**CS8383-OOPS Lab Experiments**

1.Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial). Compute the bill amount using the following tariff. If the type of the EB connection is domestic, calculate the amount to be paid as follows: 9 First 100 units - Rs. 1 per unit 9 101-200 units - Rs. 2.50 per unit 9 201 -500 units - Rs. 4 per unit 9 > 501 units - Rs. 6 per unit If the type of the EB connection is commercial, calculate the amount to be paid as follows: 9 First 100 units - Rs. 2 per unit 9 101-200 units - Rs. 4.50 per unit 9 201 -500 units - Rs. 6 per unit 9 > 501 units - Rs. 7 per unit

**import** java.util.\*;

**public** **class** Ebill

{

**public** **static** **void** main (String args[])

{

Customerdata ob = **new** Customerdata();

ob.getdata();

ob.calc();

ob.display();

}

}

**class** Customerdata

{

Scanner in = **new** Scanner(System.***in***);

Scanner ins = **new** Scanner(System.***in***);

String cname,type;

**int** bn;

**double** current,previous,tbill,units;

**void** getdata()

{

System.***out***.print ("\n\t Enter consumer number ");

bn = in.nextInt();

System.***out***.print ("\n\t Enter Type of connection (D for Domestic or C for Commercial) ");

type = ins.nextLine();

System.***out***.print ("\n\t Enter consumer name ");

cname = ins.nextLine();

System.***out***.print ("\n\t Enter previous month reading ");

previous= in.nextDouble();

System.***out***.print ("\n\t Enter current month reading ");

current= in.nextDouble();

}

**void** calc()

{

units=current-previous;

**if**(type.equals("D"))

{

**if** (units<=100)

tbill=1 \* units;

**else** **if** (units>100 && units<=200)

tbill=2.50\*units;

**else** **if**(units>200 && units<=500)

tbill= 4\*units;

**else**

tbill= 6\*units;

}

**else**

{

**if** (units<=100)

tbill= 2 \* units;

**else** **if**(units>100 && units<=200)

tbill=4.50\*units;

**else** **if**(units>200 && units<=500)

tbill= 6\*units;

**else**

tbill= 7\*units;

}

}

**void** display()

{

System.***out***.println("\n\t Consumer number = "+bn);

System.***out***.println ("\n\t Consumer name = "+cname);

**if**(type.equals("D"))

System.***out***.println ("\n\t type of connection = DOMESTIC ");

**else**

System.***out***.println ("\n\t type of connection = COMMERCIAL ");

System.***out***.println ("\n\t Current Month Reading = "+current);

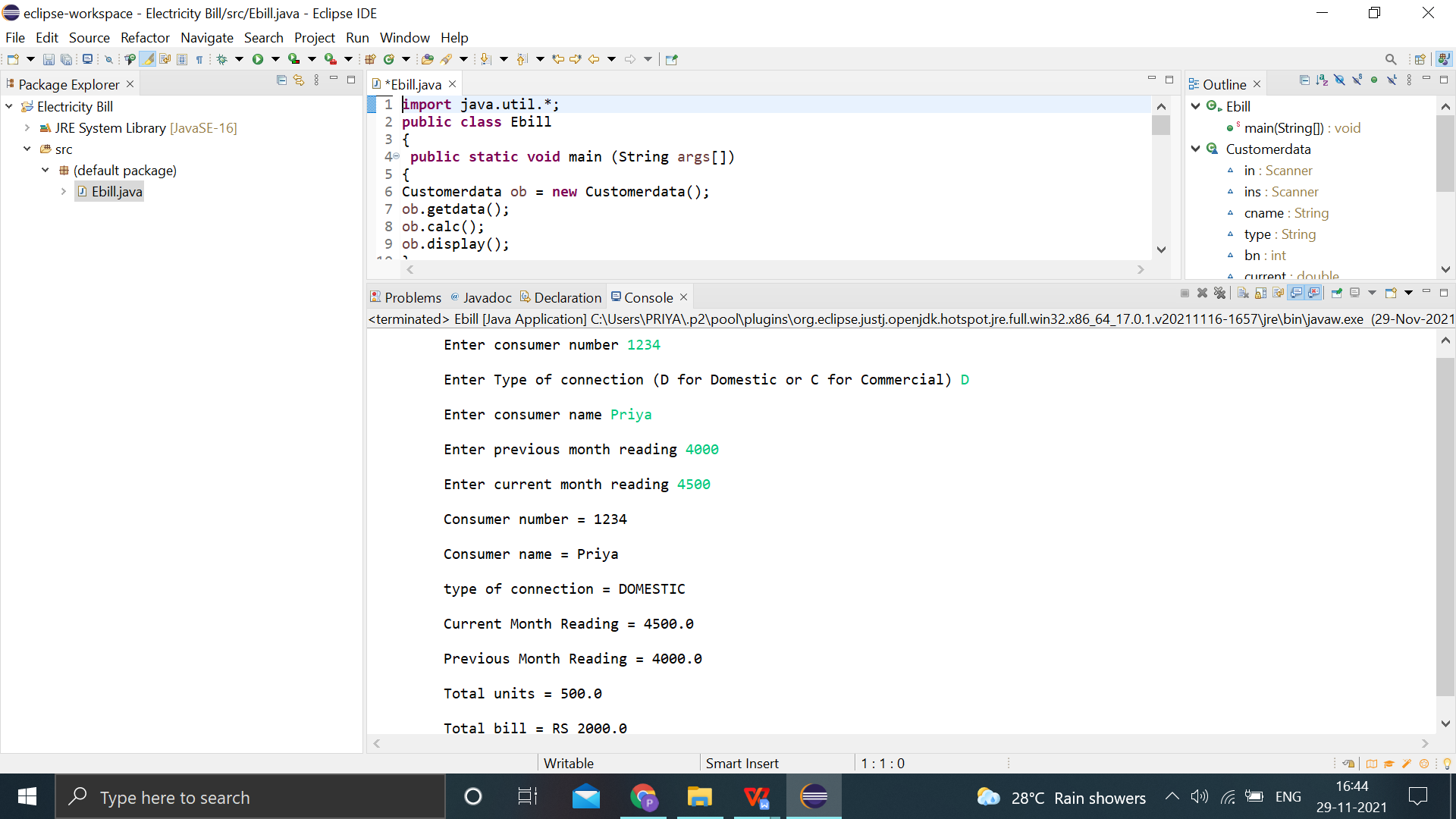
System.***out***.println ("\n\t Previous Month Reading = "+previous);

System.***out***.println ("\n\t Total units = "+units);

System.***out***.println ("\n\t Total bill = RS "+tbill);

}

}



2.Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa), time converter (hours to minutes, seconds and vice versa) using packages.

import java.util.\*;

import currencyconversion.\*;

import distanceconverion.\*;

import timeconversion.\*;

public class Converter

{

public static void main(String args[])

{

Scanner s =new Scanner(System.in);

int choice,ch;

Cost c=new Cost();

Distance d=new Distance();

Time t=new Time();

do

{

System.out.println("1.dollar to rupee ");

System.out.println("2.rupee to dollar ");

System.out.println("3.Euro to rupee ");

System.out.println("4..rupee to Euro ");

System.out.println("5.Yen to rupee ");

System.out.println("6.Rupee to Yen ");

System.out.println("7.Meter to kilometer ");

System.out.println("8.kilometer to meter ");

System.out.println("9.Miles to kilometer ");

System.out.println("10.kilometer to miles");

System.out.println("11.Hours to Minutes");

System.out.println("12.Hours to Seconds");

System.out.println("13.Seconds to Hours");

System.out.println("14.Minutes to Hours");

System.out.println("Enter ur choice");

choice=s.nextInt();

switch(choice)

{

case 1:

{

c.dollartorupee();

break;

}

case 2:

{

c.rupeetodollar();

break;

}

case 3:

{

c.eurotorupee();

break;

}

case 4:

{

c.rupeetoeuro();

break;

}

case 5:

{

c.yentorupee();

break;

}

case 6:

{

c.rupeetoyen();

break;

}

case 7:

{

d.mtokm();

break;

}

case 8:

{

d.kmtom();

break;

}

case 9:

{

d.milestokm();

break;

}

case 10 :

{

d.kmtomiles();

break;

}

case 11:

{

t.hourstominutes();

break;

}

case 12:

{

t.hourstoseconds();

break;

}

case 13:

{

t.secondstohours();

break;

}

case 14:

{

t.minutestohours();

break;

}

}

System.out.println("Enter 0 to quit and 1 to continue ");

