**CS8383 OBJECT ORIENTED PROGRAMMING LABORATORY**

**II YEAR/ III SEM (July – Oct 2018)**

**BATCH: 2017-2021**

**LAB MANUAL**

|  |
| --- |
| **CS8383 OBJECT ORIENTED PROGRAMMING LABORATORY L T P C**  **0 0 4 2** |
| **OBJECTIVES** |
| To build software development skills using java programming for real-world applications. |
| To understand and apply the concepts of classes, packages, interfaces, arraylist, exception handling and file processing. |
| To develop applications using generic programming and event handling. |
| **LIST OF 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: |
| 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 |
| 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 |
| 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. |
| 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. |
| 4. Design a Java interface for ADT Stack. Implement this interface using array. Provide necessary exception handling in both the implementations. |
| 5. Write a program to perform string operations using Array List. 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 |
| 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. |
| 7. Write a Java program to implement user defined exception handling. |
| 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. |
| 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. |
| 10. Write a java program to find the maximum value from the given type of elements using a generic function. |
| 11. Design a calculator using event-driven programming paradigm of Java with the following options. |
| a) Decimal manipulations |
| b) Scientific manipulations |
| 12. Develop a mini project for any application using Java concepts. |
| **TOTAL : 60 PERIODS**  **OUTCOMES** |
| Upon completion of the course, the students will be able to |
| Develop and implement Java programs for simple applications that make use of classes, packages and interfaces. |
| Develop and implement Java programs with arraylist, exception handling and multithreading. |
| Design applications using file processing, generic programming and event handling. |

**EX:NO: 1** **ELECTRICITY BILL**

**AIM:**

To develop a java application to generate electricity bill.

**ALGORITHM:**

**Step 1**: Start the process.

**Step2**: Create a class elec\_bill and declare the necessary variables and define necessary

Methods.

**Step 3**: Create a method unit to calculate unit.

**Step 4**: Create a method domestic to calculate domestic bill.

**Step 5**: Create a method commercial to calculate commercial bill.

**Step 6**: : Create a method to display consumer details .

**Step 7**: Read consumer details, previous month reading, current month reading from the

User in main class using Scanner.

**Step 8**: Call the method to calculate the bill and display the consumer details and their corresponding electricity bill.

**Step 9**: Stop the process.

**PROGRAM**

import java.util.Scanner;

class elec\_bill{

public String con\_no,con\_name;

public float pmr,cmr,s;

void unit()

{

s = cmr-pmr;

}

void domestic()

{

if(s<=100)

System.out.println("Your bill is " +s\*1);

else if(s>=101 && s<=200)

System.out.println("Your bill is " +s\*2.50);

else if(s>=201 && s<=500)

System.out.println("Your bill is " +s\*4);

else System.out.println("Your bill is " +s\*6);

}

void commercial()

{

if(s<=100)

System.out.println("Your bill is " +s\*2);

else if(s>=101 && s<=200)

System.out.println("Your bill is " +s\*4.50);

else if(s>=201 && s<=500)

System.out.println("Your bill is " +s\*6);

else System.out.println("Your bill is " +s\*7);

}

void con\_details()

{

System.out.println("Consumer number-" +con\_no);

System.out.println("Consumer name-" +con\_name);

}

}

class eb{

public static void main(String args[]){

elec\_bill bill = new elec\_bill();

int i;

Scanner ch = new Scanner(System.in);

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

bill.con\_no = ch.nextLine();

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

bill.con\_name = ch.nextLine();

System.out.println("Enter the prev month reading: ");

bill.pmr = ch.nextFloat();

System.out.println("Enter the current month reading: ");

bill.cmr = ch.nextFloat();

bill.con\_details();

bill.unit();

i = bill.con\_no.length();

if(i<=5) bill.domestic();

else bill.commercial();

}

}

# OUTPUT

Enter the number:

12345

Enter the name:

Ash

Enter the prev month reading:

100

Enter the current month reading:

300

Consumer number-12345

Consumer name-Ash

Your bill is 500.0

## RESULT

Thus, a java application to generate electricity bill was written and the output was verified.

**EX:NO: 2** **PACKAGE**

**AIM:**

Develop a java application to implement currency converter,distance converter,time converter using packages.

**ALGORITHM:**

**Step1**:start the process.

**Step 2**:Create a package containing the necessary classes.

**Step 3**: Inside the class, define methods to perform different currency conversion,distance conversion and time conversion.

