse exited.
Moving mouse at 165, 240

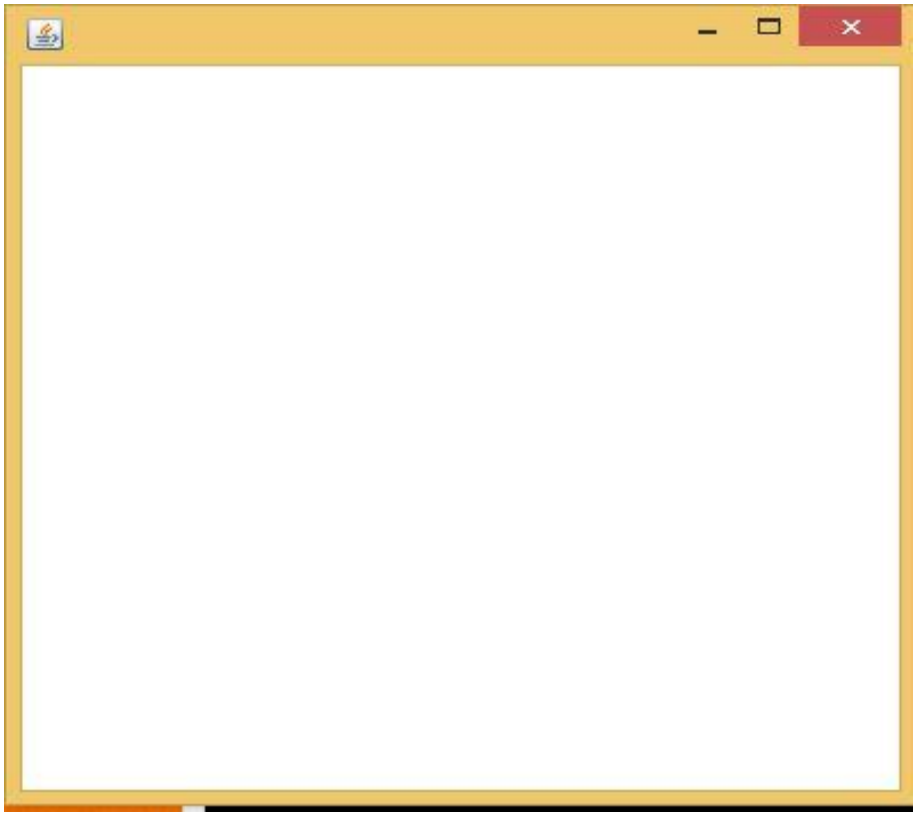

Applet Viewer: mouseevents
Applet
Mouse entered.
Moving mouse at 235, 149

windowevents.java

```
/*Q windows events*/
import java.awt.*;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
public class windowevents extends Frame implements WindowListener
{
public windowevents()
{
addWindowListener(this);
setSize(400,400);
setLayout(null);
setVisible(true);
}
public static void main(String[] args)
{
new windowevents();
}
public void windowActivated(WindowEvent arg0)
{
System.out.println("activated");
}
public void windowClosed(WindowEvent arg0)
{
System.out.println("closed");
}
public void windowClosing(WindowEvent arg0)
{
System.out.println("closing");
```

```
dispose();
}
public void windowDeactivated(WindowEvent arg0)
{
System.out.println("deactivated");
}
public void windowDeiconified(WindowEvent arg0)
{
System.out.println("deiconified");
}
public void windowIconified(WindowEvent arg0)
{
System.out.println("iconified");
}
public void windowOpened(WindowEvent arg0)
{
System.out.println("opened");
}
}
```

OUTPUT



```
activated
opened
deactivated
activated
deactivated
activated
deactivated
activated
deactivated
activated
closing
deactivated
closed
```

42) Develop a program to handle Key events.