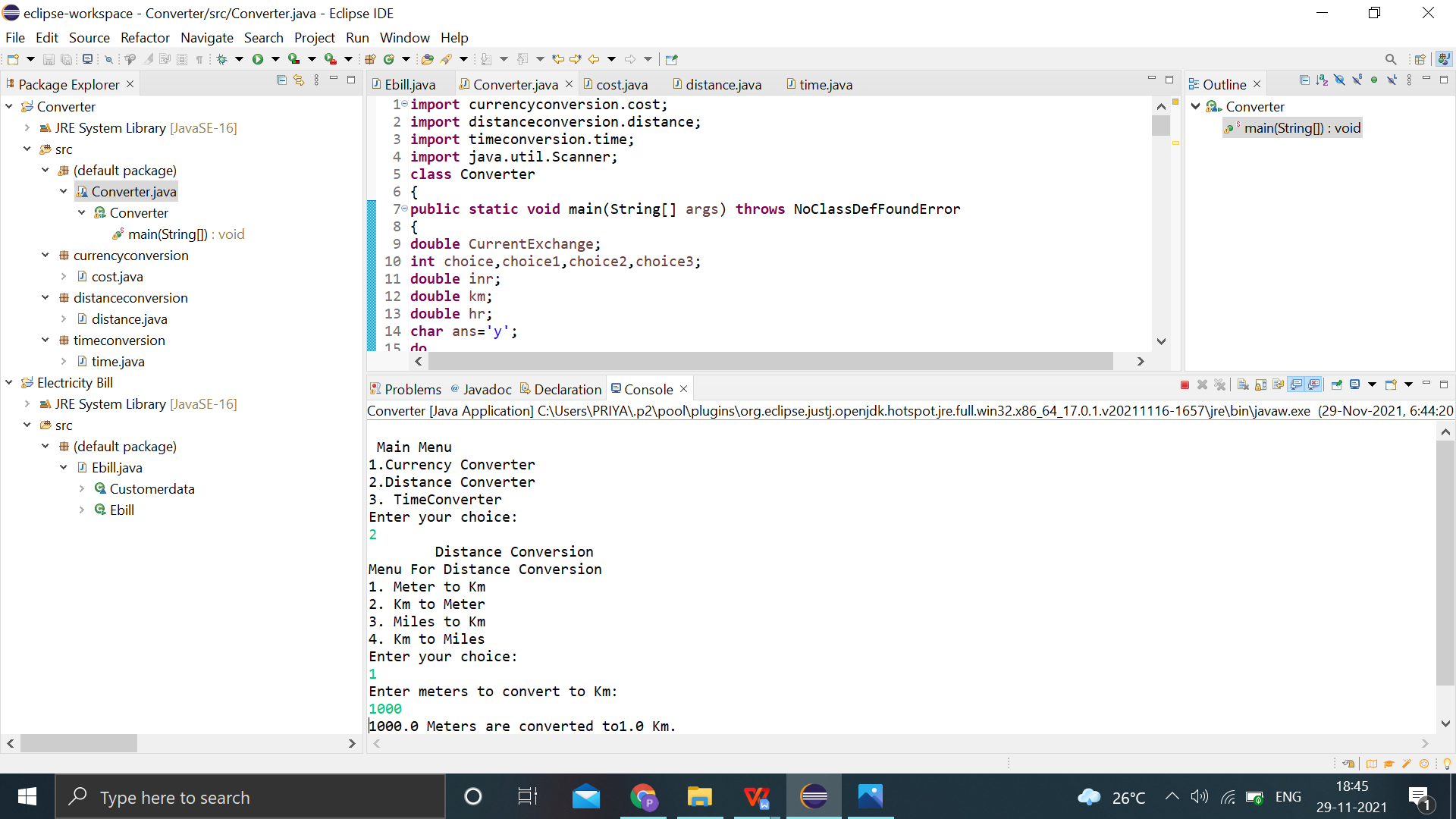
ch=s.nextInt();

}

while(ch==1);

}

}



3.Develop a java application with Employee class with Emp\_name, Emp\_id, Address, Mail\_id, Mobile\_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary

**import** java.util.\*;

**class** Employee

{

**int** empid;

**long** mobile;

String name, address, mailid;

Scanner get = **new** Scanner(System.***in***);

**void** getdata()

{

System.***out***.println("Enter Name of the Employee");

name = get.nextLine();

System.***out***.println("Enter Mail id");

mailid = get.nextLine();

System.***out***.println("Enter Address of the Employee:");

address = get.nextLine();

System.***out***.println("Enter employee id ");

empid = get.nextInt();

System.***out***.println("Enter Mobile Number");

mobile = get.nextLong();

}

**void** display()

{

System.***out***.println("Employee Name: "+name);

System.***out***.println("Employee id : "+empid);

System.***out***.println("Mail id : "+mailid);

System.***out***.println("Address: "+address);

System.***out***.println("Mobile Number: "+mobile);

}

}

**class** Programmer **extends** Employee

{

**double** salary,bp,da,hra,pf,club,net,gross;

**void** getprogrammer()

{

System.***out***.println("Enter basic pay");

bp = get.nextDouble();

}

**void** calculateprog()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("PAY SLIP FOR PROGRAMMER");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("Basic Pay: Rs. "+bp);

System.***out***.println("DA: Rs. "+da);

System.***out***.println("HRA: Rs. "+hra);

System.***out***.println("PF: Rs. "+pf);

System.***out***.println("CLUB: Rs. "+club);

System.***out***.println("GROSS PAY: Rs. "+gross);

System.***out***.println("NET PAY: Rs. "+net);

}

}

**class** Asstprofessor **extends** Employee

{

**double** salary,bp,da,hra,pf,club,net,gross;

**void** getasst()

{

System.***out***.println("Enter basic pay");

bp = get.nextDouble();

}

**void** calculateasst()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("PAY SLIP FOR ASSISTANT PROFESSOR");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("Basic Pay: Rs. "+bp);

System.***out***.println("DA: Rs. "+da);

System.***out***.println("HRA: Rs. "+hra);

System.***out***.println("PF: Rs. "+pf);

System.***out***.println("CLUB: Rs. "+club);

System.***out***.println("GROSS PAY: Rs. "+gross);

System.***out***.println("NET PAY: Rs. "+net);

}

}

**class** Associateprofessor **extends** Employee

{

**double** salary,bp,da,hra,pf,club,net,gross;

**void** getassociate()

{

System.***out***.println("Enter basic pay");

bp = get.nextDouble();

}

**void** calculateassociate()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("PAY SLIP FOR ASSOCIATE PROFESSOR");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("Basic Pay: Rs. "+bp);

System.***out***.println("DA: Rs. "+da);

System.***out***.println("HRA: Rs. "+hra);

System.***out***.println("PF: Rs. "+pf);

System.***out***.println("CLUB: Rs. "+club);

System.***out***.println("GROSS PAY: Rs. "+gross);

System.***out***.println("NET PAY: Rs. "+net);

}

}

**class** Professor **extends** Employee

{

**double** salary,bp,da,hra,pf,club,net,gross;

**void** getprofessor()

{

System.***out***.println("Enter basic pay");

bp = get.nextDouble();

}

**void** calculateprofessor()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("PAY SLIP FOR PROFESSOR");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("Basic Pay: Rs. "+bp);

System.***out***.println("DA: Rs. "+da);

System.***out***.println("HRA: Rs. "+hra);

System.***out***.println("PF: Rs. "+pf);

System.***out***.println("CLUB: Rs. "+club);

System.***out***.println("GROSS PAY: Rs. "+gross);

System.***out***.println("NET PAY: Rs. "+net);

}

}

**public** **class** Salary

{

**public** **static** **void** main(String args[])

{

**int** choice,cont;

**do**

{

System.***out***.println("PAYROLL");

System.***out***.println("1.PROGRAMMER \t 2.ASSISTANT PROFESSOR \t 3.ASSOCIATE PROFESSOR \t 4.PROFESSOR \t");

Scanner c = **new** Scanner(System.***in***);

System.***out***.print("Enter Your Choice:");

choice=c.nextInt();

**switch**(choice)

{

**case** 1:

{

Programmer p=**new** Programmer();

p.getdata();

p.getprogrammer();

p.display();

p.calculateprog();

**break**;

}

**case** 2:

{

Asstprofessor asst=**new** Asstprofessor();

asst.getdata();

asst.getasst();

asst.display();

asst.calculateasst();

**break**;

}

**case** 3:

{

Associateprofessor asso=**new** Associateprofessor();

asso.getdata();

asso.getassociate();

asso.display();

asso.calculateassociate();

**break**;

}

**case** 4:

{

Professor prof=**new** Professor();

prof.getdata();

prof.getprofessor();

prof.display();

prof.calculateprofessor();