**Step 4**: Import the package in main class.

**Step 5**: Use switch case to choose between currency,distance,and time converter.

**Step 6**: Use a nested switch to choose between the options in currency or distance or time converter.

**Step 7:** Give the corresponding method call inside the switch case itself.

**Step 8:** Display the results.

**Step 9:** Stop the process.

# **PROGRAM**

import java.util.Scanner;

import converter.\*;

public class conv {

public static void main(String args[]) {

mon\_con conv1 = new mon\_con();

dist\_con conv2 = new dist\_con();

time\_con conv3 = new time\_con();

int ch;

char ty;

System.out.println("1.Money Converter\n2.Distance Converter\n3.Time Converter");

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

Scanner sc = new Scanner(System.in);

ch = sc.nextInt();

switch(ch)

{

case 1:

System.out.println("a.Dollar to rupee\nb.Rupee to dollar\nc.Euro to rupee\nd.Rupee to euro\ne.Yen to rupee\nf.Rupee to yen\nEnter your choice: ");

ty = sc.next().charAt(0);

switch(ty)

{

case 'a':

conv1.dol\_to\_rup();

break;

case 'b':

conv1.rup\_to\_dol();

break;

case 'c':

conv1.eur\_to\_rup();

break;

case 'd':

conv1.rup\_to\_eur();

break;

case 'e':

conv1.yen\_to\_rup();

break;

case 'f':

conv1.rup\_to\_yen();

break;

}

break;

case 2:

System.out.println("a.Meter to km\nb.Km to meter\nc.Miles to km\nd.Km to miles\nEnter your choice: ");

ty = sc.next().charAt(0);

switch(ty)

{

case 'a':

conv2.met\_to\_km();

break;

case 'b':

conv2.km\_to\_met();

break;

case 'c':

conv2.mil\_to\_km();

break;

case 'd':

conv2.km\_to\_mil();

break;

}

break;

case 3:

System.out.println("a.Hours to minutes\nb.Minutes to hours\nc.Hours to Seconds\nd.Seconds to hours\nEnter your choice: ");

ty = sc.next().charAt(0);

switch(ty)

{

case 'a':

conv3.hrs\_to\_min();

break;

case 'b':

conv3.min\_to\_hrs();

break;

case 'c':

conv3.hrs\_to\_sec();

break;

case 'd':

conv3.sec\_to\_hrs();

break;

}

break;

}

}

}

package converter;

import java.util.Scanner;

public class mon\_con {

float rupee,dollar,yen;

public void dol\_to\_rup()

{

float i,rup;

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

Scanner cs = new Scanner(System.in);

i = cs.nextFloat();

rup = (float) (i\*68.74);

System.out.println("Rs. " +rup);

cs.close();

}

public void rup\_to\_dol()

{

float i,dol;

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

Scanner cs = new Scanner(System.in);

i = cs.nextFloat();

dol = (float) (i/68.74);;

System.out.println("$ " +dol);

cs.close();

}

public void eur\_to\_rup()

{

float i,rup;

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

Scanner cs = new Scanner(System.in);

i = cs.nextFloat();

rup = (float) (i\*80.64);

System.out.println("Rs. " +rup);

cs.close();

}

public void rup\_to\_eur()

{

float i,eur;

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

Scanner cs = new Scanner(System.in);

i = cs.nextFloat();

eur = (float) (i/80.64);

System.out.println("Rs. " +eur);

cs.close();

}

public void yen\_to\_rup()

{

float rup,i;

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

Scanner cs = new Scanner(System.in);

i = cs.nextFloat();

rup =(float) (i\*0.62);

System.out.println("Rs. " +rup);

cs.close();

}

public void rup\_to\_yen()

{

float yen,i;

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

Scanner cs = new Scanner(System.in);

i = cs.nextFloat();

yen = (float) (i/0.62);

System.out.println("Rs. " +yen);

cs.close();

}

}

package converter;

import java.util.Scanner;

public class dist\_con {

float km,mile,metre;

public void met\_to\_km()

{

float e,km;

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

Scanner sa = new Scanner(System.in);

e = sa.nextFloat();

km = e/1000;

System.out.println("Km- " +km);

sa.close();

}

public void km\_to\_met()

{

float e,met;

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

Scanner sa = new Scanner(System.in);

e = sa.nextFloat();

met = e\*1000;

System.out.println("Metre- " +met);

sa.close();

}

public void mil\_to\_km()

{

float e,km;

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

Scanner sa = new Scanner(System.in);

e = sa.nextFloat();

km = (float) (e\*1.609);

System.out.println("Km- " +km);

sa.close();

}

public void km\_to\_mil()

{

float e,mile;

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

Scanner sa = new Scanner(System.in);

e = sa.nextFloat();

mile = (float) (e/1.609);

System.out.println("Mile- " +mile);

sa.close();

