**DHANALAKSHMI COLLEGE OF ENGINEERING**

**TAMBARAM, CHENNAI —601 301**



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**CS8383 — OBJECT ORIENTED PROGRAMMING LABORATORY V SEMESTER - R 2017**



**Name**

**Reg. No.**

**Year**

**CS8383 – OBJECT ORIENTED PROGRAMMING LABORATORY INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S No.** | **Name of the Experiment** | **Page no.** | **Date** | **Signature of the staff** |
| 1 | Electricity Bill Generation | 3 |  |  |
| 2 | Currency converter, Distance converter  and Time converter | 6 |  |  |
| 3 | Payroll Processing | 10 |  |  |
| 4 | ADT Stack | 14 |  |  |
| 5 | String Operations | 17 |  |  |
| 6 | Abstract Class | 20 |  |  |
| 7 | Exception Handling | 22 |  |  |
| 8 | File information | 24 |  |  |
| 9 | Multithreading | 26 |  |  |
| 10 | Generic Function | 29 |  |  |
| 11 | Calculator | 31 |  |  |
| 12 | Mini Project – Text editor using String | 39 |  |  |

# `Ex. No. 1 ELECTRICITY BILL GENERATION Date: 13.08.20

**Aim:** To develop a Java application to generate Electricity bill.

**Procedure:**

1. Start the program
2. Create consumer class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial).
3. If the type of the EB connection is domestic, calculate the amount to be paid as follows:
   * First 100 units - Rs.1 per unit
   * 101-200 units - Rs.2.50 per unit
   * 201 -500 units - Rs. 4 per unit
   * > 501 units - Rs.6 per unit
4. If the type of the EB connection is commercial, calculate the amount to be paid as follows:
   * First 100 units - Rs. 2 per unit
   * 101-200 units - Rs. 4.50 per unit
   * 201 -500 units - Rs. 6 per unit
   * > 501 units - Rs. 7 per unit
5. Print the amount
6. Stop the program

**Source code:**

import java.io.\*; public class ebbill

{

int consumerno; String consumername; int previousreading; int currentreading;

int connectiontype; float billamount=0.0f; int unitconsumed=0;

public ebbill(int consumerno, String consumername,int previousreading,int currentreading,int connectiontype)

{

this.consumerno=consumerno; this.consumername=consumername; this.previousreading=previousreading; this.currentreading=currentreading; this.connectiontype=connectiontype;

}

void calculate()

{

if(previousreading>currentreading)

{

System.out.println("error in current month");

}

else

{

unitconsumed=currentreading-previousreading;

}

switch(connectiontype)

{

case 1: if(unitconsumed>=0&&unitconsumed<=100)

{

billamount=unitconsumed\*1;

}

else if(unitconsumed>100&&unitconsumed<=200)

{

billamount=unitconsumed\*2.5f;

}

else if(unitconsumed>200&&unitconsumed<=500)

{

billamount=unitconsumed\*4;

}

else{

billamount=unitconsumed\*6;

}

break; case 2:

if(unitconsumed>=0&&unitconsumed<=100)

{

billamount=unitconsumed\*2;

}

else if(unitconsumed>=100&&unitconsumed<=200)

{

billamount=unitconsumed\*4.5f;

}

else if(unitconsumed>200&&unitconsumed<=500)

{

billamount=unitconsumed\*6;

}

else

{

billamount=unitconsumed\*7;

}

break;

}

}

void print()

{

System.out.println("consumer number:"+consumerno); System.out.println("consumer name:"+consumername); System.out.println("connection type:"+connectiontype); System.out.println("unit consumed:"+unitconsumed); System.out.println("bill amount:"+billamount);

}

public static void main(String args[])

{

ebbill ebbill1=new ebbill(123,"Kumar",230,400,1); ebbill1.calculate();

ebbill1.print();

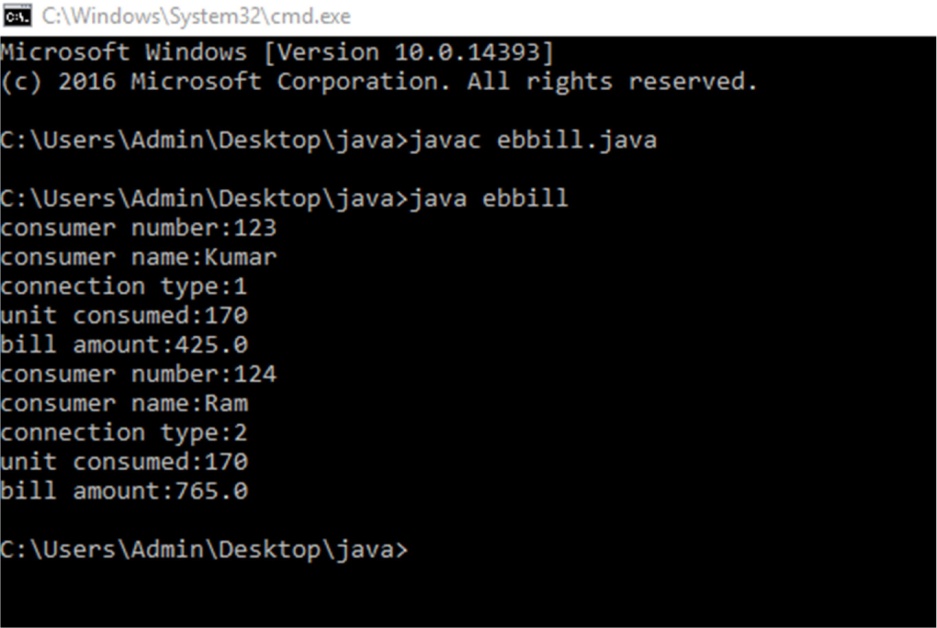
ebbill ebbill2=new ebbill(124,"Ram",230,400,2); ebbill2.calculate();

ebbill2.print();

}

}

**Sample output**:



**Result:** Thus the Electricity Bill Java application was successfully executed.



5

**Ex. No.:2 Date: 20.08.20**

**Currency Converter, Distance Converter and Time Converter**

**Aim:**

To develop a Java application to implement currency converter , distance converter and time converter using packages.

**Procedure:**

1. . Start the program
2. Create codes for currency converter , distance converter and time converter.
3. Print the converted value.
4. Stop the program

**Source code:**

import java.util.Scanner; import java.io.\*;

class si {

public void calculate()

{

double inr,USD,YEN,EURO,Meter,KM,Hr,Min,Sec; int ch;

Scanner in=new Scanner(System.in);

System.out.println("1.USD to INR and INR to USD\n2.EURO to INR and INR to EURO\n 3.Yen to INR and INR to Yen\n 4.Meter to Kilometer and Kilometer to Meter\n5.Hours to Minutes and Seconds");

System.out.println("Enter Your Choice:"); ch=in.nextInt();

switch(ch)

{

case 1:

System.out.println("Enter USD to convert into INR");

USD=in.nextInt(); inr=USD\*60;

System.out.println("USD=" +USD+ " is INR=" +inr); System.out.println("Enter INR to convert into USD"); inr=in.nextInt();

USD=inr/60;

System.out.println("INR=" +inr+ " is USD=" +USD); break;

case 2:

System.out.println("Enter EURO to convert into INR:"); EURO=in.nextInt();

inr=EURO\*80.78;

System.out.println("EURO=" +EURO+ " is INR=" +inr); System.out.println("Enter INR to convert into EURO:"); inr=in.nextDouble();

