**Ex.No:01 IMPLEMENTATION OF ADDITION USING CLASSES AND OBJECTS**

**DATE:01/08/23**

**AIM:**

To implement Addition with methods, without methods, Classes and non- static mehods.

**SOURCE CODE:**

**ADDITION USING WITHOUT METHODS:**

class j1{

public static void main(String args[]){

int a=2;

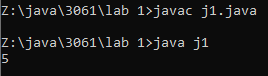
int b=3;

System.out.println(a+b);

}

}

**OUTPUT:**



**ADDITION USING WITH METHODS:**

class j2{

static void add(int a,int b){

return a+b;

}

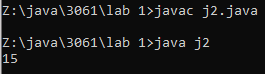
public static void main(String nums[]){

System.out.println(add(7,8));

}

}

**OUTPUT:**



**ADDITION USING CLASSES:**

class j3{

static int x=10;

static int y=10;

static int z=x+y;

//System.out.println(z);

public static void main(String n[]){

A1 obj = new A1();

System.out.println("Sum of 5 and 6 is "+obj.Add(5,6));

System.out.println(z);

}

}

class A1{

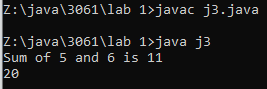
static int Add(int a,int b){

return a+b;

}

}

**OUTPUT:**



**ADDITION USING NON-STATIC METHODS:**

class j4

{

int add(int a, int b)

{

return a + b;

}

public static void main(String arg[])

{

j6 sum = new j6();

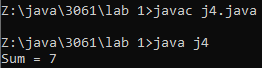
System.out.print("Sum = ");

System.out.print(sum.add(3,4));

}

}

**OUTPUT:**



**RESULT:**  
 Hence, the implementation of addition of various methods is executed successfully.

**Ex.No:02 IMPLEMENTATION OF ADDITION BY GETTING INPUT FROM USER**

**DATE:07/08/23**

**AIM:**

To implement addition using Run Time, Scanner, Bufferedreader in java.

**SOURCE CODE:**

**SUM OF TWO NUMBERS AS RUN TIME INPUT:**

class j5{

public static void main(String ip[]){

System.out.println("--Sum of two numbers as run Time Input--");

int a,b,c;

a=Integer.parseInt(ip[0]);

b=Integer.parseInt(ip[1]);

System.out.println("Input:" + a + " " + b);

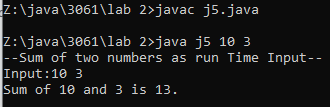
c=a+b;

System.out.println("Sum of " + a + " and " + b + " is " + c + ".");

}

}

**OUTPUT:**



**SUM OF TWO NUMBERS USING SCANNER:**

import java.util.Scanner;

class j6{

public static void main(String args[]){

Scanner p=new Scanner(System.in);

System.out.println("--Sum of two numbers Using Scanner--");

int n1,n2,n3;

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

n1=p.nextInt();

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

n2=p.nextInt();

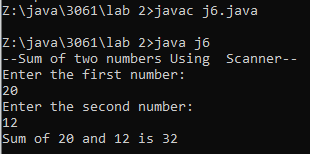
n3=n1+n2;

System.out.println("Sum of "+ n1 + " and " + n2 + " is " + n3);

}

}

**OUTPUT**



**SUM OF TWO NUMBERS USING BUFFEREDSTREAMREADER:**

import java.io.IOException;

import java.io.BufferedReader;

import java.io.InputStreamReader;

class j7{

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

System.out.println("--Sum of two numbers using BufferedStreamReader--");

InputStreamReader ir = new InputStreamReader(System.in);

BufferedReader br = new BufferedReader(ir);

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

int a = Integer.parseInt(br.readLine());

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

int b = Integer.parseInt(br.readLine());

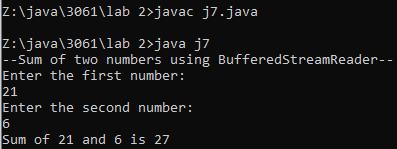
int c = a+b;

System.out.println("Sum of "+ a + " and " + b + " is " + c);

}

}

**OUTPUT**:



**RESULT**:

Thus, the addition is done by various methods executed successfully.