**break**;

}

}

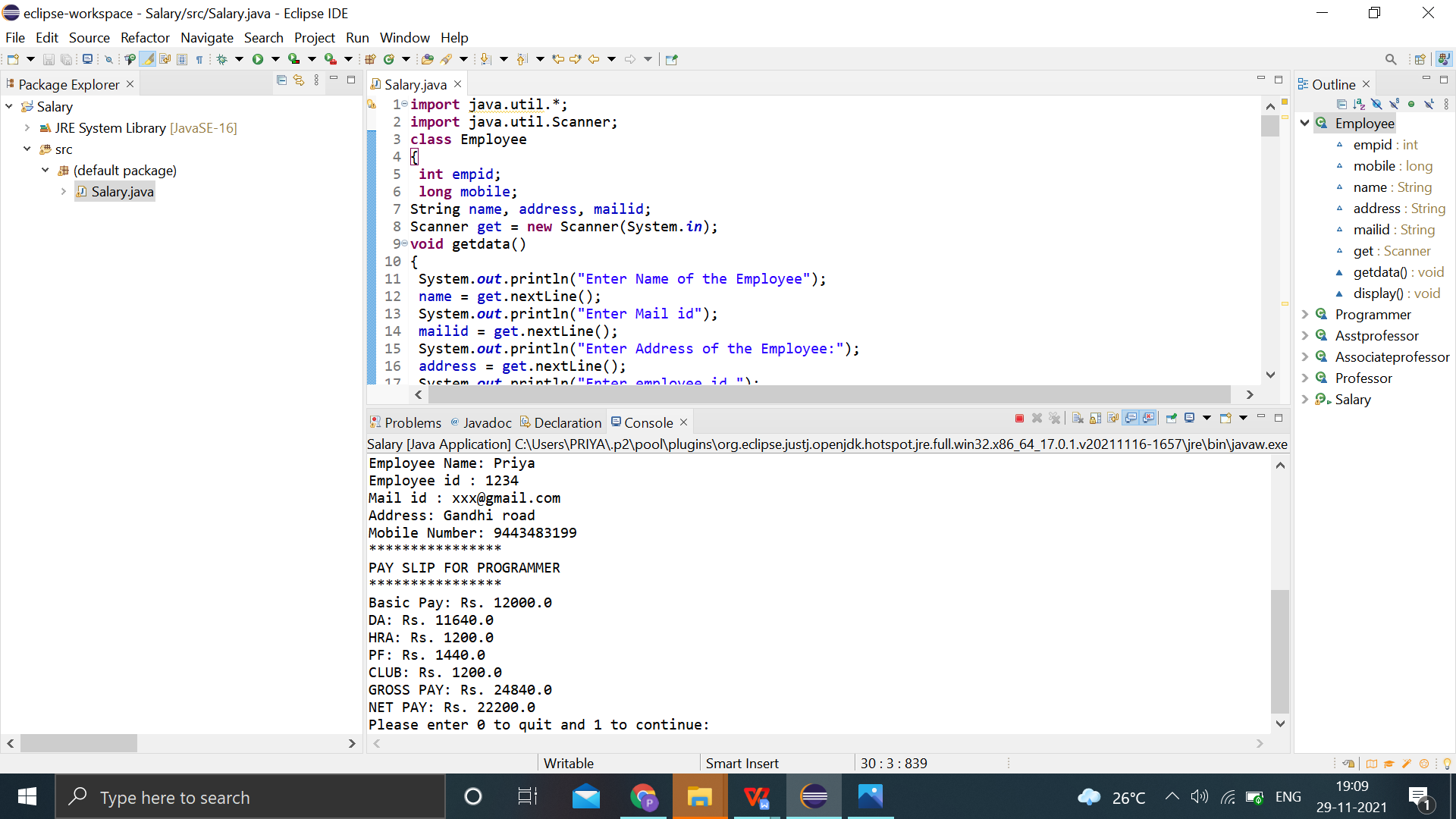
System.***out***.print("Please enter 0 to quit and 1 to continue: ");

cont=c.nextInt();

}**while**(cont==1);

}

}



4.Design a Java interface for ADT Stack. Implement this interface using array. Provide necessary exception handling in both the implementations.

**import** java.io.\*;

**class** Astack **implements** Stackoperation

{

**int** stack[]=**new** **int**[5];

**int** top=-1;

**int** i;

**public** **void** push(**int** item)

{

**if**(top>=4)

{

System.***out***.println("Overflow");

}

**else**

{

top=top+1;

stack[top]=item;

System.***out***.print("Element pushed: "+stack[top]);

}

}

**public** **void** pop()

{

**if**(top<0)

System.***out***.println("Underflow");

**else**

{

System.***out***.print("Element popped: "+stack[top]);

top=top-1;

}

}

**public** **void** display()

{

**if**(top<0)

System.***out***.println("No Element in stack");

**else**

{

**for**(i=0;i<=top;i++)

System.***out***.println("Element:"+stack[i]);

}

}

}

**public** **class** Teststack

{

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

{

**int** ch,c;

**int** i;

Astack s=**new** Astack();

DataInputStream in=**new** DataInputStream(System.***in***);

**do**

{

**try**

{

System.***out***.println("ARRAY STACK");

System.***out***.println("1.Push 2.Pop 3.Display 4.Exit");

System.***out***.print("Enter your Choice:");

ch=Integer.*parseInt*(in.~~readLine~~());

**switch**(ch)

{

**case** 1:

System.***out***.print("Enter the value to push:");

i=Integer.*parseInt*(in.~~readLine~~());

s.push(i);

**break**;

**case** 2:

s.pop();

**break**;

**case** 3:

System.***out***.println("The elements are: ");

s.display();

**break**;

**default**:

**break**;

}

}

**catch**(IOException e)

{

System.***out***.println("IO Error");

}

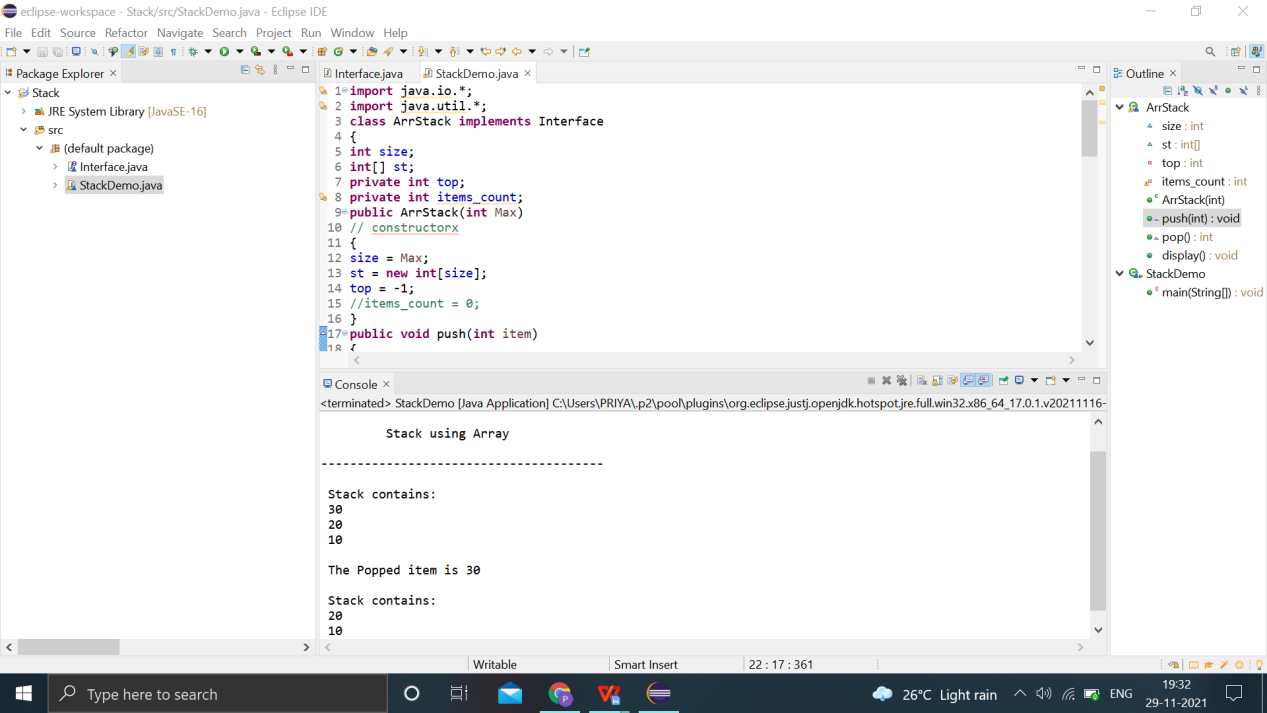
System.***out***.println("Please enter 0 to quit and 1 to continue ");

c=Integer.*parseInt*(in.~~readLine~~());

}**while**(c==1);