EURO=inr/80.78;

System.out.println("INR=" +inr+ " is EURO =" + EURO); break;

case 3:

System.out.println("Enter YEN to convert into INR"); YEN=in.nextDouble();

inr=YEN/1.64;

System.out.println("YEN=" +YEN+ " is INR=" +inr); System.out.println("Enter INR to convert into YEN"); inr=in.nextDouble();

YEN=inr\*1.64;

System.out.println("INR=" +inr+ " is YEN=" +YEN); break;

case 4:

System.out.println("Enter Meter to convert into Kilometer:"); Meter=in.nextDouble();

KM=Meter/1000;

System.out.println("Meter=" +Meter+ " is KM=" +KM); System.out.println("Enter Kilometer to convert into Meter:"); KM=in.nextDouble();

Meter=KM\*1000;

System.out.println("KM=" +KM+ " is Meter = " + Meter); break;

case 5:

System.out.println("Enter Hours to convert into Minutes and seconds:"); Hr=in.nextDouble();

Min=Hr\*60; Sec=Hr\*3600;

System.out.println("Hr=" +Hr+ "is " +Min+ "min and" + Sec+ " Seconds "); System.out.println("Covert Minutes and Seconds into Hours"); System.out.println("Enter min");

Min=in.nextDouble(); Hr=Min/60;

System.out.println("Min=" +Min+ "is" +Hr+ "hr"); System.out.println("Enter seconds"); Sec=in.nextDouble();

Hr=Sec/3600;

System.out.println("Sec=" +Sec+ "is" +Hr+ "hr"); break;

default: System.exit(0);

}

}

public static void main(String args[])

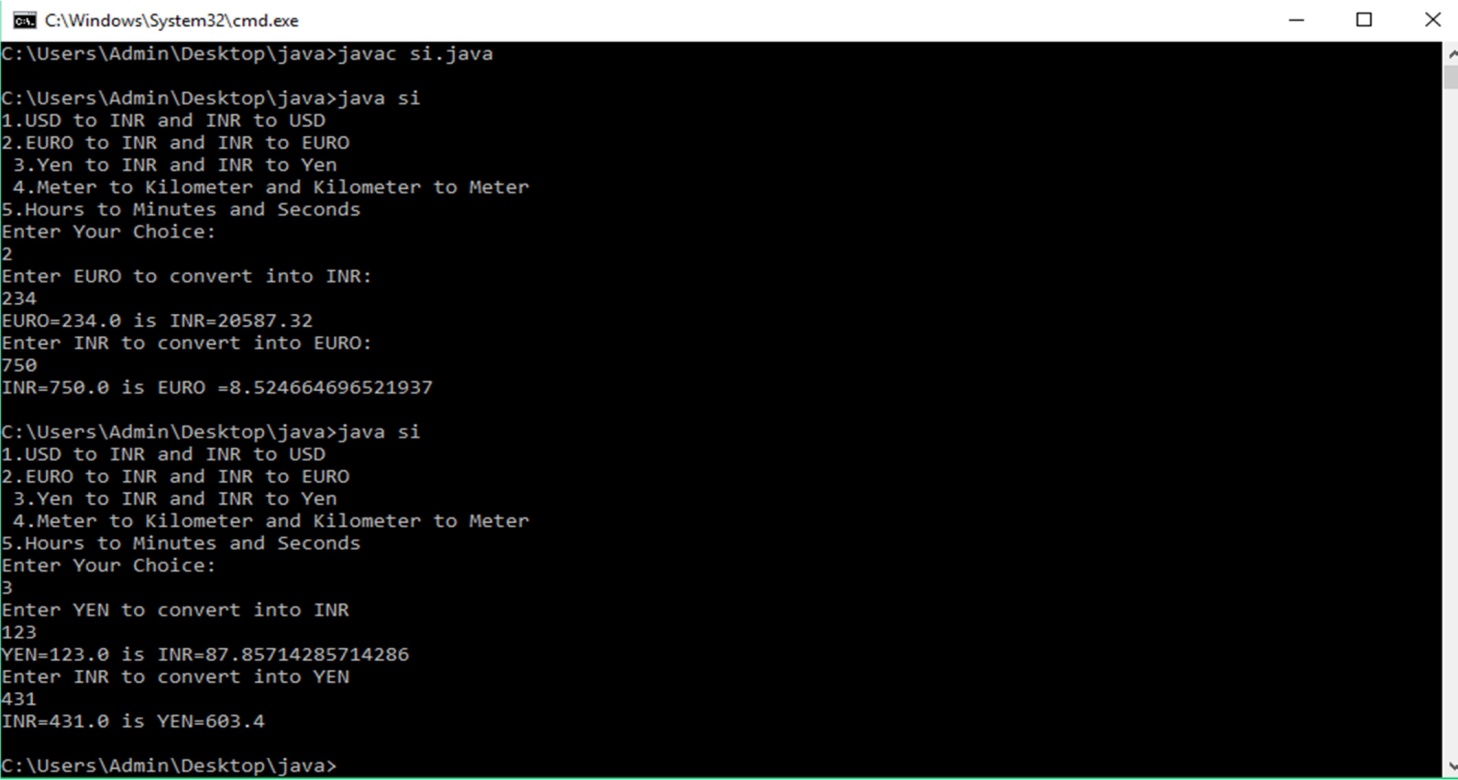
{

si x= new si(); x.calculate();

}

}

**Sample output:**



**Result:**

Thus the Java application has been created for currency conversion, distance conversion and time conversion and it was successfully executed.



9

**Ex. No.:3 PAYROLL PROCESSING Date: 27.08.20**

**Aim**:

To develop a Java application with employee class and generate pay slips for the employees with their gross and net salary.

**Procedure:**

1. Start the program
2. Create Employee class with Emp name and Emp id.
3. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class.
4. 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.

1. Generate pay slips for the employees with their gross and net salary.
2. Stop the program

**Source code:**

import java.io.\*;

import java.util.Scanner; class employee

{

Double mobno;

String name,address,eid,mailid; Scanner get =new Scanner(System.in); employee()

{

System.out.println("Enter name of the Employee:"); name=get.nextLine();

System.out.println("Enter the ID of the Employee:"); eid=get.nextLine();

}

void display()

{

System.out.println("Employee name:"+name); System.out.println("Employee id:"+eid);

}

}

class programmer extends employee

{

double bp,rate,net; programmer()

{

System.out.println("Enter basic pay:"); bp=get.nextFloat();

}

void calculatepay1()

{

rate=bp+(0.97\*bp)+(0.01\*bp)+(0.12\*bp)+(0.001\*bp); net=rate-(0.12\*bp);

}

void display()

{

System.out.println("====================="+"\n"+"Programmer Details"+"\n"+"========================"+"\n");

super.display(); System.out.println("Gross salary:"+rate); System.out.println("net salary:"+net);

}

}

class assistantprofessor extends employee

{

double bp,rate,net; assistantprofessor()

{

System.out.println("Enter basic pay:"); bp=get.nextFloat();

}

void calculatepay2()

{

rate=bp+(0.97\*bp)+(0.01\*bp)+(0.12\*bp)+(0.001\*bp); net=rate-(0.12\*bp);

}

void display()

{

System.out.println("====================="+"\n"+"Assistant Professor Details"+"\n"+"========================"+"\n");

super.display(); System.out.println("Gross salary:"+rate); System.out.println("net salary:"+net);