**Ex.No:03(a) IMPLEMENTATION OF OPERATORS**

**DATE:08/08/23**

**AIM:**

To write a program to implement basic operations using java.

**SOURCE CODE:**

import java.util.\*;

class operators

{

public static void main(String arg[])

{

int a,b;

Scanner ab=new Scanner(System.in);

System.out.print("Enter the first number:");

a=ab.nextInt();

System.out.print("Enter the second number:");

b=ab.nextInt();

System.out.println("Arithmetic Operators:");

System.out.println("The Sum is "+ (a+b));

System.out.println("The Difference is "+(a-b));

System.out.println("The Product is "+ (a\*b));

System.out.println("The Quotient is "+ (a/b));

System.out.println("The Remainder is "+ (a%b));

System.out.println("Increment Operator:");

System.out.println("Value of a before prefix increment is "+a);

System.out.println("Value of a while doing prefix increment is "+ (++a));

System.out.println("Value of a after prefix increment is "+a);

System.out.println("Value of a before postfix increment is "+a);

System.out.println("Value of a while doing postfix increment is "+ (a++));

System.out.println("Value of a after postfix increment is "+a);

System.out.println("Unary Operators:");

boolean c=true;

System.out.println("Value of c: "+c);

System.out.println("Value of NOT(c): "+ !(c));

System.out.println("Shift and bitwise operator:");

System.out.println("Shift left of a [a<<2]: "+ (a<<2));

System.out.println("Shift right of a [a>>2]: "+ (a>>2));

System.out.println("Bitwise AND [a&b]: " + (a&b));

System.out.println("Bitwise OR [a|b]: " + (a|b));

System.out.println("Bitwise NOT [~a]: " + (~a));

System.out.println("Logical Operators:");

System.out.println("Logical AND (a>b && a!=b): " + (a>b && a!=b));

System.out.println("Logical OR (a>b || a!=b): " + (a>b || a!=b));

System.out.println("Logical NOT [!(a>b)]: " + !(a>b));

System.out.println("Shorthand Assignment Operators:");

System.out.println("Value of a+=2: "+ (a+=2));

System.out.println("Value of a-=2: "+ (a-=2));

System.out.println("Value of a\*=2: "+ (a\*=2));

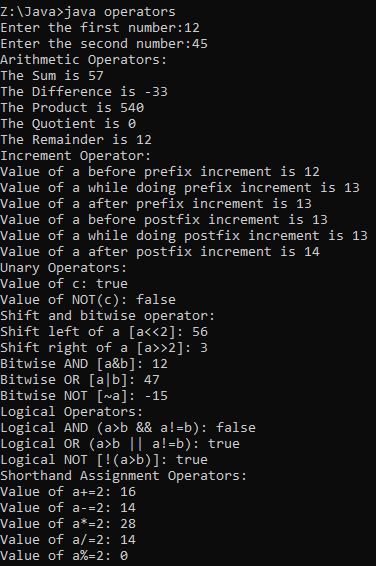
System.out.println("Value of a/=2: "+ (a/=2));

System.out.println("Value of a%=2: "+ (a%=2));

}

}

**OUTPUT:**



**RESULT:**

The above program has been successfully executed.

**Ex.No:03(b) TERNARY OPERATORS**

**DATE:08/08/23**

**AIM:**

To write program to perform ternary operations using java.

**SOURCE CODE:**

import java.util.\*;

class ternary

{

public static void main(String arg[])

{

int a,b;

Scanner ab=new Scanner(System.in);

System.out.print("Enter the first number:");

a=ab.nextInt();

System.out.print("Enter the second number:");

b=ab.nextInt();

int big;

big=a>b?a:b;

System.out.println("The biggest of 2 numbers: "+big);

int choice;

System.out.println("Choices"+'\n'+"0.Addition"+'\n'+"1.Subtraction");

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

choice=ab.nextInt();

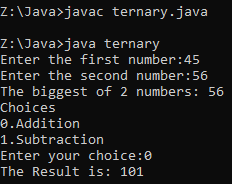
int res=(choice==0)?(a+b):(choice==1)?(a-b):(a=b=0);

System.out.println("The Result is: "+res);

}

}

**OUTPUT:**

****

**RESULT:**

The above program has been executed successfully.