}

}

package converter;

import java.util.Scanner;

public class time\_con {

float hour,min,sec;

public void hrs\_to\_min()

{

float k,min;

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

Scanner se = new Scanner(System.in);

k = se.nextFloat();

min = k\*60;

System.out.println("Min- " +min);

se.close();

}

public void min\_to\_hrs()

{

float k,hrs;

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

Scanner se = new Scanner(System.in);

k = se.nextFloat();

hrs = k/60;

System.out.println("Hour- " +hrs);

se.close();

}

public void hrs\_to\_sec()

{

float k,sec;

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

Scanner se = new Scanner(System.in);

k = se.nextFloat();

sec = k\*3600;

System.out.println("Sec- " +sec);

se.close();

}

public void sec\_to\_hrs()

{

float k,hrs;

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

Scanner se = new Scanner(System.in);

k = se.nextFloat();

hrs = k/3600;

System.out.println("Hrs- " +hrs);

se.close();

}

}

# OUTPUT

1.Money Converter

2.Distance Converter

3.Time Converter

Enter your choice:

1

a.Dollar to rupee

b.Rupee to dollar

c.Euro to rupee

d.Rupee to euro

e.Yen to rupee

f.Rupee to yen

Enter your choice:

a

Enter the money:

10

Rs. 687.4

## RESULT

Thus, a java application to implement currency converter, money converter, time converter was created using packages and executed.

**EX:NO: 3 PAY SLIP USING INHERITANCE**

**AIM:**

To write a java program to generate pay slip for employees with their gross and net salary.

**ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Define a class containing employee details and necessary methods.

**Step 3:** Define a method to calculate salary.

**Step 4:** Define a method to display the pay-salary.

**Step 5:** Create a subclasses programmer, assistant professor, associate professor and professor and declare necessary variables.

**Step 6:** Create a main class and invoke object to subclass created above.

**Step 7:** Use switch case to choose between the subclasses.

**Step 8:** Call the necessary methods on the classes.

**Step 9:** Display the payslip.

**Step 10:** Stop the process.

# **PROGRAM**

import java.util.Scanner;

class Employee{

String Emp\_name;

int Emp\_id;

String Address;

String Mail\_address;

long Mobile\_no;

public void sal\_calc(float bp)

{

float gross,net,da,hra,pf,cf;

da = (float)0.97\*bp;

hra = (float)0.1\*bp;

pf = (float)0.12\*bp;

cf = (float)0.001\*bp;

gross = bp + da + hra;

net = gross - (pf + cf);

System.out.println("The gross salary is: " +gross + "\nThe net salary is: " + net);

}

public void pay\_slip()

{

System.out.println("Employee name: " +Emp\_name + "\nEmployee id: " +Emp\_id + "\nEmployee address: " +Address + "\nEmployee Mail address: " +Mail\_address + "\nEmployee Mobile number: " +Mobile\_no);

}

}

class Programmer extends Employee{

float bp;

}

class Assistant\_professor extends Employee{

float bp;

}

class Associate\_professor extends Employee{

float bp;

}

class Professor extends Employee{

float bp;

}

public class salary\_slip{

public static void main(String args[]){

Programmer pay1 = new Programmer();

Assistant\_professor pay2 = new Assistant\_professor();

Associate\_professor pay3 = new Associate\_professor();

Professor pay4 = new Professor();

int ch;

float bp;

System.out.println("1.Programmer\n2.Assistant\_professor\n3.Associate\_professor\n4.Professor\nEnter your choice");

Scanner sc = new Scanner(System.in);

ch = sc.nextInt();

switch(ch)

{

case 1:

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

pay1.Emp\_name = sc.next();

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

pay1.Emp\_id = sc.nextInt();

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

pay1.Address = sc.next();

System.out.println("Enter your mobile number: ");

pay1.Mobile\_no = sc.nextLong();

System.out.println("Enter your mail address: ");

pay1.Mail\_address = sc.next();

bp = 100000;

pay1.pay\_slip();

pay1.sal\_calc(bp);

break;

case 2:

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

pay2.Emp\_name = sc.next();

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

pay2.Emp\_id = sc.nextInt();

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

pay2.Address = sc.next();

System.out.println("Enter your mobile number: ");

pay2.Mobile\_no = sc.nextLong();

System.out.println("Enter your mail address: ");

pay2.Mail\_address = sc.next();

bp = 75000;

pay2.pay\_slip();

pay2.sal\_calc(bp);

break;

case 3:

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

pay3.Emp\_name = sc.next();

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

pay3.Emp\_id = sc.nextInt();

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

pay3.Address = sc.next();