}

}

class associateprofessor extends employee

{

double bp,rate,net; associateprofessor()

{

System.out.println("Enter basic pay:"); bp=get.nextFloat();

}

void calculatepay3()

{

rate=bp+(0.97\*bp)+(0.01\*bp)+(0.12\*bp)+(0.001\*bp); net=rate-(0.12\*bp);

}

void display()

{

System.out.println("====================="+"\n"+"Associate Professor Details"+"\n"+"========================"+"\n");

super.display(); System.out.println("Gross salary:"+rate); System.out.println("net salary:"+net);

}

}

class professor extends employee

{

double bp,rate,net; professor()

{

System.out.println("Enter basic pay:"); bp=get.nextFloat();

}

void calculatepay4()

{

rate=bp+(0.97\*bp)+(0.01\*bp)+(0.12\*bp)+(0.001\*bp); net=rate-(0.12\*bp);

}

void display()

{

System.out.println("====================="+"\n"+" Professor Details"+"\n"+"========================"+"\n");

super.display(); System.out.println("Gross salary:"+rate); System.out.println("net salary:"+net);

}

}

class salary

{

public static void main(String args[])

{

System.out.println("====================="+"\n"+"Enter Programmer Details"+"\n"+"========================"+"\n");

programmer ob1=new programmer(); ob1.calculatepay1();

ob1.display();

System.out.println("====================="+"\n"+"Enter Assistant Professor Details"+"\n"+"========================"+"\n");

assistantprofessor ob2=new assistantprofessor(); ob2.calculatepay2();

ob2.display();

System.out.println("====================="+"\n"+"Enter Associate Professor

Details"+"\n"+"========================"+"\n");

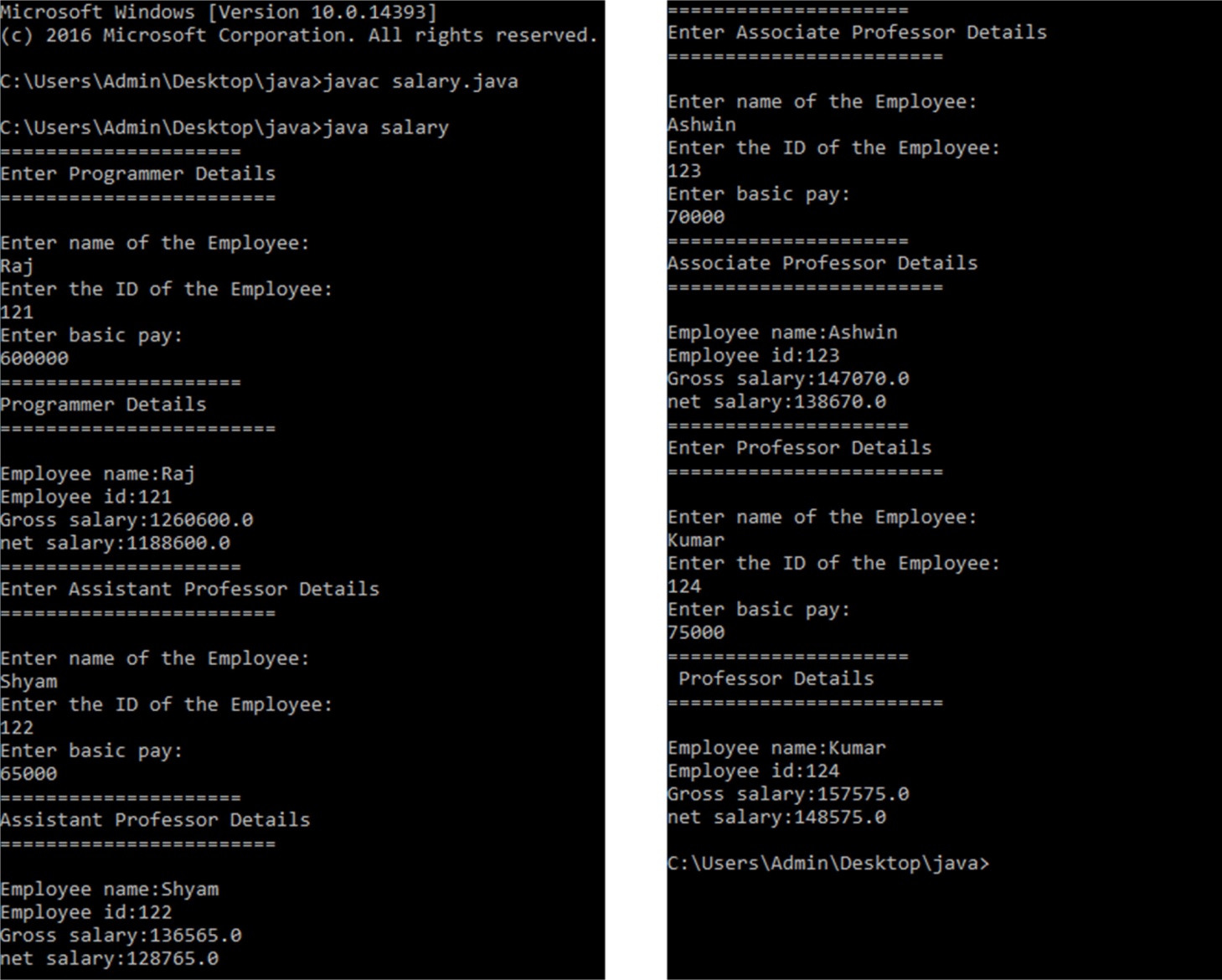
associateprofessor ob3=new associateprofessor(); ob3.calculatepay3();

ob3.display(); System.out.println("====================="+"\n"+"Enter Professor Details"+"\n"+"========================"+"\n");

professor ob4=new professor(); ob4.calculatepay4(); ob4.display();

}}

**Sample output:**



**Result:**

Thus the Java application has been created with employee class and pay slips are generated for the employees with their gross and net salary.



13

**Ex. No.:4 ADT STACK Date: 03.09.20**

**Aim** To design a Java interface for ADT Stack using array.

**Procedure:**

1. Start the program
2. Define the interface.
3. Read the elements using array.
4. Initialize stackTop pointer as zero,
5. Define and use the method Push() to insert the elements into the stack with ‘STACK OVERFLOW’ condition.
6. Define and use the method pop() to remove an element from an array with ‘STACK UNDERFLOW’ condition
7. Display the output.