}

}



5.Write a program to perform string operations using ArrayList. Write functions for the following a. Append - add at end b. Insert – add at particular index c. Search d. List all string starts with given letter

import java.util.\*;

import java.io.\*;

public class Arraylistexample

{

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

{

ArrayList<String> obj = new ArrayList<String>();

DataInputStream in=new DataInputStream(System.in);

int c,ch;

int i,j;

String str,str1;

do

{

System.out.println("STRING MANIPULATION");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("1. Append at end \t 2.Insert at particular index \t 3.Search \t");

System.out.println("4. List string that starting with letter \t");

System.out.println("5. Size \t 6.Remove \t 7.Sort \t 8.Display\t" );

System.out.println("Enter the choice ");

c=Integer.parseInt(in.readLine());

switch(c)

{

case 1:

{

System.out.println("Enter the string ");

str=in.readLine();

obj.add(str);

break;

}

case 2:

{

System.out.println("Enter the string ");

str=in.readLine();

System.out.println("Specify the index/position to insert");

i=Integer.parseInt(in.readLine());

obj.add(i-1,str);

System.out.println("The array list has following elements:"+obj);

break;

}

case 3:

{

System.out.println("Enter the string to search ");

str=in.readLine();

j=obj.indexOf(str);

if(j==-1)

System.out.println("Element not found");

else

System.out.println("Index of:"+str+"is"+j);

break;

}

case 4:

{

System.out.println("Enter the character to List string that starts with specified character\\");

str=in.readLine();

for(i=0;i<(obj.size()-1);i++)

{

str1=obj.get(i);

if(str1.startsWith(str))

{

System.out.println(str1);

}

}

break;

}

case 5:

{

System.out.println("Size of the list "+obj.size());

break;

}

case 6:

{

System.out.println("Enter the element to remove");

str=in.readLine();

if(obj.remove(str))

{

System.out.println("Element Removed"+str);

}

else

{

System.out.println("Element not present");

}

break;

}

case 7:

{

Collections.sort(obj);

System.out.println("The array list has following elements:"+obj);

break;

}

case 8:

{

System.out.println("The array list has following elements:"+obj);

break;

}

}

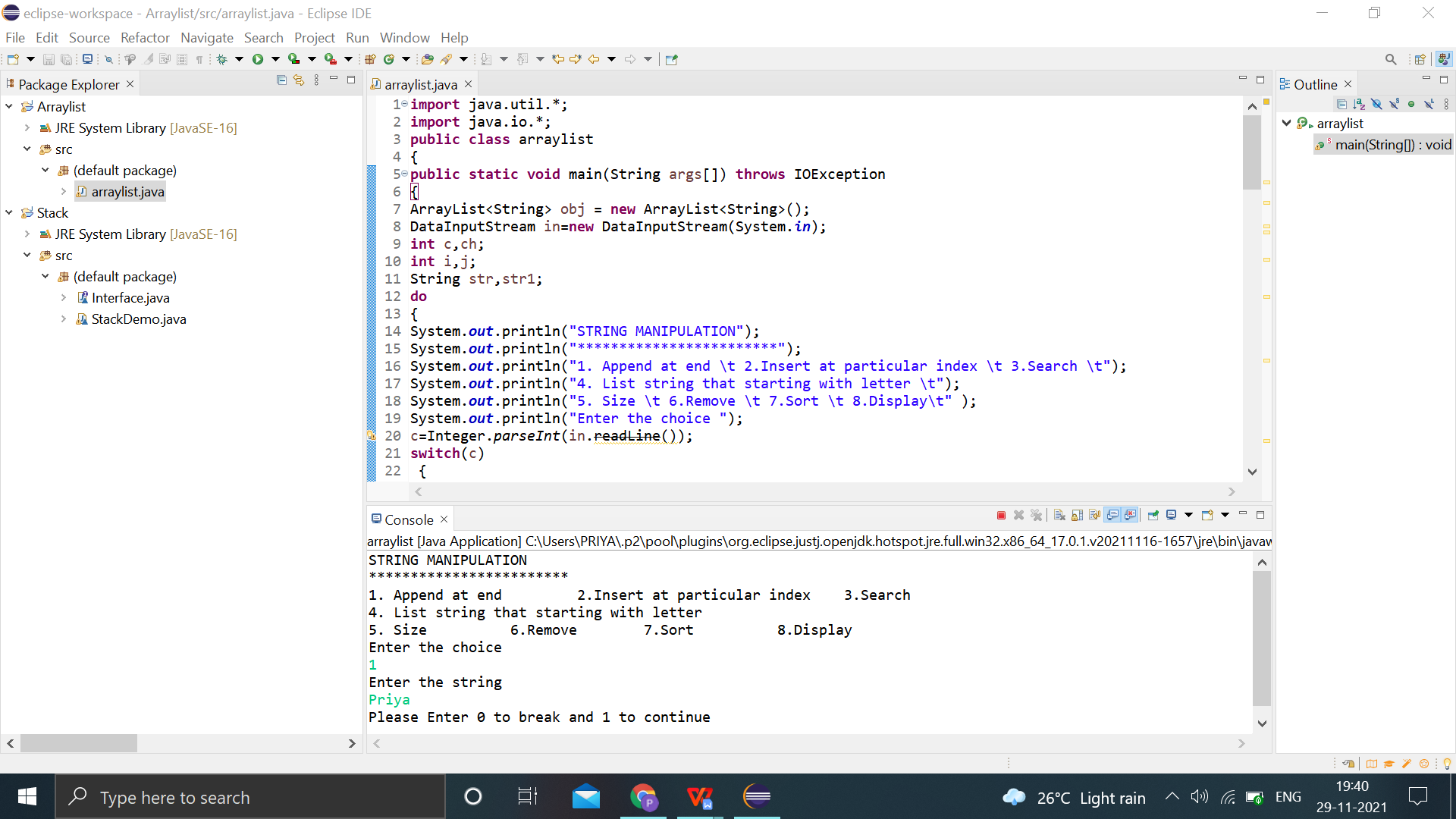
System.out.println("Please Enter 0 to break and 1 to continue");

ch=Integer.parseInt(in.readLine())

}while(ch==1);

}

}



6.Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

**import** java.util.\*;

**class** rectangle **extends** shape

{

**public** **int** area\_rect;

**public** **void** printarea()

{

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("Enter the length and breadth of rectangle");

a=s.nextInt();

b=s.nextInt();

area\_rect=a\*b;

System.***out***.println("Length of rectangle: "+a +"breadth of rectangle: "+b);

System.***out***.println("The area of rectangle is:"+area\_rect);

}

}

**class** triangle **extends** shape

{

**double** area\_tri;

**public** **void** printarea()

{

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("Enter the base and height of triangle:");

a=s.nextInt();

b=s.nextInt();

System.***out***.println("Base of triangle: "+a +"height of triangle: "+b);

area\_tri=(0.5\*a\*b);

System.***out***.println("The area of triangle is:"+area\_tri);

}

}

**class** circle **extends** shape

{

**double** area\_circle;

**public** **void** printarea()

{

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("Enter the radius of circle:");

a=s.nextInt();

area\_circle=(3.14\*a\*a);

System.***out***.println("Radius of circle:"+a);

System.***out***.println("The area of circle is:"+area\_circle);

}

}

**public** **class** Shapeclass

{

**public** **static** **void** main(String[] args)

{

rectangle r=**new** rectangle();

r.printarea();

triangle t=**new** triangle();