System.out.println("Enter your mobile number: ");

pay3.Mobile\_no = sc.nextLong();

System.out.println("Enter your mail address: ");

pay3.Mail\_address = sc.next();

bp = 70000;

pay3.pay\_slip();

pay3.sal\_calc(bp);

break;

case 4:

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

pay4.Emp\_name = sc.next();

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

pay4.Emp\_id = sc.nextInt();

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

pay4.Address = sc.next();

System.out.println("Enter your mobile number: ");

pay4.Mobile\_no = sc.nextLong();

System.out.println("Enter your mail address: ");

pay4.Mail\_address = sc.next();

bp = 80000;

pay4.pay\_slip();

pay4.sal\_calc(bp);

break;

}

}

}

# OUTPUT

1.Programmer

2.Assistant\_professor

3.Associate\_professor

4.Professor

Enter your choice

1

Enter your name:

Ash

Enter your id:

112

Enter your address:

Trichy

Enter your mobile number:

7777777777

Enter your mail address:

ash@gmail.com

Employee name: Ash

Employee id: 112

Employee address: Trichy

Employee Mail address: ash@gmail.com

Employee Mobile number: 7777777777

The gross salary is: 207000.0

The net salary is: 194900.0

## RESULT

Thus, a java application to generate pay slips for the employees with their gross and net salary was created and executed.

**EX:NO: 4 ADT STACK USING INTERFACE**

**AIM:**

To develop a java program to design a java interface for ADT stack and provide necessary exception handling.

**ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Create a interface containing push and pop method.

**Step 3:** Implement the interface by defining class.

**Step 4:** Define the methods give inside the interface.

**Step 5:** Create the main class.

**Step 6:** Create the objects for the class which implements interface.

**Step 7:** Use exception handling techniques in the block containing methods calls for push and pop.

**Step 8:** Read the index of the stack whose data is to be pointed.

**Step 9:** Close the Exception handling class

**Step 10:** Display the result.

**Step 11:** Stop the process.

# **PROGRAM**

import java.util.Scanner;

interface IntStack{

void push(int item);

int pop();

}

class FixedStack implements IntStack{

public int stck[];

public int top;

FixedStack(int size){

stck = new int[size];

top=-1;

}

public void push(int item){

if(top>=stck.length-1)

System.out.println("Stack is full.");

else

stck[++top] = item;

}

public int pop(){

if(top<0){

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

return 0;

}

else

return stck[top--];

}

}

class Adt{

public static void main(String args[]){

FixedStack mystack1 = new FixedStack(10);

FixedStack mystack2 = new FixedStack(15);

int i,j,elem1,elem2;

try{

for(i=0;i<10;i++) mystack1.push(i);

for(j=0;j<15;j++) mystack2.push(j);

Scanner sc = new Scanner(System.in);

System.out.println("Enter the index of the stack 1: ");

elem1 = sc.nextInt();

System.out.println("The element in the " + elem1 + " position is " + mystack1.stck[elem1]);

System.out.println("Enter the index of the stack 2: ");

elem2 = sc.nextInt();

System.out.println("The element in the " + elem2 + " position is " + mystack2.stck[elem2]);

}catch(ArrayIndexOutOfBoundsException e){

System.out.println("Index exceeded");

}

System.out.println("Stack in mystack1: ");

for(i=0;i<10;i++)

System.out.println(mystack1.pop());

System.out.println("Stack in mystack2: ");

for(j=0;j<15;j++)

System.out.println(mystack2.pop());

}

}

# OUTPUT

Enter the index of the stack 1:

1

The element in the 1 position is 1

Enter the index of the stack 2:

2

The element in the 2 position is 2

Stack in mystack1:

9

8

7

6

5

4

3

2

1

0

Stack in mystack2:

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

## **RESULT**

Thus, a java interface for ADT Stack was implemented using array and executed.

**EX:NO: 5 STRING OPERATION USING ARRAY LIST**

**AIM:**

To perform string operations like append, insert, search and list all string starts with given letter using array list.

**ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Create a class stringops.

**Step 3:** Create an empty array list.

**Step 4:** Delete necessary variables.

**Step 5:** Add an element to the array list.

**Step 6:** Read a string from the user to append to array list.

**Step 7:** Append the string to array list using add method.

**Step 8:** Read a string to insert into the array list and get the position to invert.

**Step 9:** Insert the string into array list using add method .

**Step 10:** Read a string from the user to search. Find the position of the search string using indexOf method.

**Step 11:** Create a string array.

**Step 12:** Read a character from the user to search for the tokens starting with that character.

**Step 13:** Display the results.

**Step 14:** Stop the process.

# **PROGRAM**

import java.util.\*;

class StringOps{

public static void main(String args[]){

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

String s1,s2,s3;

int loc,sizE,i;

char a;

al.add("Java");

System.out.println("Array list before string operations: " +al);

Scanner se = new Scanner(System.in);

//Append

System.out.println("Enter the string to be appended: ");

s1 = se.nextLine();

al.add(s1);

System.out.println("Array list after appending: " +al);

//Insertion

System.out.println("Enter the string to be inserted: ");

s2 = se.nextLine();

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

loc = se.nextInt();

al.add(loc,s2);

System.out.println("Array list after insertion: " +al);

//Search

System.out.println("Enter the string to be searched: ");

s3 = se.next();

System.out.println("The position of the searched string: " +al.indexOf(s3));

//Listing all strings starts with given letter

sizE = al.size();

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

a = se.next().charAt(0);

String[] arr = new String[al.size()];

for(i=0;i<sizE;i++)

{

arr[i] = al.get(i);

if(arr[i].charAt(0)==a) System.out.println(arr[i]+"\n");

}

se.close();

}

}

# **OUTPUT**

Array list before string operations: [Java]

Enter the string to be appended:

is awesome

Array list after appending: [Java, is awesome]

Enter the string to be inserted:

learning

Enter the position:

1

Array list after insertion: [Java, learning, is awesome]

Enter the string to be searched:

Java

The position of the searched string: 0

Enter the element:

l

learning

## RESULT

Thus, a program to perform string operations using ArrayList was written and executed.

**EX:NO: 6 ABSTRACT CLASS**

**AIM:**

To write a java program to print area of different shapes using abstract class.

**ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Create an abstract class shape.

**Step 3:** Declare the necessary variables.

**Step 4:** Create required constructors.

**Step 5**: Create separate classes for shapes like triangle, rectangle, circle which has only one method to print area.

**Step 6:** Make the function return the area of shapes.

**Step 7:** Create objects for rectangle, triangle, circle in main class.

**Step 8:** Read the dimensions from the user.

**Step 9:** Call the area function.

**Step 10:** Display the result.

**Step 11:** Stop the process.

# **PROGRAM**

//Area of shapes

import java.util.Scanner;

abstract class Shape{

int dim1,dim2;

Shape(int a,int b){

dim1=a;

dim2=b;

}

Shape(int a){

dim1=a;

}

abstract double area();

}

class Rectangle extends Shape{

Rectangle(int a,int b){

super(a,b);

}

double area(){

System.out.println("The area of the rectangle is: ");

return (double)(dim1\*dim2);

}

}

class Triangle extends Shape{

Triangle(int a,int b){

super(a,b);

}

double area(){

System.out.println("The area of the triangle is: ");

return (double)(dim1\*dim2)/2;

}

}

class Circle extends Shape{

Circle(int a){

super(a);

}

double area(){

System.out.println("The area of the circle is: ");

return (3.14\*dim1\*dim1);

}

}

class Area{

public static void main(String args[]){

Rectangle rec = new Rectangle(0,0);

Triangle tri = new Triangle(0,0);

Circle cir = new Circle(0);

Shape shaperef;

Scanner se = new Scanner(System.in);

System.out.println("Enter the first dimension for rectangle: ");

rec.dim1 = se.nextInt();

System.out.println("Enter the second dimension for rectangle: ");

rec.dim2=se.nextInt();

System.out.println("Enter the first dimension for triangle: ");

tri.dim1 = se.nextInt();

System.out.println("Enter the second dimension for triangle: ");

tri.dim2=se.nextInt();

System.out.println("Enter the dimension for circle: ");

cir.dim1 = se.nextInt();

shaperef = rec;

System.out.println(shaperef.area());

shaperef = tri;

System.out.println(shaperef.area());

shaperef = cir;

System.out.println(shaperef.area());

se.close();

}

}

# OUTPUT

Enter the first dimension for rectangle:

10

Enter the second dimension for rectangle:

20

Enter the first dimension for triangle:

30

Enter the second dimension for triangle:

10

Enter the dimension for circle:

10

The area of the rectangle is:

200.0

The area of the triangle is:

150.0

The area of the circle is:

314.0

# RESULT

Thus, a java program to create an abstract class to print the area of a shape was written and executed.

**EX:NO: 7 USER DEFINED EXCEPTION HANDLING**

**AIM:**

To write a java program to implement user defined exception handling.

**ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Create a user define exception.

**Step 3:** Use throws to throw an exception.

**Step 4:** Create a condition to check and throw the exception.

**Step** **5:** Declare the necessary variables.

**Step** **6:** Read the phone number from the user.

**Step 7:** If the number is not equal to 10 throw an exception displaying that it is an invalid number.

**Step 8:** Otherwise display that is a valid number.

**Step 9:** Stop the process.

# PROGRAM

import java.util.Scanner;

class InvalidPhoneNumberException extends Exception{

InvalidPhoneNumberException(String p){

super(p);

}

}

class UserDefException{

static void valid(int len) throws InvalidPhoneNumberException{

if(len!=10)

throw new InvalidPhoneNumberException("Phone number is not valid!");

else

System.out.println("Your phone number is valid");

}

public static void main(String args[]){

Scanner se = new Scanner(System.in);

String phoneNumber;

int len;

System.out.println("Enter your phone number: ");

phoneNumber = se.nextLine();

len = phoneNumber.length();

try{

valid(len);

}catch(InvalidPhoneNumberException e){System.out.println("Exception occured: "+e);

}

}

}

# OUTPUT

Enter your phone number:

123567890

Exception occured: InvalidPhoneNumberException: Phone number is not valid!

# RESULT

Thus, program to implement user defined exception handling was written and executed.

**EX:NO: 8 FILE HANDLING**

**AIM:**

To write a java program to implement the file handling program.

**ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Create a classes for file handling which has methods for different operations which are going to be performed on the file.

**Step 3:** In the file exists method check whether the file exists or not using file.exists() method .

**Step 4:** In the file readable method check whether the file exists or not using file.exists() method and if the file exist check whether the file is readable mode or not using file.canRead() method.

**Step 5:** In the file Writable method check whether the file exists or not using file.exists() method and if the file exist check whether the file is Writable mode or not using file.canWead() method.

**Step 6:** In the filetype method first it checks whether the file exists method check whether the file exists or not using file.exists() .if the file exist the file type method give the type of file by using file.getPath() method.

**Step 7:** In the fileLength method first it checks whether the file exists method check whether the file exists or not using file.exists() method. if file exist then it return the file length in bytes using file.length() method.

**Step 8:** Display the result.

**Step 9:** Stop the process.

# PROGRAM

import java.io.File;

import java.nio.file.\*;

import java.util.Scanner;

class FileHandlingg{

void fileExists(File file){

if(file.exists()) System.out.println("The file exists!");

else System.out.println("The file doesn't exist!");

}

void fileReadable(File file){

if(file.exists()) {

if (file.canRead()) System.out.println("The file is readable!");

else System.out.println("The file is not readable!");

}

else System.out.println("The file doesn\'t exist!");

}

void fileWritable(File file){

if (file.exists()) {

if (file.canWrite()) System.out.println("The file is writable!");

else System.out.println("The file not is writable!");

}

else System.out.println("The file doesn't exist!");

}

void fileType(File file) {

String fileString;

if (file.exists()) {

fileString = file.getPath();

String[] fileArray = fileString.split("[\\.]");

System.out.println("The file is of " +"."+fileArray[fileArray.length-1]+" type");

}

else System.out.println("The file doesn't exist!");

}

void fileLength(File file){

if (file.exists()) {

System.out.println("The length of the file in bytes: " + file.length());

}

else System.out.println("The file doesn't exist!");

}

}

public class FileHandler {

public static void main(String[] args) {

String fileName;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the file name along with its path: ");

fileName = sc.next();

Path path = Paths.get(fileName);

System.out.println(path);

File file = new File(path.toString());

FileHandlingg fileHandling = new FileHandlingg();

int ch;

System.out.println("1.Check whether the file exists\n2.Check whether the file readable\n3.Check whether the file is writable\n4.Check the type of the file\n5.Dispaly the length of the file\nEnter your choice: ");

ch = sc.nextInt();

switch(ch) {

case 1:

fileHandling.fileExists(file);

break;

case 2:

fileHandling.fileReadable(file);

break;

case 3:

fileHandling.fileWritable(file);

break;

case 4:

fileHandling.fileType(file);

break;

case 5:

fileHandling.fileLength(file);

break;

}

}

}

# OUTPUT

Enter the file name along with its path:

G:\Class\OOP\Multithreading.txt

G:\Class\OOP\Multithreading.txt

1.Check whether the file exists

2.Check whether the file readable

3.Check whether the file is writable

4.Check the type of the file

5.Dispaly the length of the file

Enter your choice:

1

The file exists!

## RESULT

Thus, a java program implementing different file operations was written and executed.