**Source code:**

import java.io.\*; interface Mystack

{

public void pop(); public void push(); public void display();

}

class Stack\_array implements Mystack

{

final static int n=5;

int stack[]=new int[n]; int top=-1;

public void push()

{

try

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in)); if(top==(n-1))

{

System.out.println(" Stack Overflow"); return;

}

else

{

System.out.println("Enter the element"); int ele=Integer.parseInt(br.readLine()); stack[++top]=ele;

}

}

catch(IOException e)

{

System.out.println("e");

}

}

public void pop()

{

if(top<0)

{

System.out.println("Stack underflow"); return;

}

else

{

int popper=stack[top]; top--;

System.out.println("Popped element:" +popper);

}

}

public void display()

{

if(top<0)

{

System.out.println("Stack is empty"); return;

}

else

{

String str=" ";

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

{str=str+"A "+stack[i]+" <--";} System.out.println("Elements are:"+str);

}

}

}

class stackADT

{

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

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in)); System.out.println("Implementation of Stack using Array");

Stack\_array stk=new Stack\_array(); int ch=0;

do

{

System.out.println("1.Push 2.Pop 3.Display 4.Exit"); System.out.println("Enter your choice:"); ch=Integer.parseInt(br.readLine());

switch(ch)

{

case 1:

stk.push(); break; case 2:

stk.pop(); break; case 3:

stk.display(); break;

case 4:

System.exit(0);

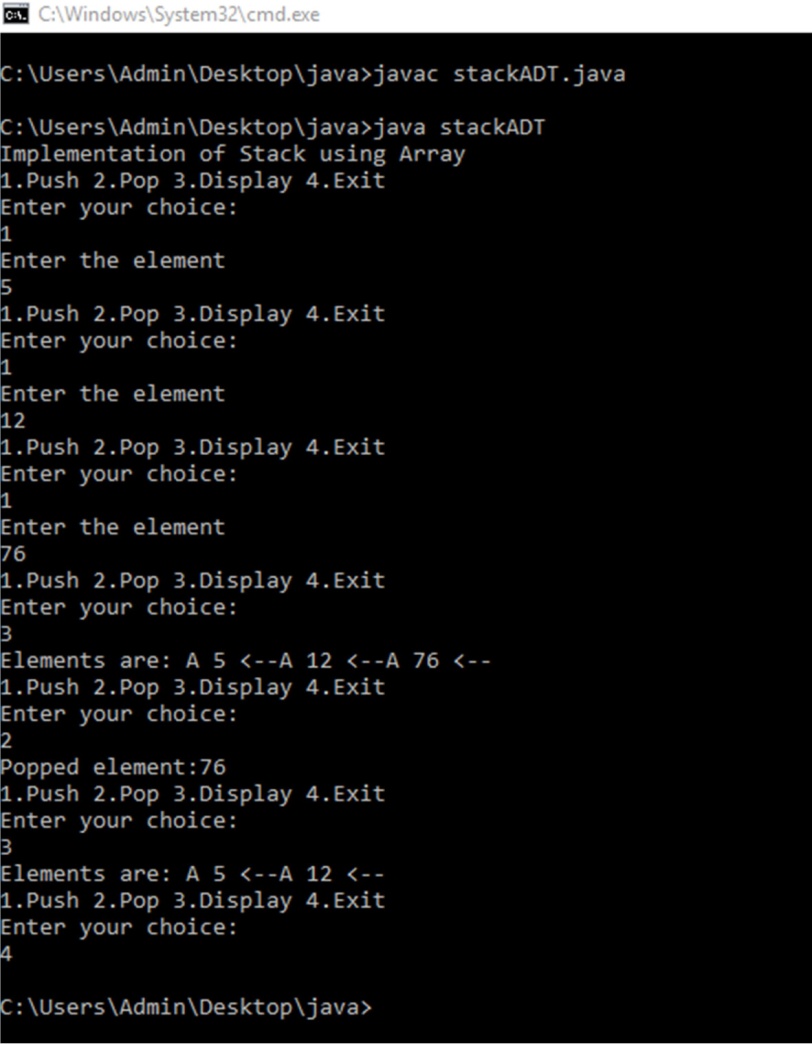
}

}

while(ch<5);

}

}

**Sample output:**

**Result:** Thus the design and implementation of ADT Stack using array has successfully executed.



16

**Ex. No.:5 Date: 10.09.20**

**STRING OPERATIONS**

**Aim** To write a program to perform string operations using ArrayList.

**Procedure:**

1. Start the program
2. Add the String as an object to List.
3. Get the choice from the user and do according to the choice
   1. Append-add at end
   2. Insert-add at particular index
   3. Search
   4. List all string starts with given letter.
4. Display the result
5. Stop the program.

**Source code:**

import java.util.Scanner; import java.util.ArrayList; class soper {

private static ArrayList<String> var = new ArrayList<>(); private static Scanner scan = new Scanner(System.in); private static void append() {

System.out.print("Enter the string to append: "); String str = scan.next();

var.add(str);

System.out.println("Appended successfully!!!!!!");

}

private static void add() {

{

System.out.print("Enter the String to add: "); String str = scan.next();

System.out.print("Enter the index of the string: "); int i = scan.nextInt();

var.add(i,str);

System.out.println("Added successfuly!!!!!!");

}

}

private static void search() {

System.out.print("Enter the string to search for: "); String str = scan.next();

int no = var.indexOf(str); System.out.println("Found in index " + no);

}

private static void find() {

System.out.print("Enter the first letter: "); String ch = scan.next();

//System.out.print(ch); for(String a : var) {

//System.out.print(" " + a.substring(0,1));

if(a.substring(0,ch.length()).equals(ch)) System.out.println(a);

}

}

private static void display() { for(String a : var)

System.out.println(a);

}

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

try

{

do {

System.out.println("\n\n\nMAIN MENU\n----------

\n1.Append\n2.add\n3.search\n4.find\n5.display\n6.exit");

System.out.print("Enter your choice: "); ch = scan.nextInt();

switch(ch) {

case 1:

}

}while(true);

}

case 2:

case 3:

case 4:

case 5:

case 6:

default:

append(); break;

add(); break;

search(); break;

find(); break;

display(); break;

System.exit(0); break;

System.out.println("Invalid Input"); break;

catch(Exception e)