**Ex.No:03(c) AREA OF A CIRCLE**

**DATE:08/08/23**

**AIM:**

To compute the area of the circle using java.

**SOURCE CODE:**

import java.util.\*;

class area{

public static void main(String arg[]){

final double pi=3.14;

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

Scanner ab=new Scanner(System.in);

float r=ab.nextFloat();

System.out.print("Area is "+(pi\*r\*r));

}

}

**OUTPUT:**

****

**Result:**

The above program has been executed successfully

**Ex.No:3(d) STATIC AND NON STATIC VARIABLES**

**DATE:08/08/2023**

**AIM:**

To write a program to use make use of static and non static variable using java.

**SOURCE CODE:**

class staticandnonstatic

{

static int count=0;

int count1=0;

staticandnonstatic()

{

count++;

count1++;

System.out.println("Static value of count: "+count+'\t'+ "Non static value of count: "+count1);

}

public static void main(String args[])

{

staticandnonstatic c1=new staticandnonstatic();

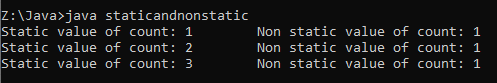
staticandnonstatic c2=new staticandnonstatic();

staticandnonstatic c3=new staticandnonstatic();

}

}

**OUTPUT:**



**RESULT:**

` The above program has been executed successfully.

**Ex.No:04 (a) CREATING A DOMAIN NAME**

**DATE:12/08/2023**

**AIM:**

To write a program for creating domain name.

**SOURCE CODE:**

import java.util.Scanner;

class j18DomainName{

public static void main(String a[]){

Scanner s = new Scanner(System.in);

System.out.print("Name: ");

String name = s.nextLine();

int x = name.length();

int y = name.indexOf(" ");

int i,j;

String fn="";

String ln="";

String dn="";

for(i=0;i<y;i++){

fn=fn+name.charAt(i);

}

System.out.print("First Name: " + fn);

for(i=y+1;i<x;i++){

ln=ln+name.charAt(i);

}

System.out.print("\nLast Name: " + ln);

for(j=0;j<3;j++){

dn=dn+fn.charAt(j);

}

for(j=0;j<3;j++){

dn=dn+ln.charAt(j);

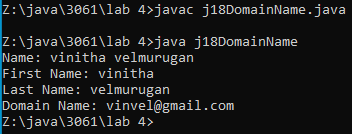
}

System.out.print("\nDomain Name: " + dn + "@gmail.com");

}

}

**OUTPUT:**



**RESULT**:

Thus, the creating domain name is successfully implemented.

**Ex.No:4(b) E-MAIL VERIFICATION**

**DATE:12/08/2023**

**AIM:**

To write a program to get an email from the user and check whether it is a valid mail using java.

**SOURCE CODE:**

import java.util.Scanner;

public class j19EmailID {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your email address: ");

String email = scanner.nextLine();

scanner.close();

if (isValidEmail(email)) {

System.out.println("Valid email address.");

} else {

System.out.println("Invalid email address.");

}

}

public static boolean isValidEmail(String email) {

if (email == null || email.length() == 0) {

return false;

}

int atIndex = email.indexOf('@');

int dotIndex = email.lastIndexOf('.');

if (atIndex <= 0 || dotIndex <= atIndex || dotIndex > email.length() - 3) {

return false;

}

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

char c = email.charAt(i);

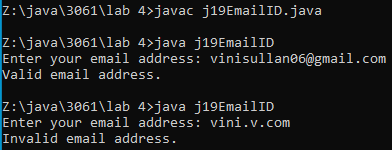
}

return true;

}

}

**OUTPUT**:



**RESULT**:

Hence, the Email ID Validation program is executed successfully.