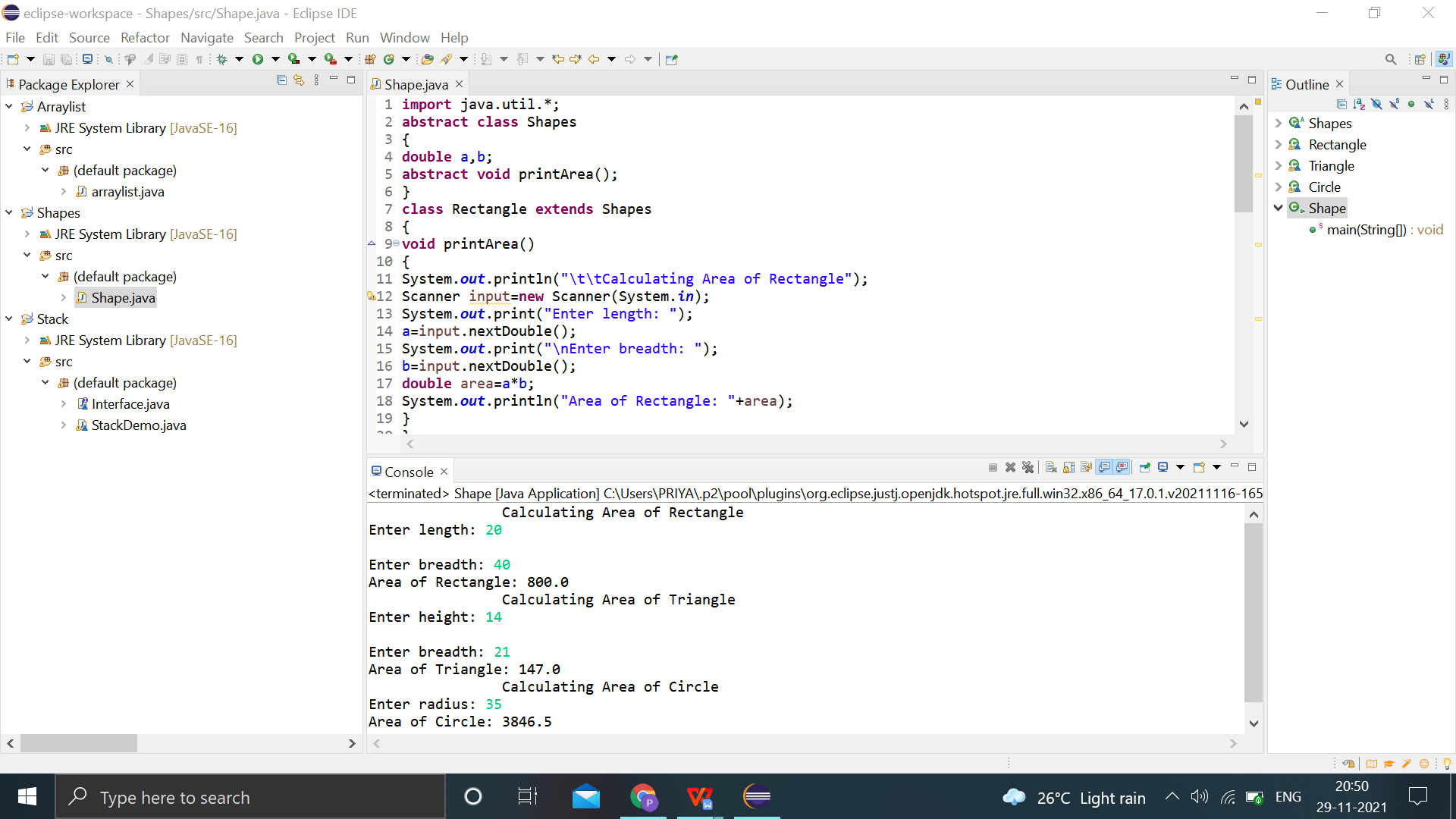
t.printarea();

circle r1=**new** circle();

r1.printarea();

}

}



7.Write a Java program to implement user defined exception handling.

**import** java.util.\*;

**class** NegativeAmtException **extends** Exception

{

String msg;

NegativeAmtException(String msg)

{

**this**.msg=msg;

}

**public** String toString()

{

**return** msg;

}

}

**public** **class** Userdefined

{

**public** **static** **void** main(String[] args)

{

Scanner s=**new** Scanner(System.***in***);

System.***out***.print("Enter Amount:");

**int** a=s.nextInt();

**try**

{

**if**(a<0)

{

**throw** **new** NegativeAmtException("Invalid Amount");

}

System.***out***.println("Amount Deposited");

}

**catch**(NegativeAmtException e)

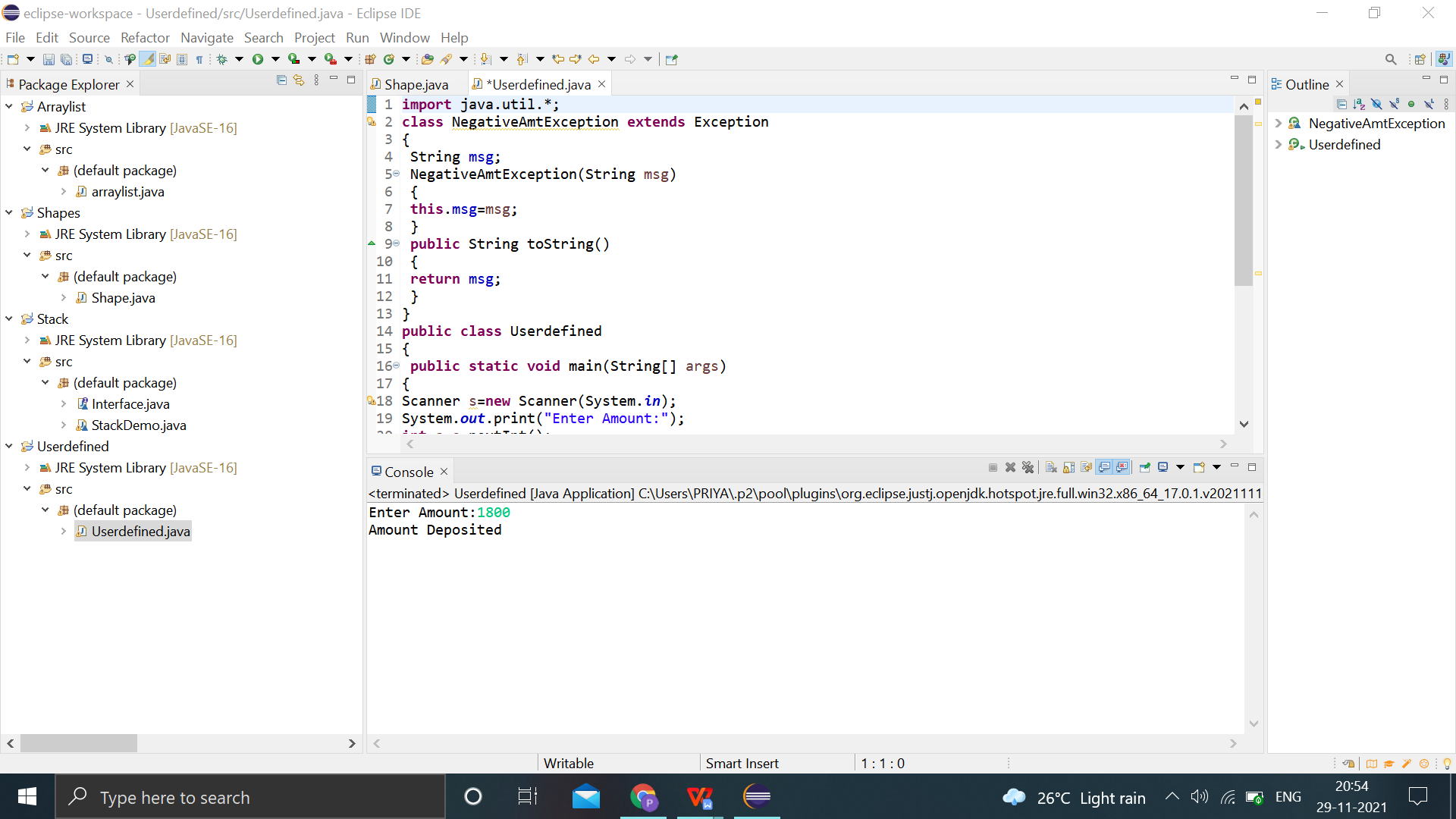
{

System.***out***.println(e);

}

}

}



8.Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

import java.io.\*;

import java.util.\*;

public class Filedemo

{

public static void main(String args[])

{

String filename;

Scanner s=new Scanner(System.in);

System.out.println("Enter the file name ");

filename=s.nextLine();

File f1=new File(filename);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("FILE INFORMATION");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("NAME OF THE FILE "+f1.getName());

System.out.println("PATH OF THE FILE "+f1.getPath());

System.out.println("PARENT"+f1.getParent());

if(f1.exists())

System.out.println("THE FILE EXISTS ");

else

System.out.println("THE FILE DOES NOT ExISTS ");

if(f1.canRead())

System.out.println("THE FILE CAN BE READ ");

else

System.out.println("THE FILE CANNOT BE READ ");

if(f1.canWrite())

System.out.println("WRITE OPERATION IS PERMITTED");

else

System.out.println("WRITE OPERATION IS NOT PERMITTED");

if(f1.isDirectory())

System.out.println("IT IS A DIRECTORY ");

else

System.out.println("NOT A DIRECTORY");

if(f1.isFile())

System.out.println("IT IS A FILE ");

else

System.out.println("NOT A FILE");