Aim:

To develop a program to handle key events

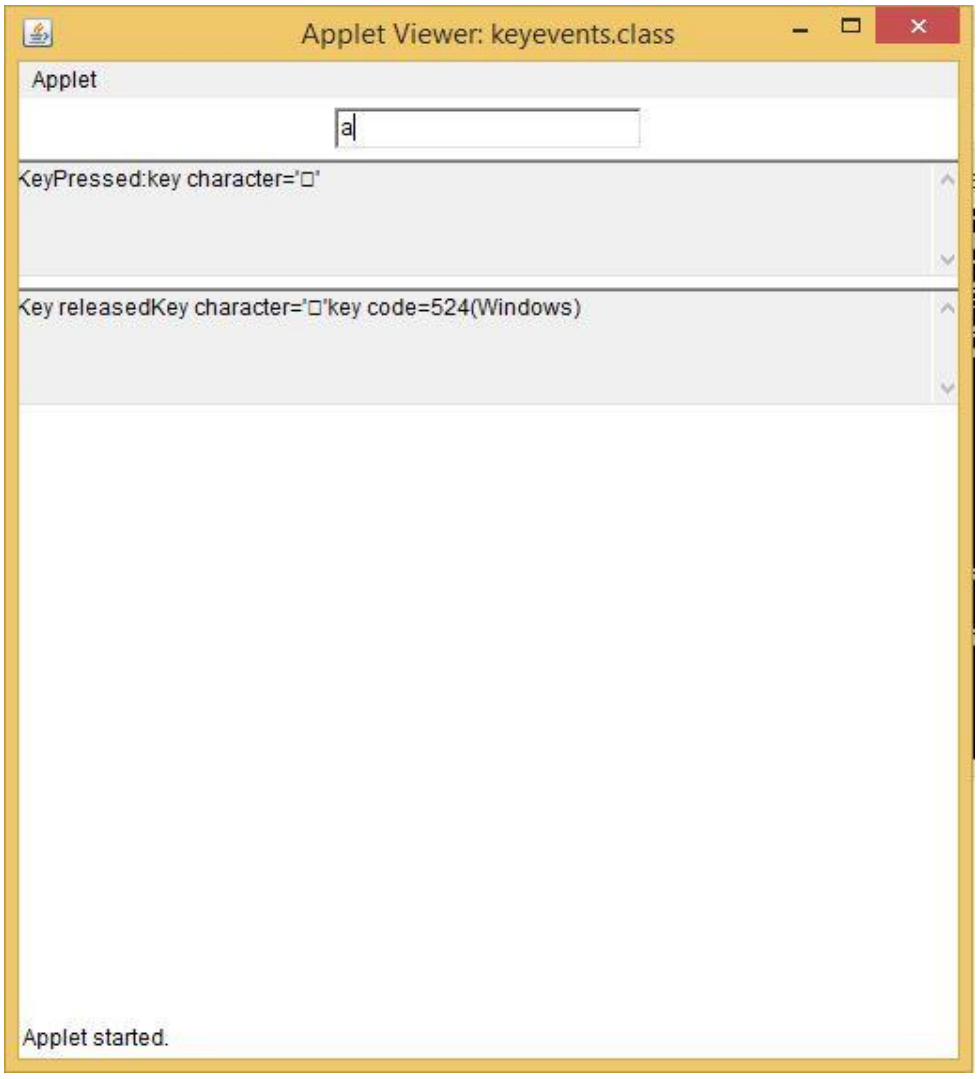
Source code:

keyevents.java

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
/* <APPLET CODE ="keyevents.class" WIDTH=500 HEIGHT=500>
</APPLET> */
public class keyevents extends Applet implements KeyListener
{
    TextArea tpress,trel;
    TextField t;
    public void init()
      {
          t=new TextField(20);
          t.addKeyListener(this);
          tpress=new TextArea(3,70);
          tpress.setEditable(false);
          trel=new TextArea(3,70);
          trel.setEditable(false);
          add(t);
          add(tpress);
          add(trel);
      }
    public void keyTyped(KeyEvent e)
          {
        disppress(e,"Key Typed:");
        }
```