**Ex.No:05 IMPLEMENTATION OF LOOPS**

**DATE:02/09/2023**

**AIM:**

To implement different loops using java.

**SOURCE CODE:**

**1)FIBONACCI SERIES USING WHILE LOOP**

import java.util.Scanner;

class Fibonacci{

public static void main(String args[]) {

System.out.println("Entet The Number:");

Scanner t=new Scanner(System.in);

int a=0, b=1, c, i = 1, n;

n=t.nextInt();

System.out.println("Fibonacci series:");

System.out.print(a+" "+b);

while(i<n) {

c = a + b; System.out.print(" ");

System.out.print(c);

a = b;

b = c;

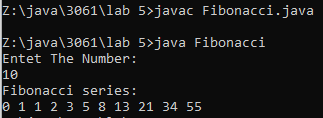
i++;

}

}

}

**OUTPUT**:



**2)AMSTRONG NUMBER USING DO WHILE LOOP**

import java.util.Scanner;

class Amstrong{

public static void main(String args[]){

Scanner t =new Scanner(System.in);

System.out.println("Enter Your Number:");

int a,temp,n,r,sum=0;

n = t.nextInt();

a =n;

temp = n ;

while(n !=0){

r = n%10;

sum += Math.pow(r, 3);

n = n/10;

}

if(temp == sum)

System.out.println("It Is Amstrong Number");

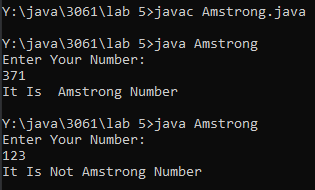
else

System.out.println("It Is Not Amstrong Number");

}

}

**OUTPUT**:



**3)SWITCHCASE**

import java.util.Scanner;

class switchcase{

public static void main(String args[]){

Scanner t = new Scanner(System.in);

int ch;

char c;

do{

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

ch = t.nextInt();

switch(ch){

case 1:

System.out.println("Your Choice Is TEA");

break;

case 2:

System.out.println("Your Choice Is COFFEE");

break;

case 3:

System.out.println("Your Choice Is JUICE");

break;

case 4:

System.out.println("Exited");

System.exit(0);

break;

default:

System.out.println("Your Choice Is Invalid");

break;

}

System.out.print("Do you want to continue ? (Y/N) : ");

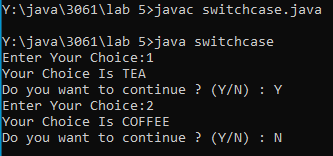
c=t.next().charAt(0);

}while(c=='y'||c=='Y');

}

}

**OUTPUT**:



**4) ODD OR EVEN**

import java.util.Scanner;

class OddorEven{

public static void main(String args[]){

Scanner t= new Scanner(System.in);

int n;

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

n=t.nextInt();

if(n%2==0){

System.out.println("The number is even" );

}

else{

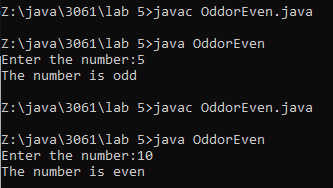
System.out.println("The number is odd");

}

}

}

**OUTPUT**:



**5)PALINDROME OR NOT**

import java.util.Scanner;

class palindrome{

public static void main(String args[]){

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

Scanner t=new Scanner(System.in);

int n,r,sum=0;

n=t.nextInt();

int a=n;

int temp=n;

while(n>0){

r=n%10;

sum=(sum\*10)+r;

n=n/10;

}

if(temp==sum){

System.out.println("The number " +a+ " is palindrome");

}

else{

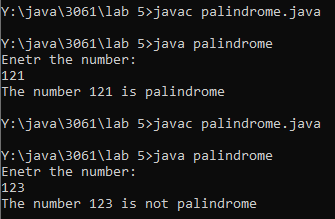
System.out.println("The number " +a+ " is not palindrome");