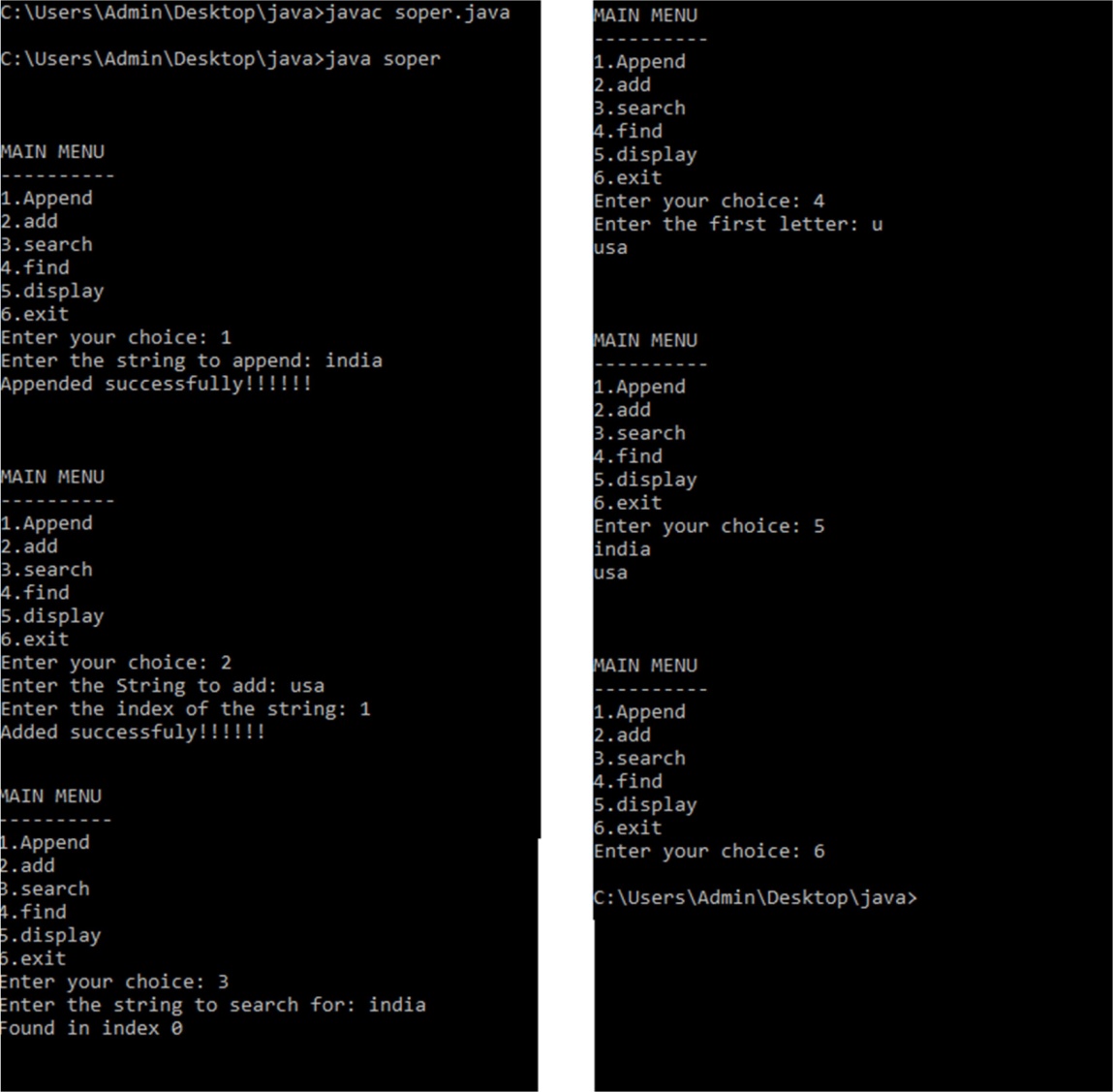
{

System.out.println(e);

}

}

}

Sample output:

**Result:**

Thus the implementation of string operations using array list has been successfully executed.



19

**Ex. No.:6 Date: 17.09.20**

**ABSTRACT CLASS**

**Aim**

To write a Java Program to create an abstract class named Shape and provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape.

**Procedure:**

1. Start the program
2. Define the abstract class shape.
3. Define the class Rectangle with PrintArea() method that extends(makes use of) Shape.
4. Define the class Triangle with PrintArea() method that extends(makes use of) Shape.

S. Define the class Circle with PrintArea() method that extends(makes use off Shape.

1. Print the area of the Rectang1e,Triang1e and Circle
2. Stop the Program.

**Source code:**

import java.io.\*; abstract class shape

{

int a=3,b=4,r=2;

abstract public void print\_area();

}

class rectangle extends shape

{

public double area\_rect; public void print\_area()

{

area\_rect=a\*b;

System.out.println("the area of rectangle is."+area\_rect);

}

}

class triangle extends shape

{

public double area\_tri; public void print\_area()

{

20

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

System.out.println("the area of triangle is."+area\_tri);

}

}

class circle extends shape

{

public double area\_cir; public void print\_area()

{

area\_cir=3.14\*r\*r;

System.out.println("the area of circle is."+area\_cir);

}

}

class eg1

{

public static void main(String args[])

{

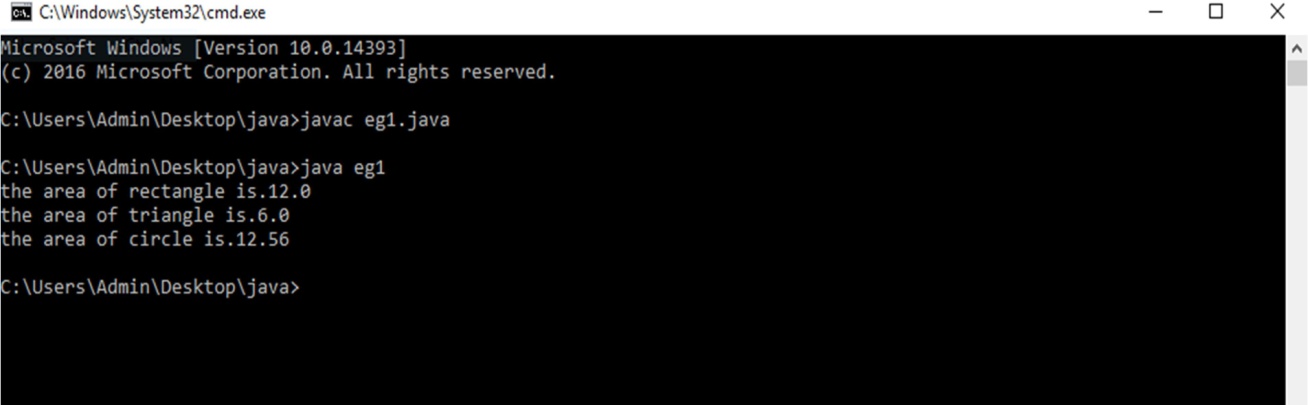
rectangle s1=new rectangle(); s1.print\_area();

triangle s2=new triangle(); s2.print\_area();

circle s3=new circle(); s3.print\_area();

}

}

**Sample output:**

**Result:**

Thus the design and implementation of Abstract class has been successfully executed.



21

# Ex. No 7 EXCEPTION HANDLING Date: 24.09.20

**Aim:** To write a Java program to implement user defined exception handling.

**Procedure:**

1. Start the program
2. Define the exception for getting a number from the user.
3. If the number is positive print the number as such.
4. If the number is negative throw the exception to the user as ‘Number must be positive’.
5. Stop the Program.

**Source code:**

import java.io.\*;

import java.lang.Exception;

class myownException extends Exception

{

myownException(String msg)

{

super(msg);

}

}

class employeetest

{

static void employeeAge(int age) throws myownException

{

if (age<0)

{

}

else

{

}

}

throw new myownException("Age can't be less than Zero");

System.out.println("Input is valid");

public static void main(String args[])

{

try

{

employeeAge(-2);

}

catch(myownException e)

{

e.printStackTrace();

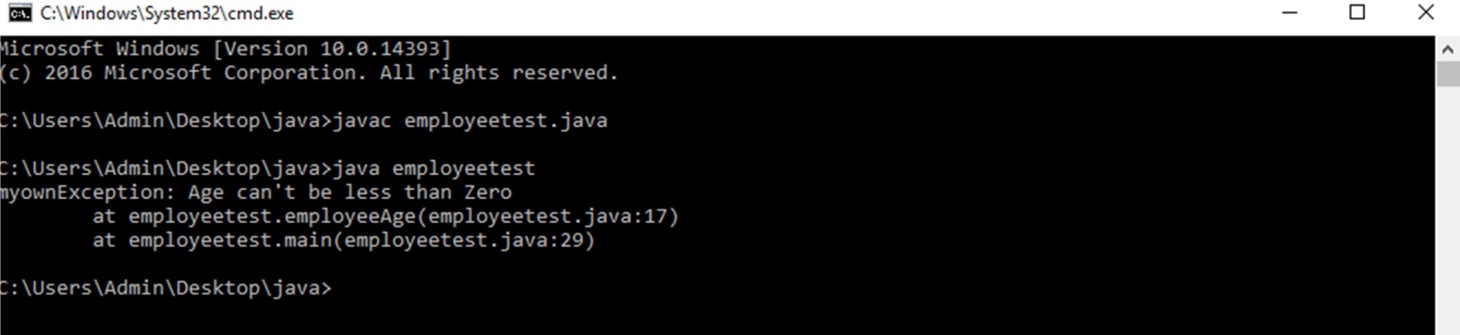
22

}

}

}

**Sample output:**



**Result:**

Thus the user defined exception has been successfully implemented.