```
        public void keyPressed(KeyEvent e)
                {
                disppress(e,"KeyPressed:");
                }
        public void keyReleased(KeyEvent e)
                {
                String charString,keyCodeString;
                char c=e.getKeyChar();
                int keyCode=e.getKeyCode();
                charString="Key character='"+c+"'";
keyCodeString="keycode="+keyCode+"("+KeyEvent.getKeyText(key
Code)+")";
  trel.setText("Key released"+charString+keyCodeString);
  }
   protected void disppress(KeyEvent e,String s)
                {
                String charString,keyCodeString,tmpString;
                char c=e.getKeyChar();
                int keyCode=e.getKeyCode();
                if(Character.isISOControl(c))
                    {
charString="key character=(an unprintable control character)";
                    }
                 else
                    {
                    charString="key character='"+c+"'";
                    }
  keyCodeString="keycode="+keyCode+"("+KeyEvent.getKeyText(key
  Code)+")";
  tpress.setText(s+charString);} }
```

OUTPUT


Applet Viewer: keyevents.class
Applet
a
KeyPressed:key character='□'
Key releasedKey character='□'key code=524(Windows)
Applet started.

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

Aim:

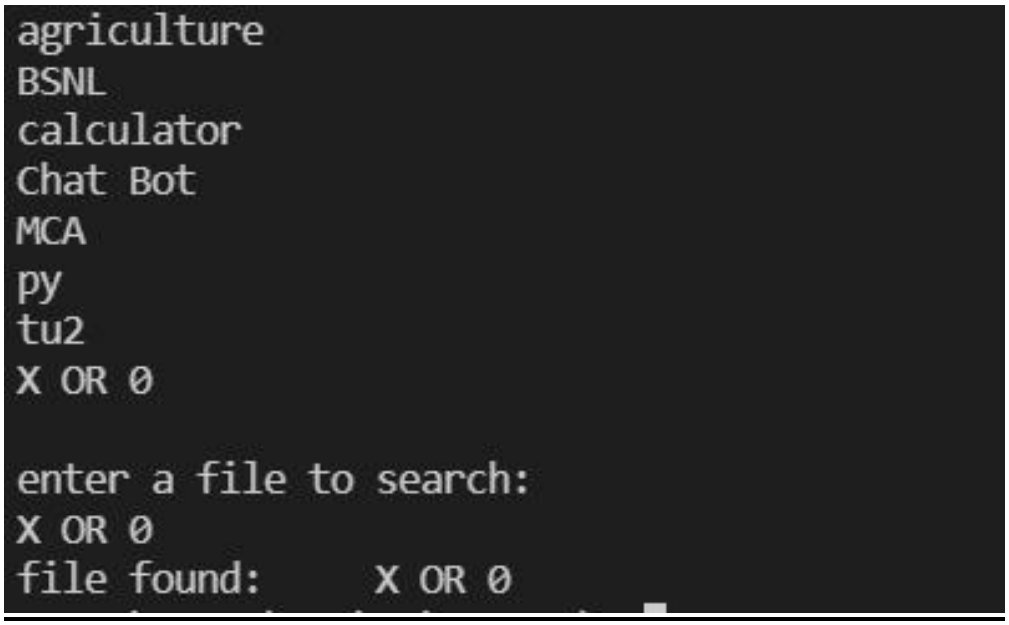
To write a program to list the sub directories and files in a given directory and also search for a file name.

Source code:

dirfile.java

```
import java.io.*;
import java.lang.*;
import java.io.File;
class dirfile
{
public static void main(String[] args) throws IOException
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
// creates a file object
File file = new File("C:\\java");
// returns an array of all files
String[] fileList = file.list();
for(String str : fileList)
{
System.out.println(str);
}
System.out.println("\n enter a file to search:");
String f = br.readLine();
for(String str : fileList)
{
if(f.equals(str))
System.out.println("File found:"+str);
}}}
```

OUTPUT


agriculture
BSNL
calculator
Chat Bot
MCA
py
tu2
X OR 0
enter a file to search:
X OR 0
file found: X OR 0

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

Aim:

To develop a program to write a file and then read from the file and display the contents on the console.

Source code:

file1.java

```
//writing and reading a file
import java.io.*;
class file1
{
public static void main(String []a)
{
byte city[]={'D','E','L','H','I','\n','M','U','M','B','A','I','\n'};
FileOutputStream outfile=null;   //writing to file
try
{
outfile= new FileOutputStream("city.txt");
outfile.write(city);
outfile.close();
}
catch(IOException e)
{
System.out.println(e);
}
FileInputStream infile=null;   //reading from file
int b;
try
{
infile= new FileInputStream("city.txt");
```

