20MCA132 – OBJECT ORIENTED PROGRAMMING

Submitted in partial fulfilment of the requirements for the award of

Masters of Computer Applications

At

COLLEGE OF ENGINEERING POONJAR

Managed by I.H.R.D., A Govt. of Kerala undertaking

(Affiliated to APJ Abdul Kalam Technological University)

SUBMITTED BY

SANITHA K.S (PJR24MCA-2016)

Department of Computer Science

COLLEGE OF ENGINEERING POONJAR

COLLEGE OF ENGINEERING POONJAR

Managed by I.H.R.D., A Govt. of Kerala undertaking

(Affiliated to APJ Abdul Kalam Technological University)

CERTIFICATE

Certified that this is a bonafide record of practical work done in Object Oriented programming Lab (20MCA132) by **SANITHA K.S**Reg No. **PJR24MCA-2016** of College of Engineering, Poonjar during the academic year 2024 - 2025.

Dr. Annie Julie Joseph
Head of the Department

Geethu A
Staff member in Charge

Submitted to the University Examination held on:

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

PROGRAM - 1

AIM

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

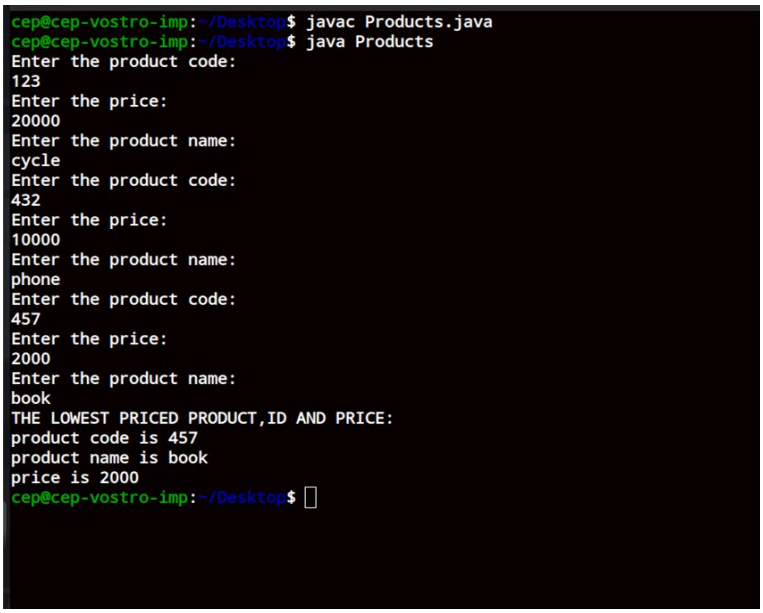
SOURCE CODE

```
import java.util.Scanner;
public class
Products{ int
pcode,price;
String pname;
Products()
{
Scanner n1=new Scanner(System.in);
System.out.println("Enter the product code:");
pcode=n1.nextInt();
System.out.println("Enter the price:");
price=n1.nextInt();
System.out.println("Enter the product name:");
pname=n1.next();
}

public static void main(String
args[]){ Products p1=new Products();
Products p2=new Products();
Products p3=new Products();
System.out.println("THE LOWEST PRICED PRODUCT,ID AND PRICE:");
if(p1.price<p2.price && p1.price<p3.price)
{
System.out.println("product code is "+p1.pcode);
System.out.println("product name is "+p1.pname);
System.out.println("price is "+p1.price);
}
```

```
else if(p2.price<p1.price && p2.price<p3.price)
{
System.out.println("product code is "+p2.pcode);
System.out.println("product name is "+p2.pname);
System.out.println("price is "+p2.price);
}
else
{
System.out.println("product code is "+p3.pcode);
System.out.println("product name is "+p3.pname);
System.out.println("price is "+p3.price);
}
}
}
```

OUTPUT



RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 2

AIM

Read two matrix from the console and perform matrix addition.

SOURCE CODE

```
import java.util.Scanner;
public class Matrices{
public static void main(String args[])
{
int p,q,m,n;
Scanner s=new Scanner(System.in);
System.out.println("Enter number of rows in first matrix:");
p=s.nextInt();
System.out.println("Enter number of column in first matrix:");
q=s.nextInt();
System.out.println("Enter number of rows in second matrix:");
m=s.nextInt();
System.out.println("Enter number of column in second matrix:");
n=s.nextInt();
if(p==m && q==n)
{
int a[][]=new int[p][q];
int b[][]=new int[m][n];
int c[][]=new int[m][n];
System.out.println("Enter all the elements of first matrix:");
for(int i=0;i<p;i++)
{
for(int j=0;j<q;j++)
{
a[i][j]=s.nextInt();

}
```

```
}
System.out.println("Enter all the elements of second matrix:");
for(int i=0;i<m;i++)
{
for(int j=0;j<n;j++)
{
b[i][j]=s.nextInt();
}
}
System.out.println("First Matrix:");
for(int i=0;i<p;i++)
{
for(int j=0;j<q;j++)
{
System.out.print(a[i][j]+" ");
}
System.out.println();
}
System.out.println("Second Matrix:");
for(int i=0;i<m;i++)
{
for(int j=0;j<n;j++)
{
System.out.print(b[i][j]+" ");
}
System.out.println();
}
for(int i=0;i<p;i++)
{
for(int j=0;j<n;j++)
{
for(int k=0;k<q;k++)
{
c[i][j]=a[i][j]+b[i][j];
}
```

```
}
}
System.out.println("Matrix after addition:");
for(int i=0;i<p;i++)
{
for(int j=0;j<n;j++)
{
System.out.print(c[i][j]+" ");
}
System.out.println();
}
}
else
{
System.out.println("Addition would not be possible");
}
}
}
```