23

**Ex. No.: 8 Date: 01.10.20**

**FILE INFORMATION**

**Aim**

To 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.

**Procedure:**

1. Start the program
2. Read the filename from the user.
3. Use getName() Method to display the filename.
4. Use getPath() Method to display the path of the file.
5. Use getParent() Method to display its parent’s information.
6. Use exists() Method to display whether the file exist or not
7. Use isFile() and isDirectory() Methods to display whether the file is file or directory.
8. Use canRead() and canWrite() methods to display whether the file is readable or writable.
9. Use length() method to display the size of the file.
10. Use isHiddden() Method to display whether the file is hidden or not.

**Source code:**

import java.io.File; import java.util.Scanner; class filedemo

{

public static void main(String args[])

{

System.out.println("Enter the File name"); Scanner input=new Scanner(System.in); String s=input.nextLine();

File f1=new File(s);

System.out.println("File name"+f1.getName()); System.out.println("Path "+f1.getPath()); System.out.println("Absolute Path"+f1.getAbsolutePath());

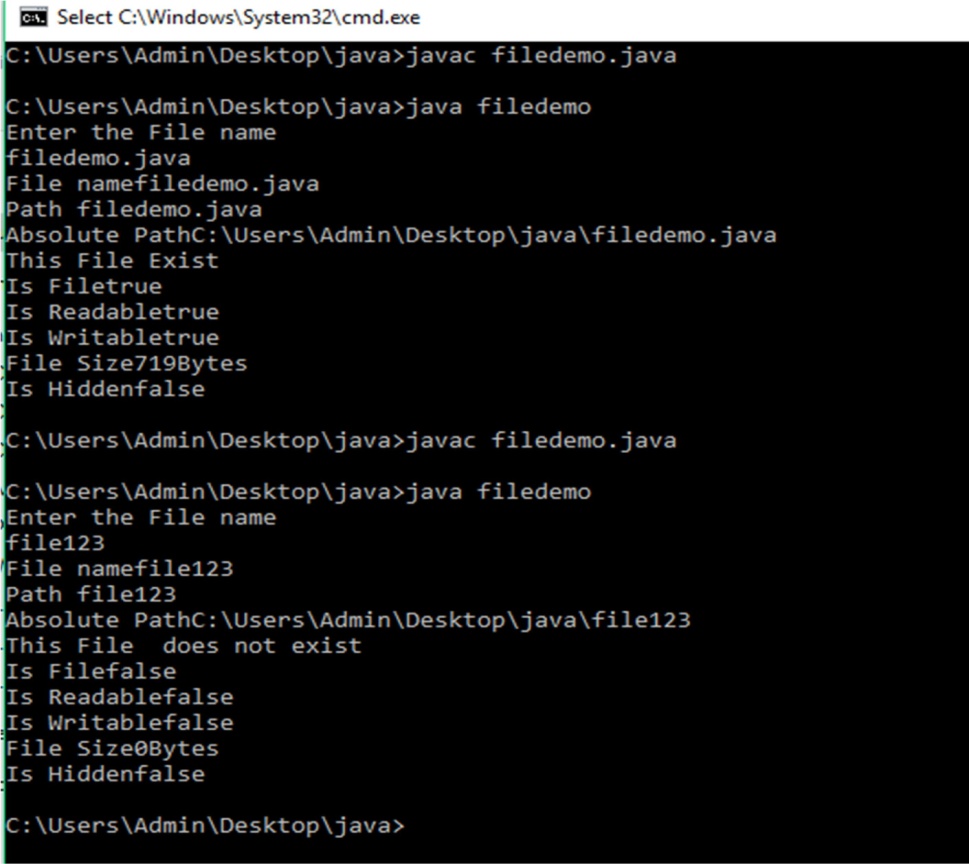
System.out.println("This File "+(f1.exists()?"Exists":"does not exist")); System.out.println("Is File"+f1.isFile());

24

System.out.println("Is Readable"+f1.canRead()); System.out.println("Is Writable"+f1.canWrite()); System.out.println("File Size"+f1.length()+"Bytes"); System.out.println("Is Hidden"+f1.isHidden());

}

}

**Sample output:**

## Result:

Thus the information of the file has been displayed successfully using various file methods.



25

**Ex. No.: 9 Date: 08.10.20**

**MULTITHREADING**

**Aim**

To write a java program that implements a multi-threaded application.

**Procedure:**

1. Start the program
2. Design the first thread that generates a random integer for every 1 second .
3. If the first thread value is even, design the second thread as the square of the number and then print it.
4. If the first thread value is odd, then third thread will print the value of cube of the number.
5. Stop the program.

**Source code:**

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 javathread

{

public static void main(String[] args)

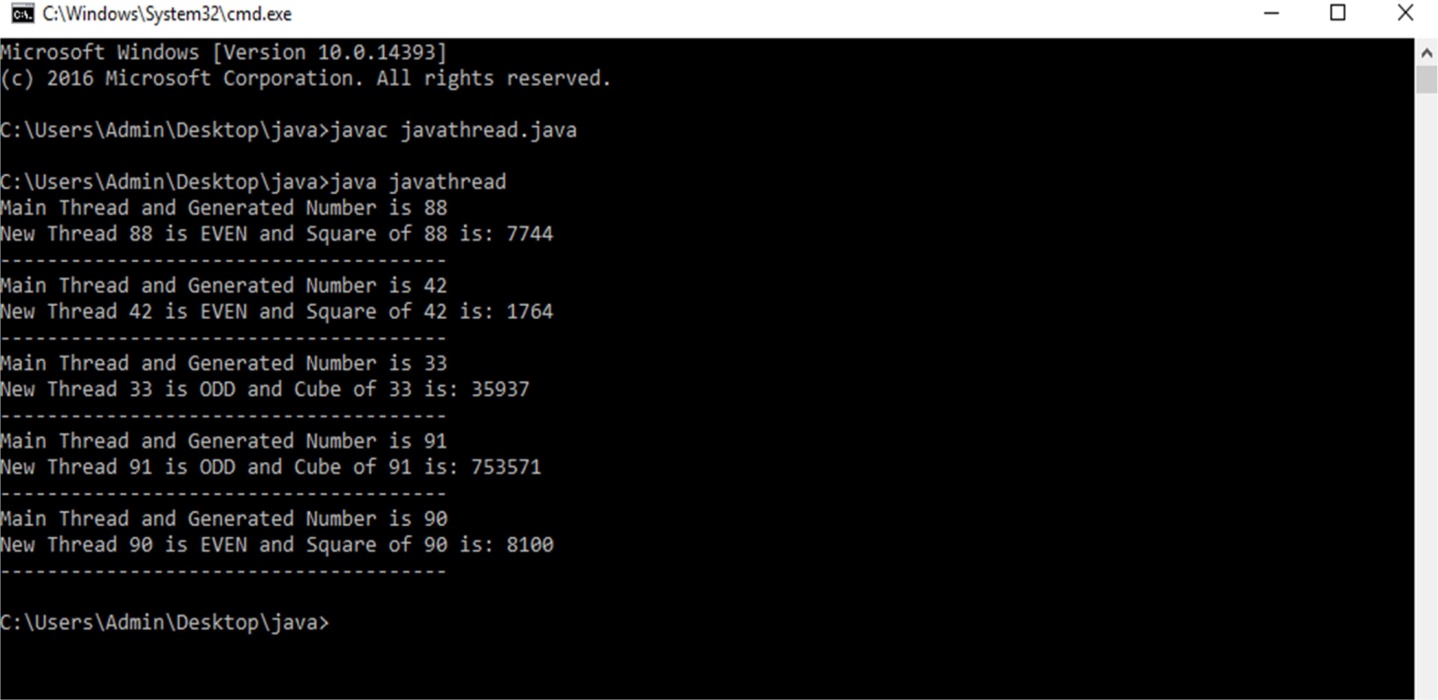
{

A a = new A(); a.start();

}

}

**Sample output:**



**Result:**

Thus the implementation of multithreading has been done using three threads.