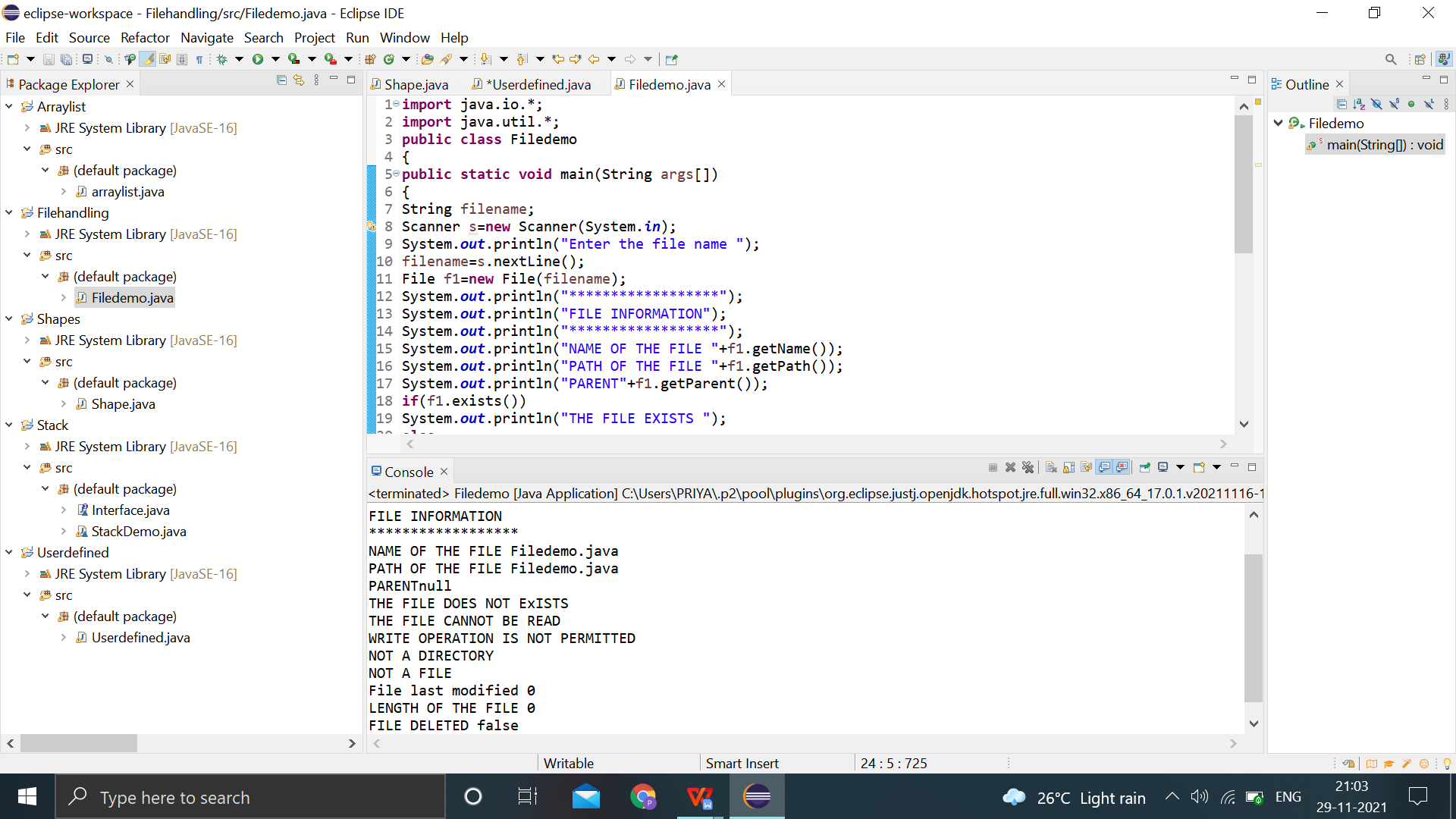
System.out.println("File last modified "+ f1.lastModified());

System.out.println("LENGTH OF THE FILE "+f1.length());

System.out.println("FILE DELETED "+f1.delete());

}

}



9.Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

**import** java.util.\*;

**class** even **implements** Runnable

{

**public** **int** x;

**public** even(**int** x)

{

**this**.x = x;

}

**public** **void** run()

{

System.***out***.println("New Thread "+ x +" is EVEN and Square of " + x + " is: " + x \* x);

}

}

**class** odd **implements** Runnable

{

**public** **int** x;

**public** odd(**int** x)

{

**this**.x = x;

}

**public** **void** run()

{

System.***out***.println("New Thread "+ x +" is ODD and Cube of " + x + " is: " + x \* x \* x);}

}

**class** A **extends** Thread

{

**public** **void** run()

{

**int** num = 0;

Random r = **new** Random();

**try**

{

**for** (**int** i = 0; i < 5; i++)

{

num = r.nextInt(100);

System.***out***.println("Main Thread and Generated Number is " + num);

**if** (num % 2 == 0)

{

Thread t1 = **new** Thread(**new** even(num));

t1.start();

}

**else**

{

Thread t2 = **new** Thread(**new** odd(num));

t2.start();

}

Thread.*sleep*(1000);

System.***out***.println("--------------------------------------");

}}

**catch** (Exception ex)

{

System.***out***.println(ex.getMessage());

}}

}

**public** **class** Multithreadprog

{

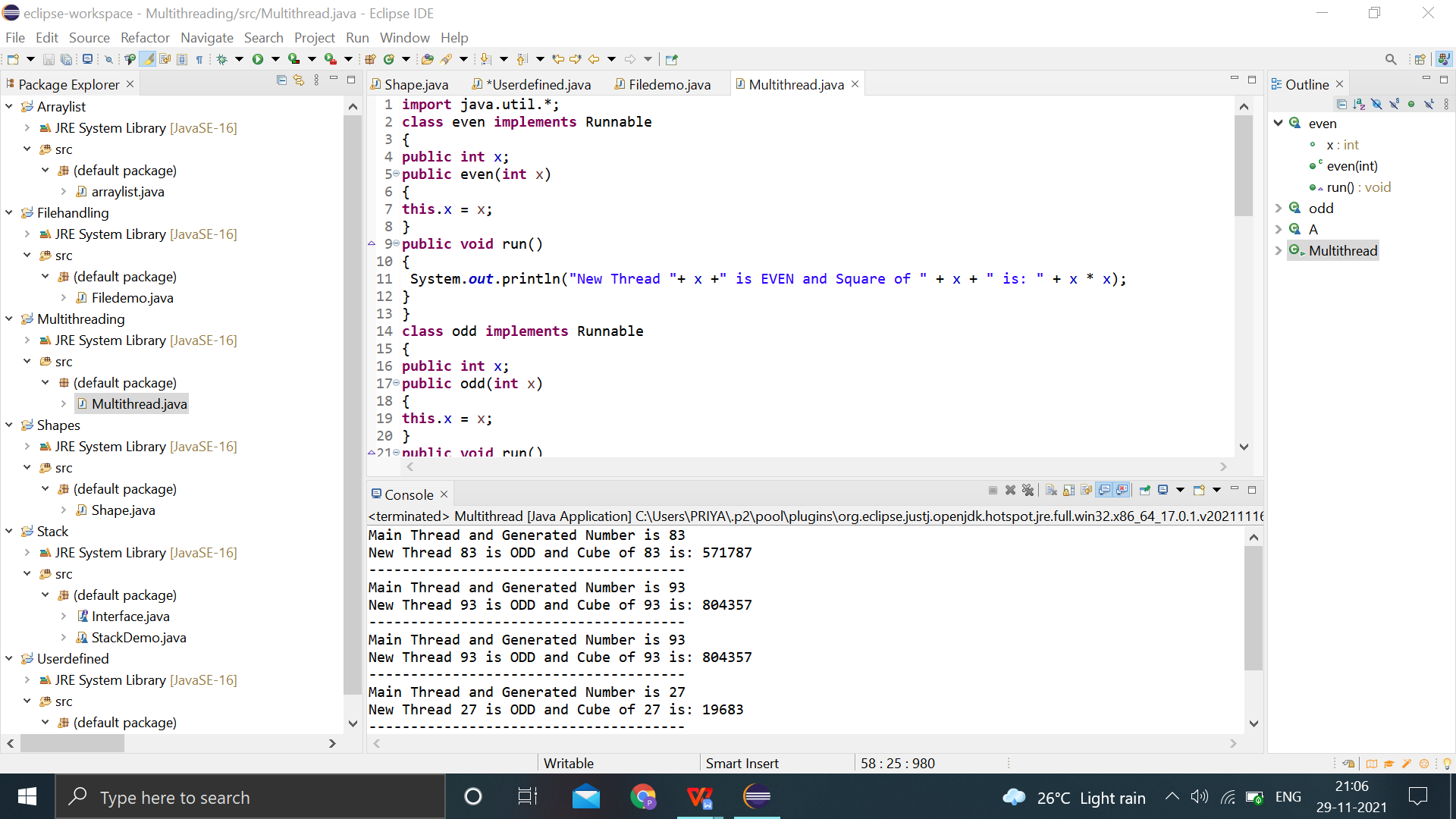
**public** **static** **void** main(String[] args)

{

A a = **new** A();

a.start();

}}



10.Write a java program to find the maximum value from the given type of elements using a generic function.