OUTPUT

```
cep@cep-vostro-imp:~/Desktop$ java Matrices
Enter number of rows in first matrix:
3
Enter number of column in first matrix:
2
Enter number of rows in second matrix:
3
Enter number of column in second matrix:
2
Enter all the elements of first matrix:
2 3
4 6
7 4
Enter all the elements of second matrix:
1 2
3 4
7 8
First Matrix:
2 3
4 6
7 4
Second Matrix:
1 2
3 4
7 8
Matrix after addition:
3 5
7 10
14 12
```

RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 3

AIM

Add complex numbers

SOURCE CODE

```
import java.util.Scanner;
public class Complex
{
double real,imaginary;
Complex(){}
Complex(double real,double imaginary)
{
this.real=real;
this.imaginary=imaginary;
}
public static Complex sum(Complex a,Complex b)
{
Complex ans=new Complex();
ans.real=a.real+b.real;
ans.imaginary=a.imaginary+b.imaginary;
return ans;
}
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
Complex num1=new Complex();
Complex num2=new Complex();
System.out.print("Enter first complex number: \nreal:");
num1.real=sc.nextDouble();
System.out.print("imaginary:");
num1.imaginary=sc.nextDouble();
System.out.print("Enter second complex number: \nreal:");
```

```
num2.real=sc.nextDouble();
System.out.print("imaginary:");
num2.imaginary=sc.nextDouble();
Complex answer=sum(num1,num2);
System.out.print("The additionof two complex number
is:"+answer.real+"+"+answer.imaginary+"i");
}
}
```

OUTPUT

```
Enter first complex number:
real:34
imaginary:9.8
Enter second complex number:
real:56
imaginary:2.9
The additionof two complex number is:90.0+12.700000000000001i
PS D:\New folder (2)>
```

RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 4

AIM

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

SOURCE CODE

```
public class
CPU{ int price;
CPU(int
p){ price=p;
}
void display(){
System.out.println("CPU Price="+price);
}
class
Processor{ int
cores;
String manufracturer;
Processor(int noc,String
manuf){ cores=noc;
manufracturer=manuf;
}
void
display(){ System.out.println("Processor
info"); System.out.println("No of
cores="+cores);
System.out.println("Mnanufracteurer="+manufracturer);
}
}
protected class RAM
{
```

int memory;

String manufracturer;

```
RAM(int mem,String manufac)
{
memory=mem;
manufracturer=manufac;
}
void display()
{
System.out.println("RAM Info");
System.out.println("memory="+memory+"GB");
System.out.println("manufacturer="+manufracturer);
}
}
public static void main(String[]args)
{
CPU cpu=new CPU(5000);
CPU.Processor pro=cpu.new Processor(8, "Intel");
CPU.RAM ram=cpu.new RAM(8,"microsoft");
cpu.display();
ram.display();
pro.display();
}
}
```

OUTPUT

```
cep@cep-vostro-imp:~/Downloads$ gedit CPU.java
cep@cep-vostro-imp:~/Downloads$ javac CPU.java
cep@cep-vostro-imp:~/Downloads$ java CPU
CPU Price=5000
RAM Info
memory=8GB
manufacturer=microsoft
Processor info
No  of cores=8
Mnanufracteurer=Intel
cep@cep-vostro-imp:~/Downloads$
```

RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 5

AIM

Program to Sort strings

SOURCE CODE

```
import java.util.Scanner;
import java.util.Arrays;
public class Sort {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the number of words:");
int num = sc.nextInt();
String[] word = new String[num];
for (int i = 0; i < num; i++)
{
System.out.println("Enter a word:");
word[i] = sc.next();
}
for (int i = 0; i < num - 1; i++)
{
for (int j = 0; j < num - 1 - i; j++)
{
if (word[j].compareTo(word[j + 1]) > 0)
{
String temp = word[j];
word[j] = word[j + 1];
```

```
word[j + 1] = temp;

}

}

}

System.out.println("Sorted Strings: " + Arrays.toString(word)).

}

}
```

OUTPUT

```
Enter the number of words:
2
Enter a word:
hello
Enter a word:
world
Sorted Strings: [hello, world]
PS D:\New folder (2)>
```

RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 6

AIM

Perform string manipulations.

SOURCE CODE

```
import java.util.Scanner;
public class
String_manipulate{ public static void
main(String args[])
{
Scanner sc=new Scanner(System.in);
System.out.print("Enter the string:");
String str1=sc.nextLine();
System.out.println("Length of string="+str1.length());
System.out.println("Uppercase="+str1.toUpperCase());
System.out.println("Lowercase="+str1.toLowerCase());
System.out.println("String ends with ="+str1.endsWith("e"));
System.out.println("Character at first position="+str1.charAt(0));
System.out.println("Replace 'Jesus'with 'Mary'="+str1.replaceAll("Jesus","Mary"));
System.out.println("String index is="+str1.indexOf("J"));
}
}
```

OUTPUT

```
Enter the string:hello world
Length of string=11
Uppercase=HELLO WORLD
Lowercase=hello world
String ends with =false
Character at first position=h
Replace 'Jesus'with 'Mary'=hello world
String index is=-1
PS D:\New folder (2)>
```

RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 7

AIM

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

SOURCE CODE

```
import java.util.Scanner;
public class Employee
{
int eNo;
String eName;
double eSalary;
public void getdetails()
{
System.out.println("\nEnter the Employee details\n");
Scanner sc=new Scanner(System.in);
System.out.print("Employee number:");
eNo=sc.nextInt();
System.out.print("Name:");
sc.nextLine();
eName=sc.nextLine();
System.out.print("Salary:");
eSalary=sc.nextDouble();
}
void display()
{
System.out.println("Employee No:"+eNo);
System.out.println("Name:"+eName);
System.out.println("Salary Amount:"+eSalary+"\n");
}
public static void main(String[]args)
{
```

```
System.out.print("Enter the No.of Employee's:");
Scanner sc1=new Scanner(System.in);
int num=sc1.nextInt();
Employee arr[]=new Employee[num];
for(int i=0;i<num;i++)
{
arr[i]=new Employee();
arr[i].getdetails();
}
System.out.println("\nInformations of all the employee's:");
for(int i=0;i<num;i++)
{
arr[i].display();
}

System.out.println("\nEnter the Employee Number to get details of a employee:");
int num2=sc1.nextInt();
for(int i=0;i<num;i++)
{
if(arr[i].eNo==num2)
{
System.out.println("\nEmployee details");
arr[i].display();
}
}
}
}
```

OUTPUT

```
Employee No:101
Name:Rahul
Salary Amount:21000.0

Employee No:201
Name:Jino
Salary Amount:35000.0


Enter the Employee Number to get details of a employee:
101

Employee details
Employee No:101
Name:Rahul
Salary Amount:21000.0

PS D:\New folder (2)>
```

RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 8

AIM

Area of different shapes using overloaded functions.

SOURCE CODE

```
import java.util.Scanner;
public class AreaShape
{
void area(float s)
{
System.out.println("Area of square="+(s*s));
}
void area(int l,int b)
{
System.out.println("Area of rectangle is="+(l*b));
}
void area(int l,int b,int h)
{
double s=(l+b+h)/2;
double triArea;
triArea=Math.sqrt(s*(s-l)*(s-b)*(s-h));
System.out.println("Area of rectangle="+triArea);
}

public static void main(String[]args)
{

Scanner  sc=new Scanner(System.in);
AreaShape obj=new AreaShape();
System.out.println("enter the side of squre:");
float side=sc.nextFloat();
obj.area(side);
```

```
System.out.println("enter the length of rectangle:");
int length=sc.nextInt();
System.out.println("enter the width of rectangle:");
int width=sc.nextInt();
obj.area(length,width);
System.out.println("enter the  side of triangle:");
int side1=sc.nextInt();
int side2=sc.nextInt();
int side3=sc.nextInt();
obj.area(side1,side2,side3);
sc.close();
}
}
```

OUTPUT

```
enter the side of squre:
4
Area of square=16.0
enter the length of rectangle:
6
enter the width of rectangle:
8
Area of rectangle is=48
enter the side of triangle:
3
4
5
Area of rectangle=6.0
PS D:\New folder (2)>
```

RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 9

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.

SOURCE CODE

```
import java.util.Scanner;
class Person
{
String name,gender,address;
protected int age;
public Person()
{
}
public Person(String n,String g,String addr,int a)
{
this.name=n;
this.gender=g;
this.address=addr;
this.age=a;
}
public void displayPerson()
{
System.out.println();
System.out.println("Name:"+name);
System.out.println("Gender:"+gender);
System.out.println("Address:"+address);
System.out.println("Age:"+age);
```

```
}
}
class Employee extends Person
{
int empid,salary;
String companyname,qualification;
public Employee()
{
}
public Employee(String n,String g,String addr,int a,int eid,String cname,String qual,int sal)
{
super(n,g,addr,a);
empid=eid;
companyname=cname;
qualification=qual;
salary=sal;
}
public void displayEmployee()
{
super.displayPerson();
System.out.println("Empid:"+empid);
System.out.println("Company name:"+companyname);
System.out.println("Qualification:"+qualification);
System.out.println("Salary:"+salary);
}
}
class Teacher extends Employee
{
String subject,department;
int teacherid;
public Teacher(String n,String g,String addr,int a,int eid,String cname,String qual,int sal,String
sub,String dept,int tid)
{
super(n,g,addr,a,eid,cname,qual,sal);
subject=sub;
```

```
department=dept;
teacherid=tid;
}
public void displayTeacher()
{
super.displayEmployee();
System.out.println("Teacherid:"+teacherid);
System.out.println("Subject:"+subject);
System.out.println("Department:"+department);
System.out.println();
}
}
public class Inheritance
{
public static void main(String args[])
{
System.out.print("Enter number of teachers:");
Scanner sc=new Scanner(System.in);
int n=sc.nextInt();
for(int i=0;i<n;i++)
{
System.out.println();
System.out.print("Enter name of the teacher:");
String name=sc.next();
System.out.print("Enter gender of the teacher:");
String gen=sc.next();
System.out.print("Enter address of the teacher:");
String addr=sc.next();
System.out.print("Enter age of the teacher:");
int ag=sc.nextInt();
System.out.print("Enter Empid of the teacher:");
int eid=sc.nextInt();
System.out.print("Enter company name:");
String cn=sc.next();
System.out.print("Enter qualification of the teacher:");
```