28

**Ex.No.:10**

**Aim**

**GENERIC FUNCTION**

# Date: 15.10.20

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

**Procedure:**

1. Start the program
2. Define the array with the elements
3. Sets the first value in the array as the current maximum
4. Find the maximum value by comparing each elements of the array
5. Display the maximum value
6. Stop the program.

**Source code:**

import java.io.\*; class maxi

{

@SuppressWarnings({“unchecked”, “varargs”})

public static <T extends Comparable<T>> T max(T... elements) { T max = elements[0];

for (T element : elements) {

if (element.compareTo(max) > 0) { max = element;

}

}

return max;

}

public static void main(String[] args) {

System.out.println("Integer Max: " + max(Integer.valueOf(32), Integer.valueOf(56), Integer.valueOf(89), Integer.valueOf(3), Integer.valueOf(456), Integer.valueOf(78), Integer.valueOf(45))); System.out.println("Double Max: " + max(Double.valueOf(5.6), Double.valueOf(7.8), Double.valueOf(2.9), Double.valueOf(18.6), Double.valueOf(10.25), Double.valueOf(18.6001))); System.out.println("String Max: " + max("Dtrawberry", "Mango", "Apple", "Pomegranate", "Guava", "Blackberry", "Cherry", "Orange", "Date"));

}

}

29

**Sample output:**



**Result:**

Thus the implementation of generic function is achieved for finding the maximum value from the given type of elements.



30

**Ex. No.: 11 Date: 22.10.20**

**CALCULATOR**

**Aim**

To design a calculator using event-driven programming paradigm of Java for Decimal manipulations and Scientific manipulations.

## Procedure:

1. Start the program
2. Using the swing components design the buttons of the calculator
3. Use key events and key listener to listen the events of the calculator.
4. Do the necessary manipulations.
5. Stop the program.

## Source code:

import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

import javax.swing.event.\*;

public class calci 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, plus, min, div,mul,eq,sqrt; Container cont;

JPanel textPanel, buttonpanel; calci() {

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(10, 4, 10, 10)); boolean t = true;

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);

eq = new JButton("="); buttonpanel.add(eq); eq.addActionListener(this);

sqrt = new JButton("Sqrt"); buttonpanel.add(sqrt); sqrt.addActionListener(this);

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(); System.out.println(s);

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("+")) {

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("Sqrt")) {

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

} else {

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

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

}

}

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

tfield.setText(""); x = 0;

y = 0;

z = 0;

}

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); System.out.println("Z=1");

z = 1;

}

}

tfield.requestFocus();

}

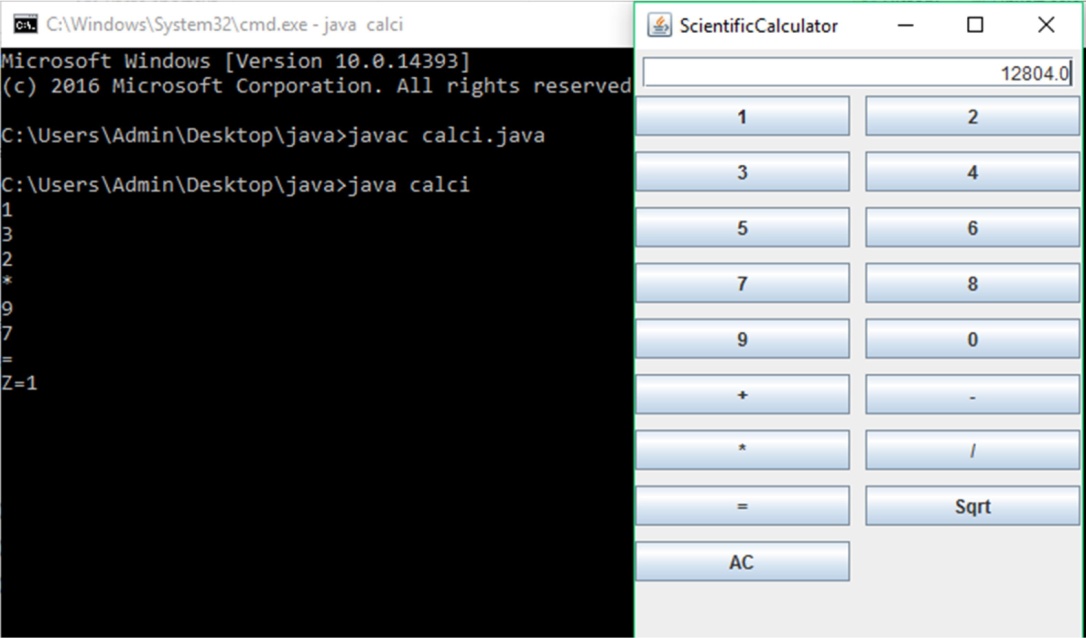
public static void main(String args[]) {

calci f = new calci(); f.setTitle("ScientificCalculator"); f.pack();

f.setVisible(true);

}

}

**Sample Output:**

**Result:**

Thus the calculator is simulated by applying the concept of event handling in java.



38

**Ex no. 12 TEXT EDITOR USING STRING Date: 29.10.20**

## Aim:

To design a text editor using java swing

## Procedure:

1. To create a simple text editor in Java Swing we will use a JTextArea, a JMenuBar and add JMenu to it and we will add JMenuItems. All the menu items will have actionListener to detect any action.
2. to create File menu

**open**: this menu item is used to open a file

**save**: this menu item is used to save a file

**print** : this menu item is used to print the components of the text area

**new** : this menu item is used to create a new blank file 3.To create Edit menu

**cut**: this menu item is to cut the selected area and copy it to clipboard

**copy**: this menu item is to copy the selected area to the clipboard

**paste** : this menu item is to paste the text from the clipboard to the text area 4.To Close this button closes the frame

## Application:

// Java Program to create a text editor using java import java.awt.\*;

import javax.swing.\*; import java.io.\*;

import java.awt.event.\*;

import javax.swing.plaf.metal.\*; import javax.swing.text.\*;

class editor extends JFrame implements ActionListener {

// Text component JTextArea t;

// Frame JFrame f;

// Constructor editor()

{

// Create a frame

f = new JFrame("editor"); try {

// Set metl look and feel UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");

// Set theme to ocean MetalLookAndFeel.setCurrentTheme(new OceanTheme());

}

catch (Exception e) {

}

// Text component

t = new JTextArea();

// Create a menubar

JMenuBar mb = new JMenuBar();

// Create amenu for menu

JMenu m1 = new JMenu("File");

// Create menu items

JMenuItem mi1 = new JMenuItem("New"); JMenuItem mi2 = new JMenuItem("Open"); JMenuItem mi3 = new JMenuItem("Save"); JMenuItem mi9 = new JMenuItem("Print");