# **EX:NO: 9 MULTITHREADING**

**PROGRAM**

import java.lang.Math;

class RandomNumbeR{

int getrandom() {

double random1 = Math.random();

double random2 = random1 \* 10;

int random3 = (int) random2;

return random3;

}

}

class Square implements Runnable{

int num;

Thread t;

Square(String string){

num = Integer.parseInt(string);

t = new Thread(this);

t.start();

}

public void run() {

synchronized(t) {

int square = num\*num;

System.out.println("Square is: "+square);

}

}

}

class Cube implements Runnable{

int num1;

Thread t;

Cube(String string1){

num1 = Integer.parseInt(string1);

t = new Thread(this);

t.start();

}

public void run() {

synchronized(t) {

int cube = num1\*num1\*num1;

System.out.println("Cube is: " +cube);

}

}

}

public class Sync {

public static void main(String [] args){

RandomNumbeR randomNumbeR = new RandomNumbeR();

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

int r = randomNumbeR.getrandom();

String s = ""+r;

System.out.println("The random number is: "+s);

if(r % 2 == 0) {

Square square1 = new Square(s);

}

if(r % 2 != 0) {

Cube cube1 = new Cube(s);

}

try

{

Thread.sleep(1000);

}catch(InterruptedException e) {

System.out.println("Main thread interrupted");

}

}

}

}

# OUTPUT

The random number is: 6

Square is: 36

The random number is: 3

Cube is: 27

The random number is: 6

Square is: 36

The random number is: 9

Cube is: 729

The random number is: 1

Cube is: 1

## RESULT

Thus, a java program implementing multithreading which contains three threads performing different operations was written and executed.

**EX:NO: 10 GENERICS**

**AIM:**

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

**ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Create a generic class by passing the data types handled by that class as argument to if.

**Step 3:** Declare instance variables for the array of different types.

**Step 4:** Create a method to find the maximum in a given list.overload it for all the data types handled by the class .

**(I).** Use for loop to compare integer values.

**(II).** Use for loop and compareTo method to compare the string values

**(III).** Use for loop to compare double values

**(IV).** Use for loop and compareTo method character values

**Step 5**: Create an array of all data type in main method .

**Step 6:** Create an object for the generic class in main class.

**Step 7:** Call the method which compares and display the ,maximum value in the particular list by passing the arrays of different data types,one array at time.

**Step 8:** Display the result.

**Step 9:** Stop the process.

# PROGRAM

import java.lang.String;

class MaximumInList <Integer,String,Double,Character> {

int [] ob1;

String [] ob2;

double [] ob3;

char [] ob4;

MaximumInList(int[] obj1, String[] obj2, double[] obj3, char[] obj4) {

ob1 = obj1;

ob2 = obj2;

ob3 = obj3;

ob4 = obj4;

}

void maximum(int ob1[]) {

int max = ob1[0];

for (int i = 1; i < ob1.length; i++) {

if (ob1[i] > max) {

max = ob1[i];

}

}

System.out.println("The maximum integer in the list is: "+max);

}

void maximum(String ob2[]) {

String max = ob2[0];

for (int i = 1; i < ob2.length; i++) {

if (((java.lang.String) max).compareTo(ob2[i].toString()) < 0)

max = ob2[i];

}

System.out.println("The maximum string in the list is: "+max);

}

void maximum(double ob1[]) {

double max = ob3[0];

for (int i = 1; i < ob3.length; i++) {

if (ob3[i] > max) {

max = ob3[i];

}

}

System.out.println("The maximum double in the list is: "+max);

}

void maximum(char ob4[]) {

char max = ob4[0];

for (int i = 1; i < ob4.length; i++) {

if (((java.lang.Character)max).compareTo(ob4[i]) < 0)

max = ob4[i];

}

System.out.println("The maximum character in the list is: "+max);

}

}

public class Generics {

public static void main(String[] args) {

int [] Int = {6,3,8,2,1,7,9};

String [] Str = {"hello","hi","bye","sayonara","goodbyee"};

double [] Doub = {10.12,23,45.87,100.5,2,4.5};

char [] Char = {'a','z','q','b','g'};

MaximumInList<Integer,String,Double,Character> maxi = new MaximumInList<>(Int, Str, Doub, Char);

maxi.maximum(Int);

maxi.maximum(Str);

maxi.maximum(Doub);

maxi.maximum(Char);

}

}

# OUTPUT

The maximum integer in the list is: 9

The maximum string in the list is: sayonara

The maximum double in the list is: 100.5

The maximum character in the list is: z

## RESULT

Thus, a java program to find the maximum value from the given type of elements using a generic function was written and executed.

**EX:NO: 11 EVENT HANDLING**

**AIM:**

To implement a java program for design a calculator using event handlers.

**ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Create the class scientific calculator .Define and declare its variables.