**class** MyClass<T **extends** Comparable<T>>

{

T[] vals;

MyClass(T[] o)

{

vals = o;

}

**public** T min()

{

T v = vals[0];

**for**(**int** i=1; i < vals.length; i++)

**if**(vals[i].compareTo(v) < 0)

v = vals[i];

**return** v;

}

**public** T max()

{

T v = vals[0];

**for**(**int** i=1; i < vals.length;i++)

**if**(vals[i].compareTo(v) > 0)

v = vals[i];

**return** v;

}

}

**public** **class** Genericdemo

{

**public** **static** **void** main(String args[]) {

**int** i;

Integer inums[]={10,2,5,4,6,1};

Character chs[]={'v','p','s','a','n','h'};

Double d[]={20.2,45.4,71.6,88.3,54.6,10.4};

MyClass<Integer> iob = **new** MyClass<Integer>(inums);

MyClass<Character> cob = **new** MyClass<Character>(chs);

MyClass<Double>dob = **new** MyClass<Double>(d);

System.***out***.println("Max value in inums: " + iob.max());

System.***out***.println("Min value in inums: " + iob.min());

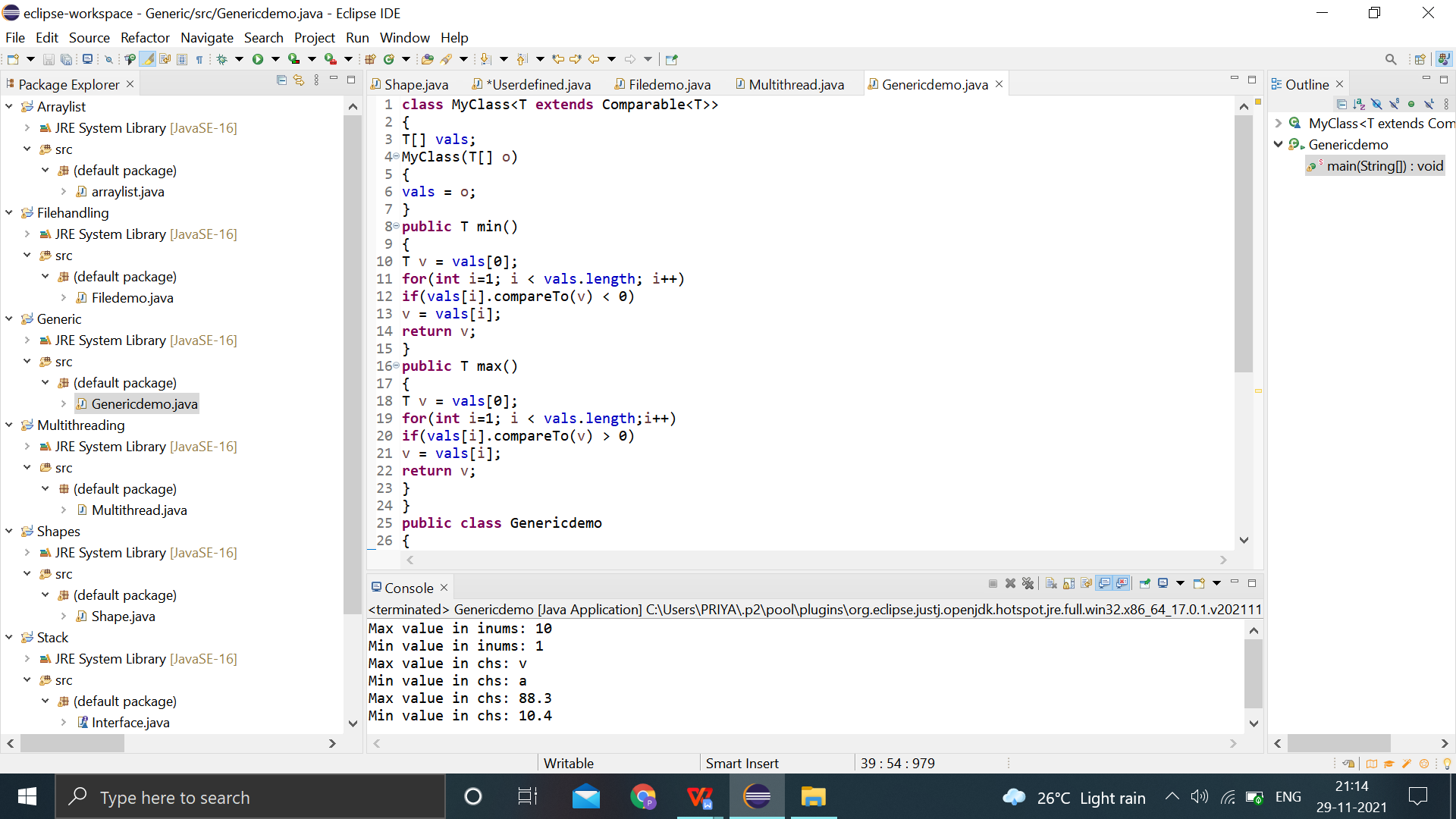
System.***out***.println("Max value in chs: " + cob.max());

System.***out***.println("Min value in chs: " + cob.min());

System.***out***.println("Max value in chs: " + dob.max());

System.***out***.println("Min value in chs: " + dob.min());

}}



11.Design a calculator using event-driven programming paradigm of Java with the following options. a) Decimal manipulations b) Scientific manipulations

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import javax.swing.event.\*;

public class ScientificCalculator extends JFrame implements ActionListener

{

JTextField tfield;

double temp, temp1, result, a;

static double m1, m2;

int k = 1, x = 0, y = 0, z = 0;

char ch;

JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, zero, clr, pow2, pow3, exp,

fac, plus, min, div, log, rec, mul, eq, addSub, dot, mr, mc, mp,

mm, sqrt, sin, cos, tan;

Container cont;

JPanel textPanel, buttonpanel;

ScientificCalculator()

{

cont = getContentPane();

cont.setLayout(new BorderLayout());

JPanel textpanel = new JPanel();

tfield = new JTextField(25);

tfield.setHorizontalAlignment(SwingConstants.RIGHT);

tfield.addKeyListener(new KeyAdapter()

{

public void keyTyped(KeyEvent keyevent)

{

char c = keyevent.getKeyChar();

if (c >= '0' &&c <= '9') { }

else

{

keyevent.consume();

}

}

});

textpanel.add(tfield);

buttonpanel = new JPanel();

buttonpanel.setLayout(new GridLayout(8, 4, 2, 2));

boolean t = true;

mr = new JButton("MR");

buttonpanel.add(mr);

mr.addActionListener(this);

mc = new JButton("MC");

buttonpanel.add(mc);

mc.addActionListener(this);

mp = new JButton("M+");

buttonpanel.add(mp);

mp.addActionListener(this);

mm = new JButton("M-");

buttonpanel.add(mm);

mm.addActionListener(this);

b1 = new JButton("1");

buttonpanel.add(b1);

b1.addActionListener(this);

b2 = new JButton("2");

buttonpanel.add(b2);

b2.addActionListener(this);

b3 = new JButton("3");

buttonpanel.add(b3);

b3.addActionListener(this);

b4 = new JButton("4");

buttonpanel.add(b4);

b4.addActionListener(this);

b5 = new JButton("5");

buttonpanel.add(b5);

b5.addActionListener(this);

b6 = new JButton("6");

buttonpanel.add(b6);

b6.addActionListener(this);

b7 = new JButton("7");

buttonpanel.add(b7);

b7.addActionListener(this);

b8 = new JButton("8");

buttonpanel.add(b8);

b8.addActionListener(this);

b9 = new JButton("9");

buttonpanel.add(b9);

b9.addActionListener(this);

zero = new JButton("0");

buttonpanel.add(zero);

zero.addActionListener(this);

plus = new JButton("+");

buttonpanel.add(plus);

plus.addActionListener(this);

min = new JButton("-");

buttonpanel.add(min);

min.addActionListener(this);

mul = new JButton("\*");

buttonpanel.add(mul);

mul.addActionListener(this);

div = new JButton("/");

div.addActionListener(this);

buttonpanel.add(div);

addSub = new JButton("+/-");

buttonpanel.add(addSub);

addSub.addActionListener(this);

dot = new JButton(".");

buttonpanel.add(dot);

dot.addActionListener(this);

eq = new JButton("=");

buttonpanel.add(eq);

eq.addActionListener(this);

rec = new JButton("1/x");

buttonpanel.add(rec);

rec.addActionListener(this);

sqrt = new JButton("Sqrt");

buttonpanel.add(sqrt);

sqrt.addActionListener(this);

log = new JButton("log");

buttonpanel.add(log);

log.addActionListener(this);

sin = new JButton("SIN");

buttonpanel.add(sin);

sin.addActionListener(this);

cos = new JButton("COS");

buttonpanel.add(cos);

cos.addActionListener(this);

tan = new JButton("TAN");

buttonpanel.add(tan);

tan.addActionListener(this);

pow2 = new JButton("x^2");

buttonpanel.add(pow2);

pow2.addActionListener(this);

pow3 = new JButton("x^3");

buttonpanel.add(pow3);

pow3.addActionListener(this);

exp = new JButton("Exp");

exp.addActionListener(this);

buttonpanel.add(exp);

fac = new JButton("n!");

fac.addActionListener(this);

buttonpanel.add(fac);

clr = new JButton("AC");

buttonpanel.add(clr);

clr.addActionListener(this);

cont.add("Center", buttonpanel);

cont.add("North", textpanel);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent e)