}

}

}

**OUTPUT**:

****

**RESULT**:  
 Thus, the different loops are implemented successfully.

**Ex.No:06 IMPLEMENTATION OF ARRAY**

**DATE:05/09/2023**

**AIM**:  
 To implement foloeing [programs

1) Sorting an array

2) Matrix Addition and Multiplication

**SOURCE CODE:**

**1)SORTING AN ARRAY**

import java.util.Scanner;

class array1D{

public static void main(String arg[]){

System.out.println("\n--Sorting an 1D Array--");

Scanner s = new Scanner(System.in);

System.out.print("Enter the Number of Elements in an Array: ");

int n=s.nextInt();

int[] arr = new int[n];

System.out.print("Enter the Elements in the Array: ");

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

arr[i]=s.nextInt();

int temp;

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

for(int j=0;j<n;j++){

if(arr[i]<arr[j]){

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

System.out.print("\nArray sorted in Ascending Order: ");

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

System.out.print(arr[i] + " ");

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

for(int j=0;j<n;j++){

if(arr[i]>arr[j]){

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

System.out.print("\nArray sorted in Descending Order: ");

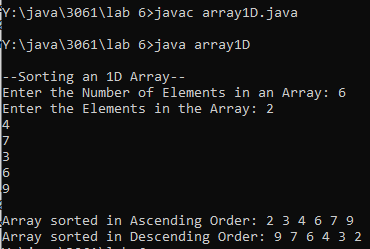
for(int i=0;i<n;i++)

System.out.print(arr[i] + " ");

}

}

**OUTPUT:**



**2) MATRIX ADDITION AND MULTIPLICATION**

import java.util.Scanner;

class MatrixADDnMUL{

public static void main(String args[]){

Scanner s=new Scanner(System.in);

System.out.print("Enter the number of rows in Matrix A: ");

int r1=s.nextInt();

System.out.print("Enter the number of columns in Matrix A: ");

int c1=s.nextInt();

int[][] a = new int[r1][c1];

System.out.print("Enter the number of rows in Matrix B: ");

int r2=s.nextInt();

System.out.print("Enter the number of columns in Matrix B: ");

int c2=s.nextInt();

int[][] A = new int[r1][c1];

int[][] B = new int[r2][c2];

System.out.println("Enter elements of Matrix A:");

for(int i=0;i<r1;i++){

for(int j=0;j<c1;j++){

A[i][j]=s.nextInt();

}

}

System.out.println("Enter elements of Matrix B:");

for(int i=0;i<r2;i++){

for(int j=0;j<c2;j++){

B[i][j]=s.nextInt();

}

}

if(r1==r2 && c1==c2){

System.out.println("Result of Matrix Addition:");

int C[][] = new int[r1][c1];

for(int i=0;i<r1;i++){

for(int j=0;j<c1;j++){

C[i][j]=A[i][j]+B[i][j];

System.out.print(C[i][j] + " "); }

System.out.print("\n");

}

}

else

System.out.println("Matrix Addition not possible..");

if(r2==c1){

System.out.println("Result of Matrix Multiplication:");

int D[][] = new int[r1][c2];

for(int i=0;i<r1;i++){

for(int j=0;j<c2;j++){

for(int k=0;k<r2;k++){

D[i][j]+=A[i][k]+B[k][j];

}

System.out.print(D[i][j] + " ");

}

System.out.print("\n");

}

}

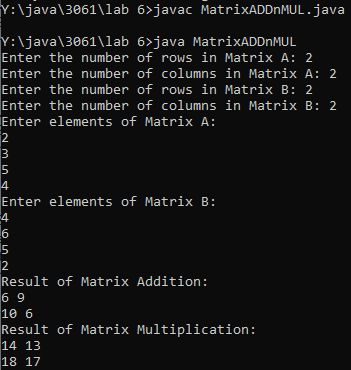
else

System.out.println("Matrix Multiplication not possible..");

}

}

**OUTPUT:**



**RESULT:**

Thus Sorting an array, matrix addition and multiplication are implemented successfully

**Ex.No:07 CONSTRUCTORS AND FUNCTION OVERLOADING**

**DATE:12/09/2023**

**AIM**:

To implement a Constructors and Function Overoading using in java.