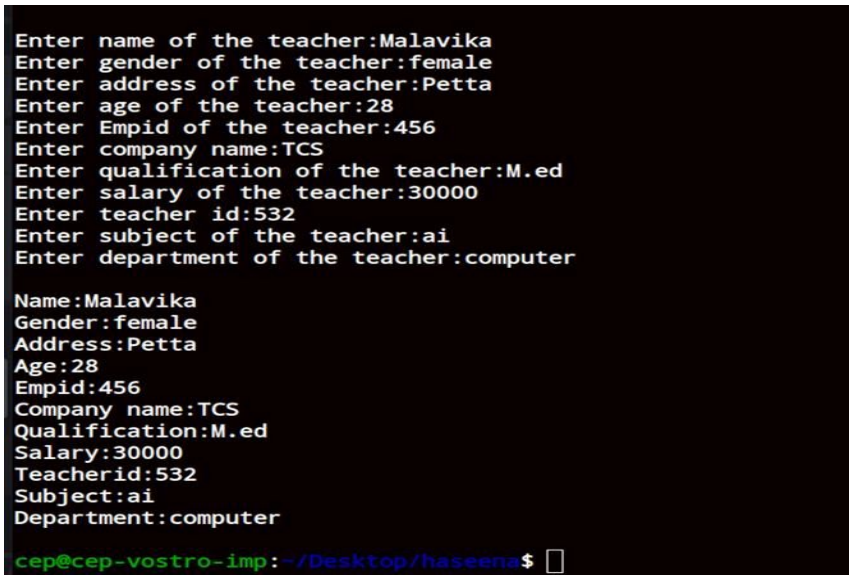
```
String qual=sc.next();
System.out.print("Enter salary of the teacher:");
int sal=sc.nextInt();
System.out.print("Enter teacher id:");
int tid=sc.nextInt();
System.out.print("Enter subject of the teacher:");
String sub=sc.next();
System.out.print("Enter department of the teacher:");
String dep=sc.next();
Teacher t=new Teacher(name,gen,addr,ag,eid,cn,qual,sal,sub,dep,tid);
t.displayTeacher();
}
}
}
```

OUTPUT

```
cep@cep-vostro-imp:~/Desktop/haseena$ gedit Inheritance.java
cep@cep-vostro-imp:~/Desktop/haseena$ java Inheritance
Enter number of teachers:2

Enter name of the teacher:Hari
Enter gender of the teacher:male
Enter address of the teacher:Pala
Enter age of the teacher:45
Enter Empid of the teacher:123
Enter company name:ICFOSS
Enter qualification of the teacher:Mtech
Enter salary of the teacher:45000
Enter teacher id:678
Enter subject of the teacher:python
Enter department of the teacher:computer

Name:Hari
Gender:male
Address:Pala
Age:45
Empid:123
Company name:ICFOSS
Qualification:Mtech
Salary:45000
Teacherid:678
Subject:python
Department:computer
```



RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 10

AIM

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

SOURCE CODE

```
import java.util.*;
interface shapes
{
void area();
void perimeter();
}
class circle implements shapes
{
 int r;
double pi=3.14,area,perimeter;

public circle()
{
Scanner inp1=new Scanner(System.in);
System.out.print("Entere radius:");
r=inp1.nextInt();
}
public void area()
{
area=pi*r*r;
System.out.print("Area of circle with radius" +r+ "is" +area);
}
public void perimeter()
{
perimeter=2*pi*r;
```

```
System.out.print("Perimeter of circle with radius" +r+ "is" +perimeter);
}
}
class rectangle implements shapes
{
int l,b;
int area,perimeter;
public rectangle()
{
Scanner inp2=new  Scanner(System.in);
System.out.print("Enter length of rectangle:");
l=inp2.nextInt();
System.out.print("Enter breadth of rectangle:");
b=inp2.nextInt();
}
public void
area(){ area=l*b;
System.out.print("Area of rectangle is:" +area);
}
public void perimeter()
{
perimeter=2*(l+b);
System.out.print("Perimeter of rectangle is:" +perimeter);
}
}

public class Interfacearea{
public static void main(String args[])
{
int ch1,ch2;
Scanner inp3=new Scanner(System.in);
do{
System.out.println("\nSelect a shape \n1.circle \n2.Rectangle \n3.exit\n");
System.out.print("Enter your choice:");
ch1=inp3.nextInt();
```

```
switch(ch1)
{
case 1:circle objc=new circle();
System.out.println("Find\n1.area\n2.Perimeter\n");
System.out.print("Enter your choice:");
ch2=inp3.nextInt();
switch(ch2)
{
case 1:objc.area();
break;
case 2:objc.perimeter();
break;
default:System.out.println("Invalid choice");
}
break;
case 2:rectangle objr=new rectangle();
System.out.println("Find\n1.Area\n2.Perimeter\n");
System.out.print("Enter your choice:");
ch2=inp3.nextInt();
switch(ch2)
{
case 1:objr.area();
break;
case 2:objr.perimeter();
break;
default:System.out.println("Invalid choice");
}
break;
default:System.out.println("Invalid choice");
case 3:return;
}
}
while(ch1!=3);
}}
```

OUTPUT

```
Select a shape
1.circle
2.Rectangle
3.exit

Enter your choice:1
Entere radius:5
Find
1.area
2.Perimeter

Enter your choice:1
Area of circle with radius5is78.5
Select a shape
1.circle
2.Rectangle
3.exit
```

```
Enter your choice:2
Enter length of rectangle:6
Enter breadth of rectangle:5
Find
1.Area
2.Perimeter

Enter your choice:2
Perimeter of rectangle is:22
Select a shape
1.circle
2.Rectangle
3.exit

Enter your choice:3
PS D:\New folder (2)>
```

RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 11

AIM

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

SOURCE CODE

```
import java.util.Scanner;
class Username extends
Exception{ public Username(String
msg)
{
super(msg);
}
}
class Password extends
Exception{ public Password(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(!username.equals("admin"))
throw new Username("username must be admin");
```

```
else if(!password.equals("admin"))
throw new Password("incorrect password\ntype correct passwoprd???");
else
System.out.println("login successfull");
}
catch(Username u){
u.printStackTrace();
}
catch(Password p){
p.printStackTrace();
}
finally
{
System.out.println("The final statement is executed");
}
}
}
```

OUTPUT

```
Enter username:admin
Enter password:user@123
Password: incorrect password
type correct passwoprd???
        at CheckLogin.main(CheckLogin.java:22)
The final statement is executed
```

```
Enter username:Anu
Enter password:admin
Username: username must be admin
        at CheckLogin.main(CheckLogin.java:19)
The final statement is executed
```

```
Enter username:admin
Enter password:admin
login successfull
The final statement is executed
PS D:\New folder (2)>
```

RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 12

AIM

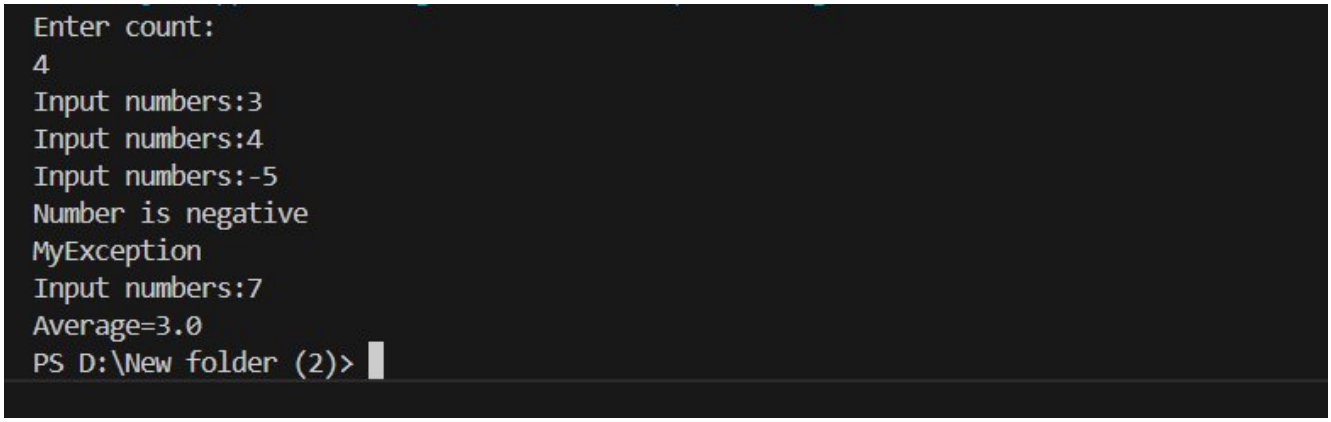
Find the average of N positive integers, raising a user defined exception for each negative input

SOURCE CODE

```
import java.io.BufferedReader;
import java.util.Scanner;
import java.io.IOException;
import java.io.InputStreamReader;
class MyException extends Exception
{
public MyException(String str)
{
System.out.println(str);
}
}
public class Sign
{
public static void main(String args[])throws IOException
{
int sum=0;
float avg;
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
Scanner sc=new Scanner(System.in);
System.out.println("Enter count:");
int n=sc.nextInt();
for(int i=0;i<n;i++)
{
System.out.print("Input numbers:");
try
{
int num=Integer.parseInt(br.readLine());
```

```
if(num<0)
throw new MyException("Number is negative");
else
sum=sum+num;
//throw new MyException("Number is positive");
}
catch(MyException m)
{
System.out.println(m);
}
}
avg=sum/n;
System.out.println("Average="+avg);
}
}
```

OUTPUT



RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 13

AIM

Program to find maximum of three numbers using AWT.

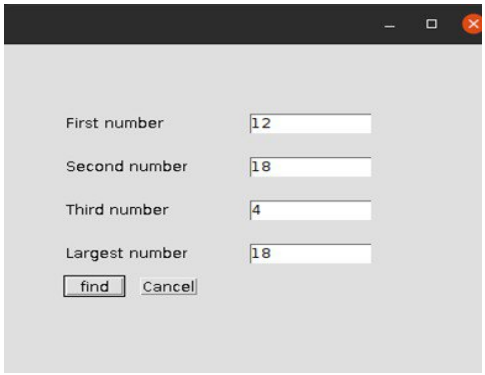
SOURCE CODE

```
import java.awt.*;
import java.awt.event.*;
public class Awtbiggest implements ActionListener
{
Frame f=new Frame();
Label l1=new Label("First number");
Label l2=new Label("Second number");
Label l3=new Label("Third number");
Label l4=new Label("Largest number");
TextField t1=new TextField();
TextField t2=new TextField();
TextField t3=new TextField();
TextField t4=new TextField();
Button b1=new Button("find");
Button b2=new Button("Cancel");
public Awtbiggest()
{
l1.setBounds(50,100,100,20);
l2.setBounds(50,140,100,20);
l3.setBounds(50,180,100,20);
l4.setBounds(50,220,100,20);
t1.setBounds(200,100,100,20);
t2.setBounds(200,140,100,20);
t3.setBounds(200,180,100,20);
t4.setBounds(200,220,100,20);
b1.setBounds(50,250,50,20);
b2.setBounds(110,250,50,20);
```

```
f.add(l1);
f.add(l2);
f.add(l3);
f.add(l4);
f.add(t1);
f.add(t2);
f.add(t3);
f.add(t4);
f.add(b1);
f.add(b2);
b1.addActionListener(this);
b2.addActionListener(this);
f.setLayout(null);
f.setVisible(true);
f.setSize(400,350);
}
public void actionPerformed(ActionEvent e)
{
int l;
int  n1=Integer.parseInt(t1.getText());
int  n2=Integer.parseInt(t2.getText());
int  n3=Integer.parseInt(t3.getText());
if(e.getSource()==b1)
{
if(n1>n2 && n1>n3)
l=n1;
else if(n2>n3 && n2>n1)
l=n2;
else
l=n3;
t4.setText(String.valueOf(l));
}
if(e.getSource()==b2)
{
System.exit(0);
```

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

OUTPUT



RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 14

AIM

Implement a simple calculator using AWT components.