// Add action listener mi1.addActionListener(this); mi2.addActionListener(this); mi3.addActionListener(this); mi9.addActionListener(this); m1.add(mi1);

m1.add(mi2);

m1.add(mi3);

m1.add(mi9);

// Create amenu for menu

JMenu m2 = new JMenu("Edit");

// Create menu items

JMenuItem mi4 = new JMenuItem("cut"); JMenuItem mi5 = new JMenuItem("copy"); JMenuItem mi6 = new JMenuItem("paste");

// Add action listener mi4.addActionListener(this);

mi5.addActionListener(this); mi6.addActionListener(this); m2.add(mi4);

m2.add(mi5);

m2.add(mi6);

JMenuItem mc = new JMenuItem("close"); mc.addActionListener(this);

mb.add(m1);

mb.add(m2);

mb.add(mc); f.setJMenuBar(mb); f.add(t); f.setSize(500, 500); f.setVisible(true);

}

// If a button is pressed

public void actionPerformed(ActionEvent e)

{

String s = e.getActionCommand();

if (s.equals("cut")) { t.cut();

}

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

t.copy();

}

else if (s.equals("paste")) { t.paste();

}

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

// Create an object of JFileChooser class JFileChooser j = new JFileChooser("f:");

// Invoke the showsSaveDialog function to show the save dialog int r = j.showSaveDialog(null);

if (r == JFileChooser.APPROVE\_OPTION) {

// Set the label to the path of the selected directory

File fi = new File(j.getSelectedFile().getAbsolutePath()); try {

// Create a file writer

FileWriter wr = new FileWriter(fi, false);

// Create buffered writer to write BufferedWriter w = new BufferedWriter(wr);

// Write w.write(t.getText()); w.flush();

w.close();

}

catch (Exception evt) { JOptionPane.showMessageDialog(f, evt.getMessage());

}

}

// If the user cancelled the operation else

JOptionPane.showMessageDialog(f, "the user cancelled the operation");

}

else if (s.equals("Print")) { try {

// print the file t.print();

}

catch (Exception evt) { JOptionPane.showMessageDialog(f, evt.getMessage());

}

}

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

// Create an object of JFileChooser class JFileChooser j = new JFileChooser("f:");

// Invoke the showsOpenDialog function to show the save dialog int r = j.showOpenDialog(null);

// If the user selects a file

if (r == JFileChooser.APPROVE\_OPTION) {

// Set the label to the path of the selected directory

File fi = new File(j.getSelectedFile().getAbsolutePath()); try {

// String

String s1 = "", sl = "";

// File reader

FileReader fr = new FileReader(fi);

// Buffered reader

BufferedReader br = new BufferedReader(fr);

// Initilize sl

sl = br.readLine();

// Take the input from the file

while ((s1 = br.readLine()) != null) { sl = sl + "\n" + s1;

}

// Set the text t.setText(sl);

}

catch (Exception evt) { JOptionPane.showMessageDialog(f, evt.getMessage());

}

}

// If the user cancelled the operation else

JOptionPane.showMessageDialog(f, "the user cancelled the operation");

}

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

t.setText("");

}

else if (s.equals("close")) { f.setVisible(false);

}

}

// Main class

public static void main(String args[])

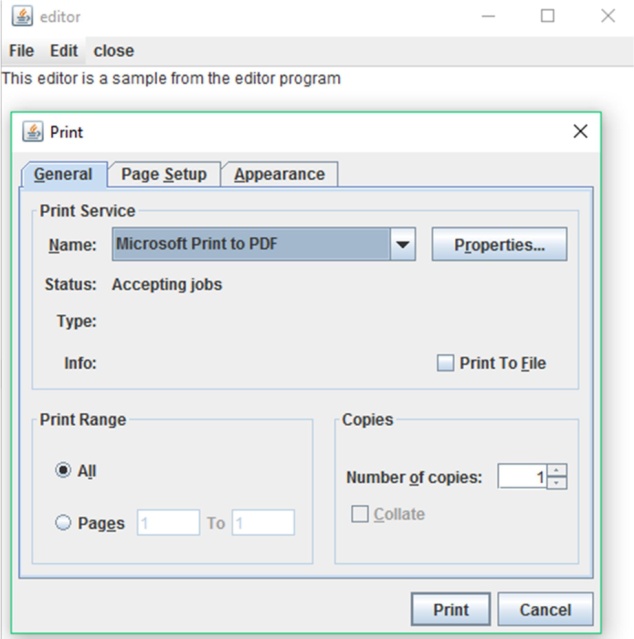
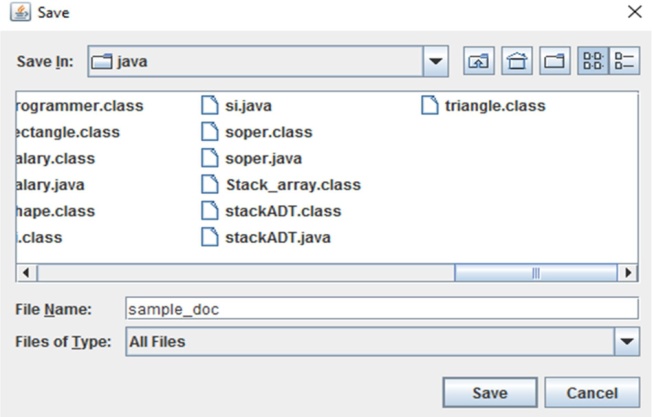
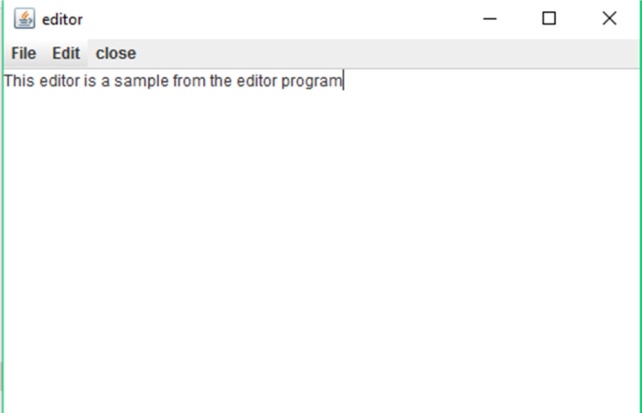
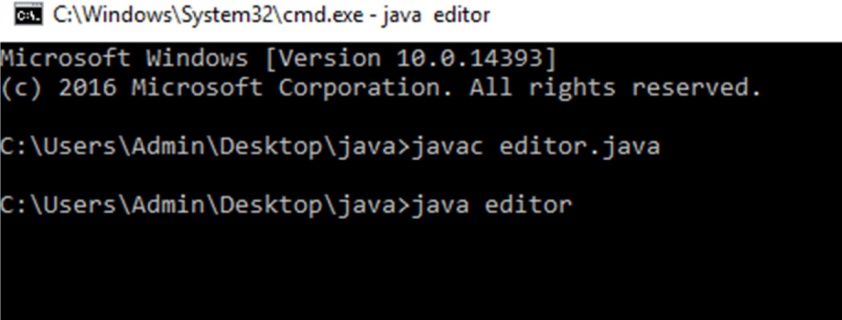
{

editor e = new editor();

}

}

**Sample output:**



## Result:

Thus the application of text editor has been successfully completed and executed.



43