**SOURCE CODE:**

**CONSTRUCTORS:**

class Person {

String name;

int age;

public Person() {

System.out.println("Default Constructor");

name = "Unknown";

age = 0;

}

public Person(String name, int age) {

System.out.println("Parameterized Constructor");

this.name = name;

this.age = age;

}

}

public class Constructor {

public static void main(String[] args) {

Person p1 = new Person();

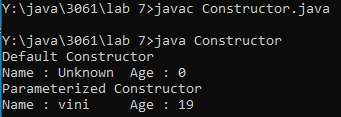
System.out.println("Name : " + p1.name +"\tAge : " + p1.age);

Person p2 = new Person("vini", 19);

System.out.println("Name : " + p2.name +"\tAge : " + p2.age);

}

}  
**OUTPUT:**



**FUNCTION OVERLOADING:**

public class FunctionOverloading {

public static int add(int a, int b) {

return a + b;

}

public static int add(int a, int b, int c) {

return a + b + c;

}

public static double add(double a, double b) {

return a + b;

}

public static String add(String str1, String str2) {

return str1 + str2;

}

public static void main(String[] args) {

int x = 5, y = 10, z = 15;

int r1 = add(x, y);

int r2 = add(x, y, z);

double a = 2.5, b = 3.5;

double r3 = add(a, b);

String str1 = "Hello ", str2 = "World!";

String r4 = add(str1, str2);

System.out.println(x + " + " + y + " = " + r1);

System.out.println(x + " + " + y + " + " + z + " = " + r2);

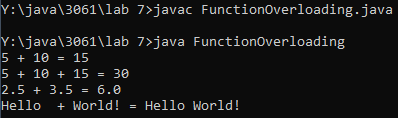
System.out.println(a + " + " + b + " = " + r3);

System.out.println(str1 + " + " + str2 + " = " + r4);

}

}

**OUTPUT:**



**RESULT:**

Thus, the constructors and function overloading are implemented successfullu.

**Ex.No:08 IMPLEMENTATION OF INHERITANCE**

**DATE:19/09/2023**

**AIM:**

To implement different types of inheritance.

**SOURCE CODE:**

**1)SINGLE LEVEL INHERITANCE**

class ParentClass {

double salary = 40000.0;

}

class ChildClass extends ParentClass {

int bonus = 10000;

}

class SingleInheritance{

public static void main(String[] args) {

ChildClass obj = new ChildClass();

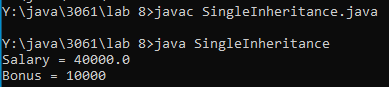
System.out.println("Salary = " + obj.salary);

System.out.println("Bonus = " + obj.bonus);

}

}

**OUTPUT**:



**2)MULTI LEVEL INHERITANCE**

class Car {

public Car() {

System.out.println("Class Car");

}

public void vehicleType() {

System.out.println("Vehicle Type: Car");

}

}

class Maruti extends Car {

public Maruti() {

System.out.println("Class Maruti");

}

public void brand() {

System.out.println("Brand: Maruti");

}

public void speed() {

System.out.println("Speed: 90 kmph");

}

}

public class MultiInheritance extends Maruti {

public MultiInheritance() {

System.out.println("Maruti Model: 800");

}

public void speed() {

System.out.println("Max Speed: 80 kmph");

}

public static void main(String args[]) {

MultiInheritance obj = new MultiInheritance();

obj.vehicleType();

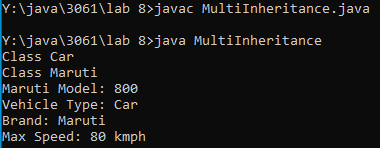
obj.brand();

obj.speed();

}

}

**OUTPUT:**



**3)HIERARCHICAL INHERITANCE**

class Employee {

float salary = 75000;

}

class PermanentEmployee extends Employee {

double hike = 0.5;

}

class TemporaryEmployee extends Employee {

double hike = 0.35;