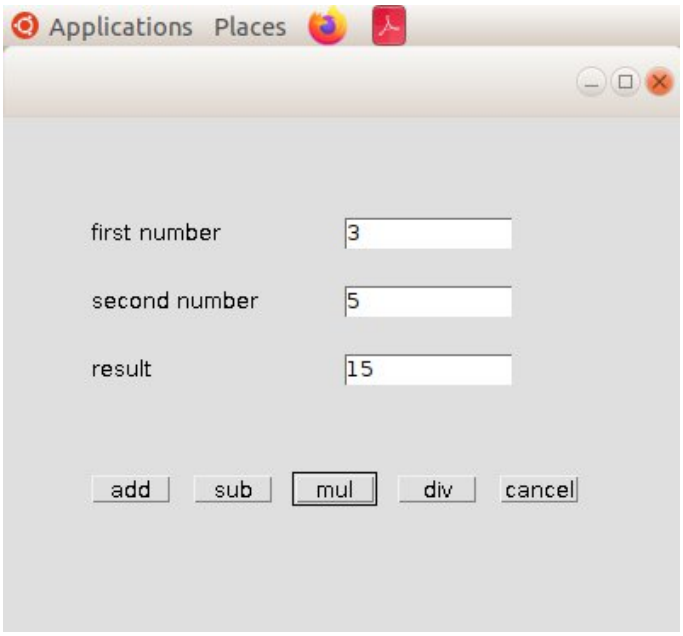
SOURCE CODE

```
import java.awt.*;
import java.awt.event.*;
public class Awtcal implements ActionListener
{
Frame f=new Frame();
Label l1=new Label("First Number :");
Label l2=new Label("Second Number :");
Label l3=new Label("Result");
TextField t1=new TextField();
TextField t2=new TextField();
TextField t3=new TextField();
Button b1=new Button("Add");
Button b2=new Button("Sub");
Button b3=new Button("Mul");
Button b4=new Button("Div");
Button b5=new Button("Cancel");

public Awtcal()
{
l1.setBounds(50,100,100,20);
l2.setBounds(50,140,100,20);
l3.setBounds(50,180,100,20);
t1.setBounds(200,100,100,20);
t2.setBounds(200,140,100,20);
t3.setBounds(200,180,100,20);
b1.setBounds(50,250,50,20);
b2.setBounds(110,250,50,20);
b3.setBounds(170,250,50,20);
b4.setBounds(230,250,50,20);
b5.setBounds(290,250,50,20);
f.add(l1);
f.add(l2);
f.add(l3);
f.add(t1);
f.add(t2);
f.add(t3);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(b4);
```

```
f.add(b5);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
f.setLayout(null);
f.setVisible(true);
f.setSize(400,350);
}
public void actionPerformed(ActionEvent e)
{
int  n1=Integer.parseInt(t1.getText());
int  n2=Integer.parseInt(t2.getText());
if(e.getSource()==b1)
{
t3.setText(String.valueOf(n1+n2));
}
if(e.getSource()==b2)
{
t3.setText(String.valueOf(n1-n2));
}
if(e.getSource()==b3)
{
t3.setText(String.valueOf(n1*n2));
}
if(e.getSource()==b4)
{
t3.setText(String.valueOf(n1/n2));
}
if(e.getSource()==b5)
{
System.exit(0);
}
}
public static void main(String args[])
{
new Awtcal();
}
}
```

OUTPUT



RESULT

The program has been executed successfully and output is obtained

PROGRAM - 15

AIM

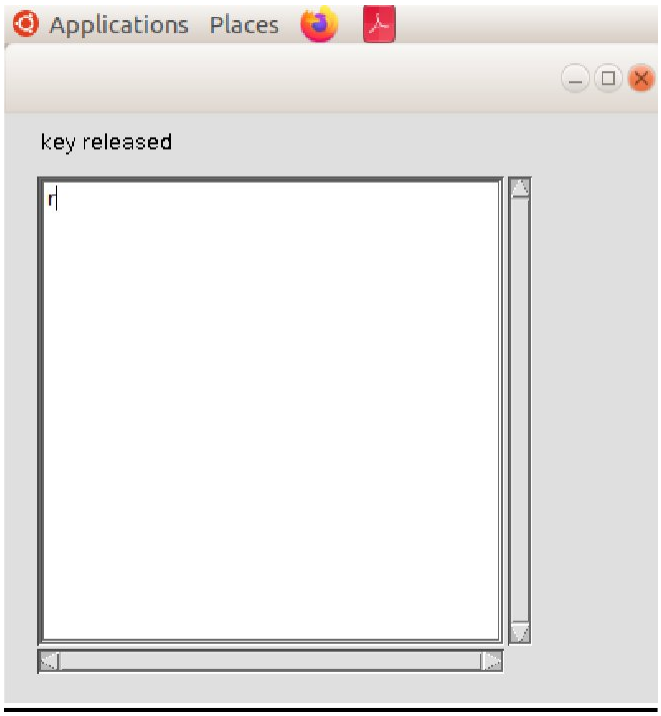
Develop a program to handle Key events.