**Step 3:** Using scientific calculator constructor create buttons that are in the scientific calculator.

**Step 4:** Using actionPerformed() method define the function that has to be done when the  corresponding buttons are pressed.

**Step 5:** In the main function create the objects for the class scientific calculator and then using  that objects set the title for the program as scientific calculator and then press the buttons in the calculator to get the results which you want.

**Step 6:** Display the result.

**Step 7:** Stop the process.

**PROGRAM**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.\*;

class Calculator extends JFrame {

private final Font BIGGER\_FONT = new Font("monspaced",Font.PLAIN, 20);

private JTextField textfield;

private boolean number = true;

private String equalOp = "=";

private CalculatorOp op = new CalculatorOp();

public Calculator() {

textfield = new JTextField("", 12);

textfield.setHorizontalAlignment(JTextField.RIGHT);

textfield.setFont(BIGGER\_FONT);

ActionListener numberListener = new NumberListener();

String buttonOrder = "1234567890 ";

JPanel buttonPanel = new JPanel();

buttonPanel.setLayout(new GridLayout(4, 4, 4, 4));

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

String key = buttonOrder.substring(i, i+1);

if (key.equals(" ")) {

buttonPanel.add(new JLabel(""));

} else {

JButton button = new JButton(key);

button.addActionListener(numberListener);

button.setFont(BIGGER\_FONT);

buttonPanel.add(button);

}

}

ActionListener operatorListener = new OperatorListener();

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(4, 4, 4, 4));

String[] opOrder = {"+", "-", "\*", "/","=","C","sin","cos","log"};

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

JButton button = new JButton(opOrder[i]);

button.addActionListener(operatorListener);

button.setFont(BIGGER\_FONT);

panel.add(button);

}

JPanel pan = new JPanel();

pan.setLayout(new BorderLayout(4, 4));

pan.add(textfield, BorderLayout.NORTH );

pan.add(buttonPanel , BorderLayout.CENTER);

pan.add(panel , BorderLayout.EAST);

this.setContentPane(pan);

this.pack();

this.setTitle("Calculator");

this.setResizable(false);

}

private void action() {

number = true;

textfield.setText("");

equalOp = "=";

op.setTotal("");

}

class OperatorListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

String displayText = textfield.getText();

if (e.getActionCommand().equals("sin"))

{

textfield.setText("" + Math.sin(Double.valueOf(displayText).doubleValue()));

}else

if (e.getActionCommand().equals("cos"))

{

textfield.setText("" + Math.cos(Double.valueOf(displayText).doubleValue()));

}

else

if (e.getActionCommand().equals("log"))

{

textfield.setText("" + Math.log(Double.valueOf(displayText).doubleValue()));

}

else if (e.getActionCommand().equals("C"))

{

textfield.setText("");

}

else

{

if (number)

{

action();

textfield.setText("");

}

else

{

number = true;

if (equalOp.equals("="))

{

op.setTotal(displayText);

}else

if (equalOp.equals("+"))

{

op.add(displayText);

}

else if (equalOp.equals("-"))

{

op.subtract(displayText);

}

else if (equalOp.equals("\*"))

{

op.multiply(displayText);

}

else if (equalOp.equals("/"))

{

op.divide(displayText);

}

textfield.setText("" + op.getTotalString());

equalOp = e.getActionCommand();

}

}

}

}

class NumberListener implements ActionListener {

public void actionPerformed(ActionEvent event) {

String digit = event.getActionCommand();

if (number) {

textfield.setText(digit);

number = false;

} else {

textfield.setText(textfield.getText() + digit);

}

}

}

public class CalculatorOp {

private int total;

public CalculatorOp() {

total = 0;

}

public String getTotalString() {

return ""+total;

}

public void setTotal(String n) {

total = convertToNumber(n);

}

public void add(String n) {

total += convertToNumber(n);

}

public void subtract(String n) {

total -= convertToNumber(n);

}

public void multiply(String n) {

total \*= convertToNumber(n);

}

public void divide(String n) {

total /= convertToNumber(n);

}

private int convertToNumber(String n) {

return Integer.parseInt(n);

}

}

}

class SwingCalculator {

public static void main(String[] args) {

JFrame frame = new Calculator();

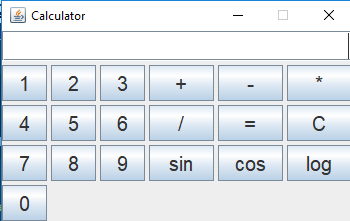
frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

}

}

## OUTPUT



# RESULT

Thus, a calculator was designed using event-driven paradigm of java with the decimal and scientific manipulations.