{

String s = e.getActionCommand();

if (s.equals("1"))

{

if (z == 0)

{

tfield.setText(tfield.getText() + "1");

}

else {

tfield.setText("");

tfield.setText(tfield.getText() + "1");

z = 0;

}

}

if (s.equals("2")) {

if (z == 0) {

tfield.setText(tfield.getText() + "2");

}

else

{

tfield.setText("");

tfield.setText(tfield.getText() + "2");

z = 0;

}

}

if (s.equals("3")) {

if (z == 0) {

tfield.setText(tfield.getText() + "3");

}

else

{

tfield.setText("");

tfield.setText(tfield.getText() + "3");

z = 0;

}

}

if (s.equals("4")) {

if (z == 0) {

tfield.setText(tfield.getText() + "4");

}

else

{

tfield.setText("");

tfield.setText(tfield.getText() + "4");

z = 0;

}

}

if (s.equals("5")) {

if (z == 0) {

tfield.setText(tfield.getText() + "5");

}

else

{

tfield.setText("");

tfield.setText(tfield.getText() + "5");

z = 0;

}

}

if (s.equals("6")) {

if (z == 0) {

tfield.setText(tfield.getText() + "6");

}

else

{

tfield.setText("");

tfield.setText(tfield.getText() + "6");

z = 0;

}

}

if (s.equals("7")) {

if (z == 0) {

tfield.setText(tfield.getText() + "7");

}

else

{

tfield.setText("");

tfield.setText(tfield.getText() + "7");

z = 0;

}

}

if (s.equals("8")) {

if (z == 0) {

tfield.setText(tfield.getText() + "8");

}

else

{

tfield.setText("");

tfield.setText(tfield.getText() + "8");

z = 0;

}

}

if (s.equals("9")) {

if (z == 0) {

tfield.setText(tfield.getText() + "9");

}

else

{

tfield.setText("");

tfield.setText(tfield.getText() + "9");

z = 0;

}

}

if (s.equals("0"))

{

if (z == 0) {

tfield.setText(tfield.getText() + "0");

}

else

{

tfield.setText("");

tfield.setText(tfield.getText() + "0");

z = 0;

}

}

if (s.equals("AC")) {

tfield.setText("");

x = 0;

y = 0;

z = 0;

}

if (s.equals("log"))

{

if (tfield.getText().equals("")) {

tfield.setText("");

}

else

{

a = Math.log(Double.parseDouble(tfield.getText()));

tfield.setText("");

tfield.setText(tfield.getText() + a);

}

}

if (s.equals("1/x")) {

if (tfield.getText().equals("")) {

tfield.setText("");

}

else

{

a = 1 / Double.parseDouble(tfield.getText());

tfield.setText("");

tfield.setText(tfield.getText() + a);

}

}

if (s.equals("Exp")) {

if (tfield.getText().equals("")) {

tfield.setText("");

}

else

{

a = Math.exp(Double.parseDouble(tfield.getText()));

tfield.setText("");

tfield.setText(tfield.getText() + a);

}

}

if (s.equals("x^2")) {

if (tfield.getText().equals("")) {

tfield.setText("");

}

else

{

a = Math.pow(Double.parseDouble(tfield.getText()), 2);

tfield.setText("");

tfield.setText(tfield.getText() + a);

}

}

if (s.equals("x^3")) {

if (tfield.getText().equals("")) {

tfield.setText("");

}

else

{

a = Math.pow(Double.parseDouble(tfield.getText()), 3);

tfield.setText("");

tfield.setText(tfield.getText() + a);

}

}

if (s.equals("+/-")) {

if (x == 0) {

tfield.setText("-" + tfield.getText());

x = 1;

}

else

{

tfield.setText(tfield.getText());

}

}

if (s.equals(".")) {

if (y == 0) {

tfield.setText(tfield.getText() + ".");

y = 1;

}

else

{

tfield.setText(tfield.getText());

}

}

if (s.equals("+"))

{

if (tfield.getText().equals(""))

{

tfield.setText("");

temp = 0;

ch = '+';

}

else

{

temp = Double.parseDouble(tfield.getText());

tfield.setText("");

ch = '+';

y = 0;

x = 0;

}

tfield.requestFocus();

}

if (s.equals("-"))

{

if (tfield.getText().equals(""))

{

tfield.setText("");

temp = 0;

ch = '-';

}

else

{

x = 0;

y = 0;

temp = Double.parseDouble(tfield.getText());

tfield.setText("");

ch = '-';

}

tfield.requestFocus();

}

if (s.equals("/")) {

if (tfield.getText().equals(""))

{

tfield.setText("");

temp = 1;

ch = '/';

}

else

{

x = 0;

y = 0;

temp = Double.parseDouble(tfield.getText());

ch = '/';

tfield.setText("");

}

tfield.requestFocus();

}

if (s.equals("\*")) {

if (tfield.getText().equals(""))

{

tfield.setText("");

temp = 1;

ch = '\*';

}

else

{

x = 0;

y = 0;

temp = Double.parseDouble(tfield.getText());

ch = '\*';

tfield.setText("");

}

tfield.requestFocus();

}

if (s.equals("MC"))

{

m1 = 0;

tfield.setText("");

}

if (s.equals("MR"))

{

tfield.setText("");

tfield.setText(tfield.getText() + m1);

}

if (s.equals("M+"))

{

if (k == 1) {

m1 = Double.parseDouble(tfield.getText());

k++;

}

else

{

m1 += Double.parseDouble(tfield.getText());

tfield.setText("" + m1);

}

}

if (s.equals("M-"))

{

if (k == 1) {

m1 = Double.parseDouble(tfield.getText());

k++;

}

else

{

m1 -= Double.parseDouble(tfield.getText());

tfield.setText("" + m1);

}

}

if (s.equals("Sqrt"))

{

if (tfield.getText().equals(""))

{

tfield.setText("");

}

else

{

a = Math.sqrt(Double.parseDouble(tfield.getText()));

tfield.setText("");

field.setText(tfield.getText() + a);

}

}

if (s.equals("SIN"))

{

if (tfield.getText().equals(""))

{

tfield.setText("");

}

else

{

a = Math.sin(Double.parseDouble(tfield.getText()));

tfield.setText("");

tfield.setText(tfield.getText() + a);

}

}

if (s.equals("COS"))

{

if (tfield.getText().equals(""))

{

tfield.setText("");

}

else

{

a = Math.cos(Double.parseDouble(tfield.getText()));

tfield.setText("");

tfield.setText(tfield.getText() + a);

}

}

if (s.equals("TAN")) {

if (tfield.getText().equals("")) {

tfield.setText("");

}

else

{

a = Math.tan(Double.parseDouble(tfield.getText()));

tfield.setText("");

tfield.setText(tfield.getText() + a);

}

}

if (s.equals("="))

{

if (tfield.getText().equals(""))

{

tfield.setText("");

}

else

{

temp1 = Double.parseDouble(tfield.getText());

switch (ch)

{

case '+':

result = temp + temp1;

break;

case '-':

result = temp - temp1;

break;

case '/':

result = temp / temp1;

break;

case '\*':

result = temp \* temp1;

break;

}

tfield.setText("");

tfield.setText(tfield.getText() + result);

z = 1;

}

}

if (s.equals("n!"))

{

if (tfield.getText().equals(""))

{

tfield.setText("");

}

else

{

a = fact(Double.parseDouble(tfield.getText()));

tfield.setText("");

tfield.setText(tfield.getText() + a);

}

}

tfield.requestFocus();

}

double fact(double x)

{

int er = 0;

if (x < 0)

{

er = 20;

return 0;

}

double i, s = 1;

for (i = 2; i <= x; i += 1.0)

s \*= i;

return s;

}

public static void main(String args[])

{

try

{

UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");

}

catch (Exception e)

{

}

ScientificCalculator f = new ScientificCalculator();

f.setTitle("ScientificCalculator");

f.pack();

f.setVisible(true);

}

}