}

class Hierarchical {

public static void main(String args[])

{

PermanentEmployee obj1 = new PermanentEmployee();

TemporaryEmployee obj2 = new TemporaryEmployee();

System.out.println(obj1.salary);

System.out.println(obj1.hike);

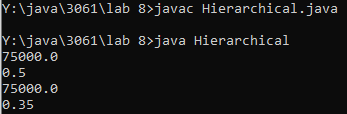
System.out.println(obj2.salary);

System.out.println(obj2.hike);

}

}

**OUTPUT:**



**4) MULTIPLE INHERITANCE**

interface CNG\_car{

void drive();

void CNG\_car();

}

interface Petrol\_car{

void drive();

void Petrol\_car();

}

class Hybrid implements CNG\_car,Petrol\_car{

public void drive(){

System.out.println("Driving a Hybrid Car");

}

public void CNG\_car(){

System.out.println("Using CNG Kit for Hybrid Car");

}

public void Petrol\_car(){

System.out.println("Using Petrol Kit for Hybrid Car");

}

}

class MultipleInheritance{

public static void main(String args[]){

Hybrid car = new Hybrid();

car.drive();

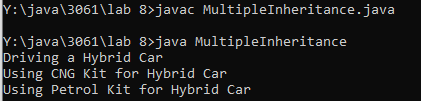
car.CNG\_car();

car.Petrol\_car();

}

}

**OUTPUT**:



**RESULT:**  
 Thus, the different types of inheritance are executed successfully

**Ex.No:09 PACKAGE AND INTERFACE**

**DATE:26/09/2023**

**AIM:**

To implement package and interface.

**SOURCE CODE:  
1)PACKAGE:**

**PACKAGE FILE**

package Cal;

public class CalcOP{

public static double add(double n1,double n2){

return n1+n2;

}

public static double sub(double n1,double n2){

return n1-n2;

}

public static double pro(double n1,double n2){

return n1\*n2;

}

public static double quo(double n1,double n2){

return n1/n2;

}

}

**PACKAGE USING FILE**

import java.util.Scanner;

import Cal.CalcOP;

public class Calculator {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

System.out.print("Enter first number : ");

double num1 = s.nextDouble();

System.out.print("Enter second number : ");

double num2 = s.nextDouble();

System.out.print("1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\nEnter the choice :");

int ch=s.nextInt();

double result;

CalcOP obj = new CalcOP();

switch (ch) {

case 1:

result = obj.add(num1, num2);

break;

case 2:

result = obj.sub(num1, num2);

break;

case 3:

result = obj.pro(num1, num2);

break;

case 4:

result = obj.quo(num1, num2);

break;

default:

System.out.println("Invalid operator");

return;

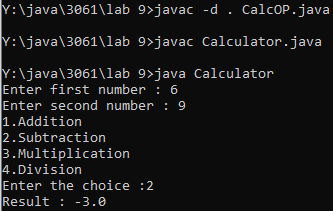
}

System.out.println("Result : " + result);

}

}

**OUTPUT**:



**2)INTERFACE:**

interface Shape {

double getArea();

}

class Circle implements Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double getArea() {

return 3.14 \* radius \* radius;

}

}

class Rectangle implements Shape {

private double length;

private double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

@Override

public double getArea() {

return length \* width;

}

}

public class Interface {

public static void main(String[] args) {

Circle c = new Circle(5.0);

Rectangle r = new Rectangle(4.0, 6.0);

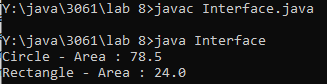
System.out.println("Circle - Area : " + c.getArea());

System.out.println("Rectangle - Area : " + r.getArea());

}

}

**OUTPUT:**



**RESULT:**

Thus, the package and interface are implemented.

**Ex.No:10 IMPLEMENTATION OF SUPER, INSTANCEOF, ACCESS SPECIFIERS AND ABSRACT CLASS AND METHODS**

**DATE:03/10/2023**

**AIM:**

To implement of Super, InstanceOf, access Specifiers and Abstract Class and Methods.