```
while((b=infile.read())!= -1)
{
System.out.println((char)b);
}
infile.close();
}
catch(IOException e1)
{
System.out.println(e1);
}
}
}
```

OUTPUT

45) Write a program to copy one file to another.

Aim:

To write a program to copy one file to another.

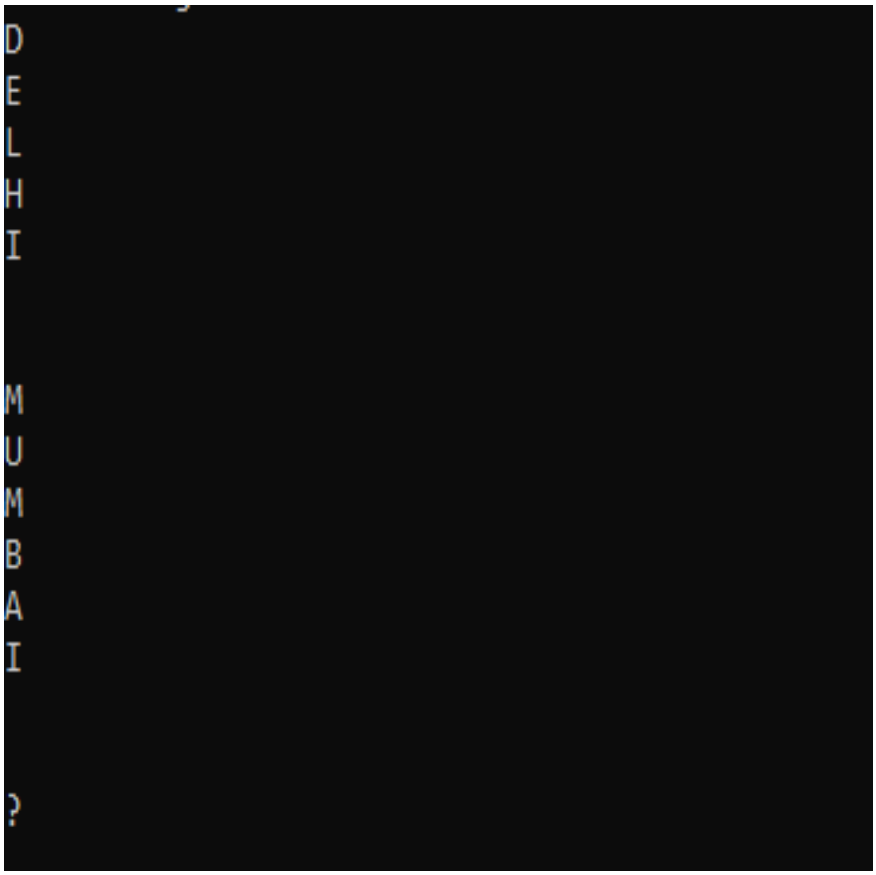
Source code:

file2.java

```
import java.io.*;
class file2
{
public static void main(String []a)
{
FileInputStream infile=null;
FileOutputStream outfile=null;
byte b;
try
{
infile = new FileInputStream("city.txt");
outfile= new FileOutputStream("citynew.txt");
do
{
b=(byte) infile.read();
outfile.write(b);
System.out.println((char)b);
}while(b != -1);
}
catch(IOException e)
{
System.out.println(e);
//System.exit(-1);
}
```