SOURCE CODE

```
import java.awt.*;
import java.awt.event.*;
public class Keyevents extends Frame implements KeyListener
{
Label l;
TextArea area;
public Keyevents() throws HeadlessException
{
l=new Label();
l.setBounds(20,50,100,20);
area=new TextArea();
area.setBounds(20,80,300,300);
area.addKeyListener(this);
add(l);
add(area);
setSize(400,400);
setLayout(null);
setVisible(true);
}
public void keyPressed(KeyEvent e)
{
l.setText("key pressed");
}
public void keyReleased(KeyEvent e)
{
l.setText("key released");
}
public void keyTyped(KeyEvent e)
{
l.setText("key typed");
}
public static void main(String args[])
{
new Keyevents();

}
}
```

OUTPUT



RESULT

The program has been executed successfully and output is obtained

PROGRAM - 16

AIM

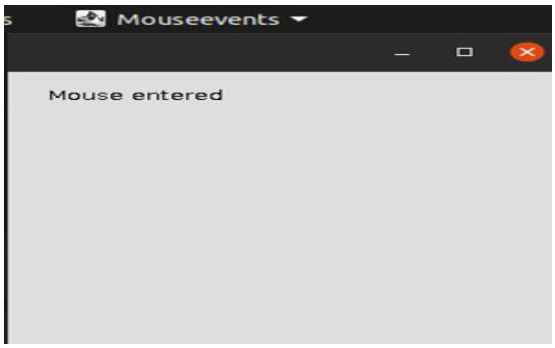
Develop a program to handle all mouse events.

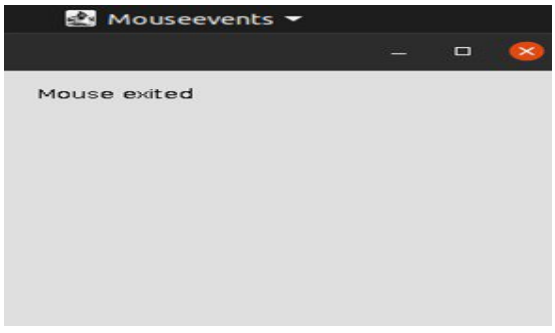
SOURCE CODE

```
import java.awt.event.*;
import java.awt.*;
public class Mouseevents extends Frame implements
MouseListener{ Label l;
public Mouseevents() throws HeadlessException
{
addMouseListener(this);
l=new Label();
l.setBounds(20,50,100,20);
add(l);
setSize(300,300);
setLayout(null);
setVisible(true);
}
public void mouseClicked(MouseEvent e)
{
l.setText("Mouse clicked");
}
public void mouseEntered(MouseEvent e)
{
l.setText("Mouse entered");
}
public void mouseExited(MouseEvent e)
{
l.setText("Mouse exited");
}
public void mousePressed(MouseEvent e)
```

```
{
l.setText("Mouse presssed");
}
public void mouseReleased(MouseEvent e)
{
l.setText("Mouse released");
}
public static void main(String args[])
{
new Mouseevents();
}
}
```

OUTPUT





RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 17

AIM

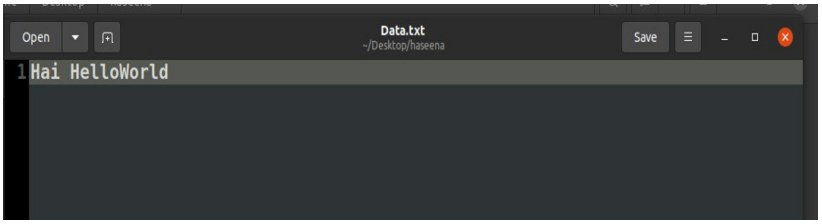
Write a program to write to a file, then read from the file and display the contents on the Console.

SOURCE CODE

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
public class Filewrt{
public static void main(String args[])throws IOException
{
try
{
FileWriter dataWriter=new FileWriter("Data.txt");
dataWriter.write("Hai Hello");
dataWriter.write("World");
dataWriter.close();
}
catch(IOException ex)
{
System.err.println("An error occured");
ex.printStackTrace();
}
try
{
File dataFile=new File("Data.txt");
Scanner dataRead=new Scanner(dataFile);
while(dataRead.hasNextLine())
{
System.out.println(dataRead.nextLine());
```

```
}
dataRead.close();
}
catch(FileNotFoundException ex)
{
System.out.println("An error occured");
ex.printStackTrace();
}
}
}
```

OUTPUT



RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 18

AIM

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

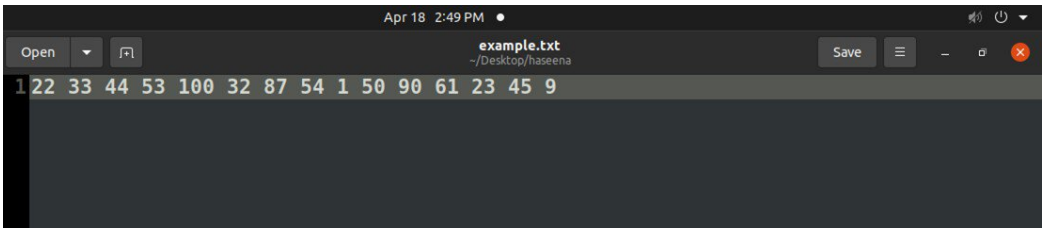
SOURCE CODE

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
public class Fileoddeven
{
static String data="";
static File dataFile=new File("example.txt");
public static void main(String args[])
{
try
{
FileWriter oddFile=new FileWriter("odd.txt");
FileWriter evenFile=new FileWriter("even.txt");
Scanner dataRead=new Scanner(dataFile);
while(dataRead.hasNextLine())
{
data+=dataRead.nextLine();
}
dataRead.close();
String values[]=data.split(" ");
for(String i:values)
{
if(Integer.parseInt(i)%2==0)
{
```