```
finally
{
try
{
outfile.close();
infile.close();
}
catch(IOException e1)
{
System.out.println(e1);
}
}
}
}
```

OUTPUT



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

Aim:

To write a program that reads from a file having integers and copy even numbers and odd numbers and separate to another file.

Source code:

file3.java

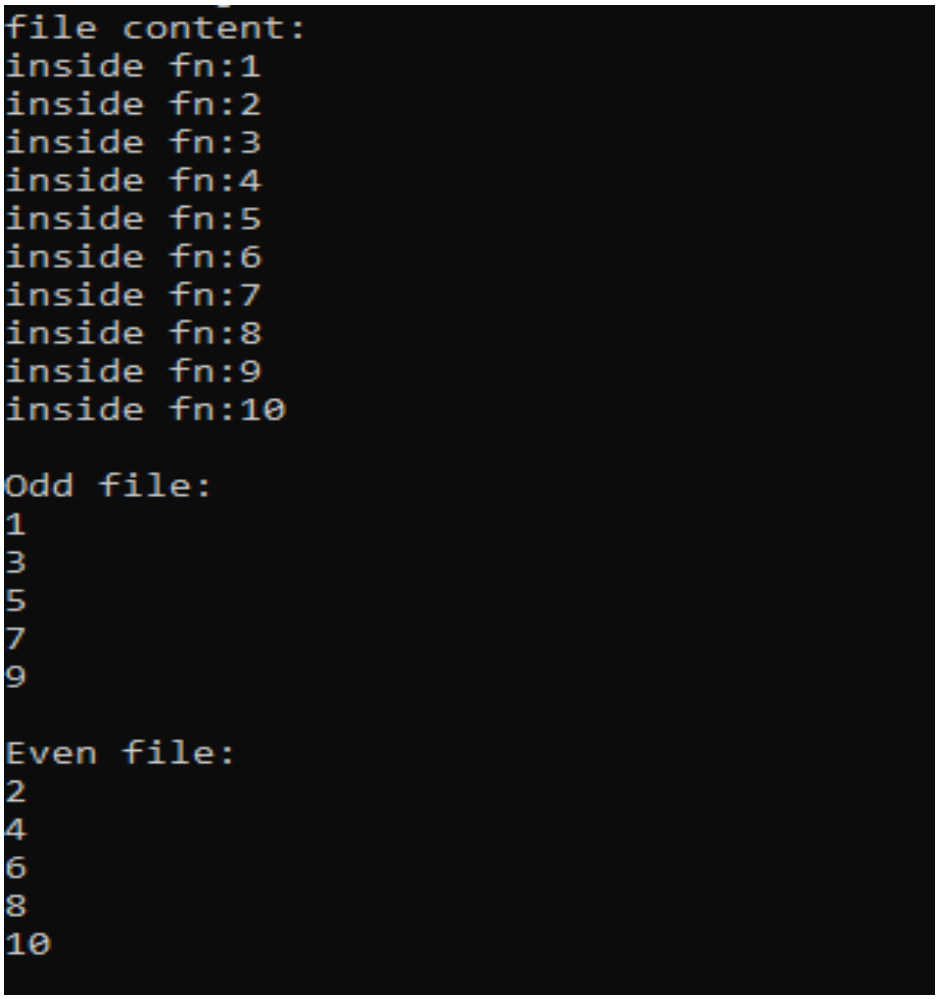
```
import java.io.*;
class file3
{
public static void main(String []a) throws IOException
{
int i;
File num= new File("integers.txt");    //creating main file
FileOutputStream fos=new FileOutputStream(num);
DataOutputStream dos= new DataOutputStream(fos); //for handling primitive data types
try
{
for(i=1;i<=10;i++)
dos.writeInt(i);
}
catch(IOException e)
{System.out.println("from 1"+e);}
dos.close();
fos.close();
FileInputStream fis=new FileInputStream(num);
DataInputStream dis= new DataInputStream(fis);
File num1= new File("odd.txt");
```

```
//creating odd number file
FileOutputStream fos1=new FileOutputStream(num1);
DataOutputStream dos1= new DataOutputStream(fos1);
File num2= new File("even.txt");        //creating even number file
FileOutputStream fos2=new FileOutputStream(num2);
DataOutputStream dos2= new DataOutputStream(fos2);
try
{
System.out.println("file content:");
for(int j=1;j<=10;j++)
{
i=dis.readInt();
System.out.println("inside fn:"+i);
if(i%2==0)
dos2.writeInt(i);
else
dos1.writeInt(i);
}
}
catch(IOException e1)
{System.out.println("from 2"+e1);}
dos1.close();
fos1.close();
dos2.close();
fos2.close();
dis.close();
fis.close();
```

```
FileInputStream fis1=new FileInputStream(num1);
DataInputStream dis1= new DataInputStream(fis1);
System.out.println("\nOdd file: ");
try
{
for(int j=1;j<=5;j++)
{
i=dis1.readInt();
System.out.println(i + " ");
}
}
catch(IOException e2)
{System.out.println("from 3"+e2);}
fis1.close();
dis1.close();
FileInputStream fis3=new FileInputStream(num2);
DataInputStream dis3= new DataInputStream(fis3);
System.out.println("\nEven file: ");
try
{
for(int k=1;k<=5;k++)
{
i=dis3.readInt();
System.out.println(i + " ");
}
}
catch(IOException e2)
```

```
{System.out.println("from 4"+e2);}
fis1.close();
dis1.close();
}
}
```

OUTPUT



47) Client server communication using Socket – TCP/IP

Aim:

To write a program for client server communication.

Source code:

Tcpclient.java

```
import java.io.*;

import java.net.*;

class Tcpclient

{

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

{

String s,ms;

BufferedReader infromuser = new BufferedReader(new
InputStreamReader(System.in));

System.out.println("hai");

Socket clientsocket = new Socket("127.0.0.1",5000);

DataOutputStreamouttoserver=newDataOutputStream(clientsocket.get
OuputStream());

System.out.println("\n enter a string:");

s=infromuser.readLine();

outtoserver.writeBytes(s+'\n');

BufferedReader infromserver = new BufferedReader(new
InputStreamReader(clientsocket.getInputStream()));

ms=infromserver.readLine();

System.out.println("From server:"+ms);

clientsocket.close();

}

}
```

Tcpserver.java

```
import java.io.*;

import java.net.*;
```

```
class Tcpserver
{
public static void main(String []args) throws Exception
{
String s1,ms1;
ServerSocket serversocket = new ServerSocket(5000);
while(true)
{
Socket clientsocket = serversocket.accept();
BufferedReaderinfromclient=new
BufferedReader(newInputStreamReder(clientsocket.getInputStream()));
s1=infromclient.readLine();
ms1=s1.toUpperCase()+'\n';
DataOutputStream outtoclient= new
DataOutputStream(clientsocket.getOutputStream());
outtoclient.writeBytes(ms1);
}
}
}
```

OUTPUT

48) Client Server communication using DatagramSocket – UDP

Aim:

To write a program for client server communication using datagram socket.

Source code:

udpclient.java

```
import java.io.*;

import java.net.*;

class udpclient

{

public static void main(String a[]) throws IOException

{

BufferedReader infromuser = new BufferedReader (new
InputStreamReader(System.in));

DatagramSocket clientsocket = new DatagramSocket();

InetAddress ipaddress = InetAddress.getByName("127.0.0.1");

byte[] receivedata =new byte[1024];

byte[] senddata =new byte[1024];

System.out.println("\n enter a string:");

String str= infromuser.readLine();

senddata= str.getBytes();

DatagramPacket sendpacket = new DatagramPacket
(senddata,senddata.length,ipaddress,5000);

clientsocket.send(sendpacket);

DatagramPacket receivepacket = new DatagramPacket
(receivedata,receivedata.length);

clientsocket.receive(receivepacket);

String modified = new String(receivepacket.getData());

System.out.println("from Server"+modified);

clientsocket.close();

}

}
```

udpserver.java

```
import java.io.*;
import java.net.*;
class udpserver
{
public static void main(String a[]) throws IOException
{
DatagramSocket  serversocket=new DatagramSocket(5000);
byte[] receivedata =new byte[1024];
byte[] senddata =new byte[1024];
while(true)
{
  DatagramPacket receivepacket=new
  DatagramPacket(receivedata,receivedata.length);
serversocket.receive(receivepacket);
String sentence=new String(receivepacket.getData());
InetAddress ipaddress =receivepacket.getAddress();
int port=receivepacket.getPort();
String capital=sentence.toUpperCase();
System.out.println(capital);
senddata=capital.getBytes();
DatagramPacket sendpacket=new
DatagramPacket(senddata,senddata.length,ipaddress,port);
serversocket.send(sendpacket);
}
}
}
```

OUTPUT