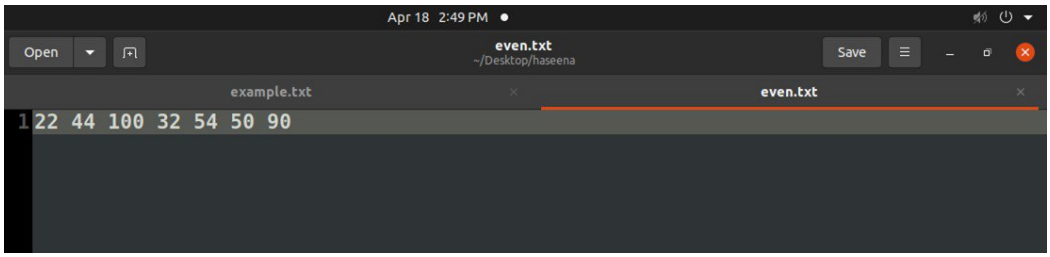
```
evenFile.write(i+" ");
}
else
{
oddFile.write(i+" ");
}
}
oddFile.close();
evenFile.close();
}
catch(IOException ex)
{
System.out.println("An error occured");
ex.printStackTrace();
}
catch(Exception ex)
{
System.out.println("An error occured:");
System.out.println(ex.getMessage());
}
}
}
```

OUTPUT

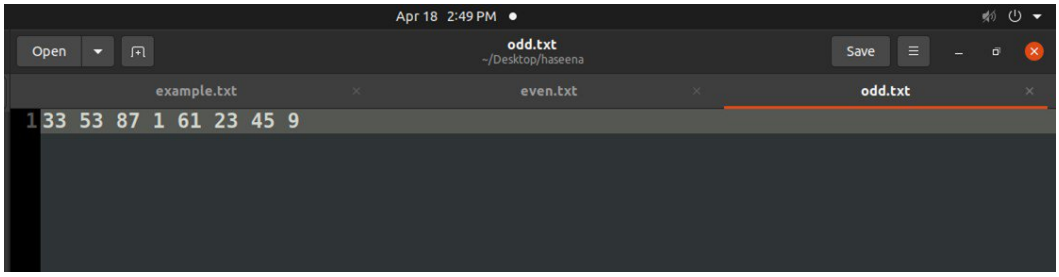
//create an example.txt file having integers



//even.txt



//odd.txt



RESULT

The program has been executed successfully and output is obtained.

PROGRAM - 19

AIM

Write a program to copy one file to another.

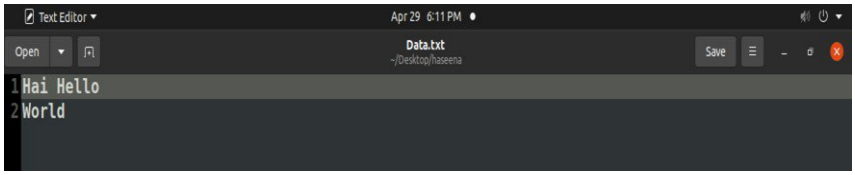
SOURCE CODE

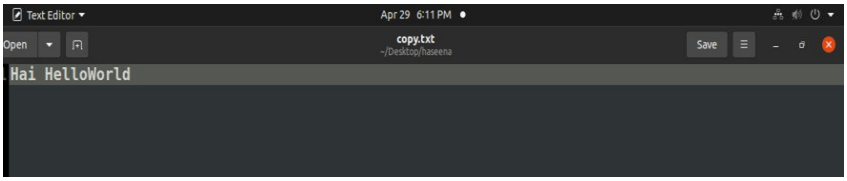
```
import java.io.FileWriter;
import java.io.IOException;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
public class File1{
public static void main(String args[])throws IOException
{
try
{
FileWriter dataWriter=new FileWriter("Data.txt");
dataWriter.write("Hai Hello");
dataWriter.write("\nWorld");
dataWriter.close();
}
catch(IOException ex)
{
System.err.println("An error occured");
ex.printStackTrace();
}
try
{
File dataFile=new File("Data.txt");
FileWriter fw=new FileWriter("copy.txt");
Scanner dataRead=new Scanner(dataFile);
String s;
while(dataRead.hasNextLine())
```

```
{
s=dataRead.nextLine();
System.out.println(s);
fw.write(s);
}
fw.close();
dataRead.close();
}
catch(FileNotFoundException ex)
{
System.out.println("An error occured");
ex.printStackTrace();
}
}
}
```

OUTPUT

```
Hai Hello
World
PS D:\New folder (2)>
```





RESULT

The program has been executed successfully and output is obtaine.

PROGRAM - 20

AIM

Client server communication using Socket – TCP/IP

SOURCE CODE

//Client.java

```
import java.io.*;
import java.net.*;
public class Client{
public static void main(String args[]) throws Exception
{
Socket socket=new Socket("localhost",4000);
PrintWriter out=new PrintWriter(socket.getOutputStream(),true);
BufferedReader in=new BufferedReader(new InputStreamReader(socket.getInputStream()));
out.println("Hello from the client");
String response=in.readLine();
System.out.println("Server response:"+response);
socket.close();
}
}
```

//Server.java

```
import java.io.*;
import java.net.*;
public class Server{
public static void main(String args[]) throws Exception
{
ServerSocket serversocket=new ServerSocket(4000);
Socket socket=serversocket.accept();
PrintWriter out=new PrintWriter(socket.getOutputStream(),true);
BufferedReader in=new BufferedReader(new InputStreamReader(socket.getInputStream()));
```

```
String message=in.readLine();
System.out.println("Client message:"+message);
out.println("Hello from the server");
socket.close();
}
}
```

OUTPUT

```
cep@cep-vostro-imp:~/Desktop/haseena$ javac Server.java
cep@cep-vostro-imp:~/Desktop/haseena$ java Server
Client message:Hello from the client
cep@cep-vostro-imp:~/Desktop/haseena$
```

```
cep@cep-vostro-imp:~/Desktop/haseena$ javac Client.java
cep@cep-vostro-imp:~/Desktop/haseena$ java Client
Server response:Hello from the server
cep@cep-vostro-imp:~/Desktop/haseena$
```

RESULT

The program has been executed successfully and output is obtained.