**SOURCE CODE:**

**1)SUPER AND INSTANCEOF**

class InstanceOf extends Parent

{

public static void main(String args[])

{

InstanceOf i=new InstanceOf();

System.out.println(i instanceof InstanceOf );

InstanceOf j=null ;

System.out.println(j instanceof InstanceOf );

Parent p1=new InstanceOf ();

System.out.println(p1 instanceof InstanceOf );

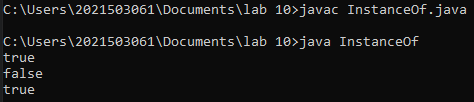
}

}

class Parent

{

}  
**OUTPUT:**



**2)ACCESS SPECIFIERS**

class A {

int i;

int j;

void setij(int x, int y) {

i = x;

j = y;

}

}

class B extends A {

int total;

void sum() {

total = i + j;

}

}

class AccessSpecifiers{

public static void main(String args[]) {

B subOb = new B();

subOb.setij(10, 12);

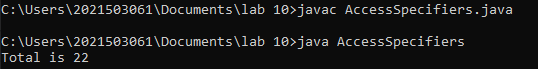
subOb.sum();

System.out.println("Total is " + subOb.total);

}

}

**OUTPUT:**



**3)ABSTRACT CLASS AND METHOD**

abstract class Shape {

double dim1;

double dim2;

Shape(double a, double b) {

dim1 = a;

dim2 = b;

}

abstract double area();

}

class Rectangle extends Shape {

Rectangle(double a, double b) {

super(a, b);

}

double area() {

System.out.println("Inside Area for Rectangle.");

return dim1 \* dim2;

}

}

class Triangle extends Shape {

Triangle(double a, double b) {

super(a, b);

}

double area() {

System.out.println("Inside Area for Triangle.");

return dim1 \* dim2 / 2;

}

}

class AbstractClass {

public static void main(String args[]) {

Rectangle r = new Rectangle(9, 5);

Triangle t = new Triangle(10, 8);

Shape shaperef;

shaperef = r;

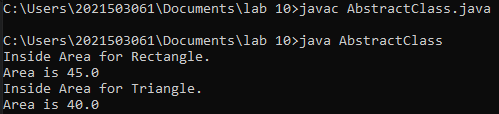
System.out.println("Area is " + shaperef.area());

shaperef = t;

System.out.println("Area is " + shaperef.area());

}

}

**OUTPUT:**  


**RESULT:**

Thus, Super, Instanceof, Access Specifiers, Abstract Class and Methods are executed successfully.

**Ex.No:11 IMPLEMENTATION OF GUI – EMPLOYEE FEEDBACK FORM**

**DATE:10/10/2023**

**AIM:**

To create an Employee Feeadback Form using swings and awt.

**SOURCE CODE:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class EmpFeedbackForm extends JFrame {

public EmpFeedbackForm() {

setTitle("Employee Feedback Form");

setSize(500, 400);

setLocationRelativeTo(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JLabel nameLabel = new JLabel("Employee Name:");

JTextField nameField = new JTextField(20);

JLabel feedbackLabel = new JLabel("Feedback:");

JTextArea feedbackArea = new JTextArea(5, 20);

JScrollPane scrollPane = new JScrollPane(feedbackArea);

JCheckBox checkBox = new JCheckBox("Anonymous");

JRadioButton excellentRadioButton = new JRadioButton("Excellent");

JRadioButton goodRadioButton = new JRadioButton("Good");

JRadioButton fairRadioButton = new JRadioButton("Fair");

JRadioButton poorRadioButton = new JRadioButton("Poor");

ButtonGroup radioButtonGroup = new ButtonGroup();

radioButtonGroup.add(excellentRadioButton);

radioButtonGroup.add(goodRadioButton);

radioButtonGroup.add(fairRadioButton);

radioButtonGroup.add(poorRadioButton);

JLabel ratingLabel = new JLabel("Rating:");

String[] departments = {"HR", "IT", "Finance", "Operations"};

JComboBox<String> departmentComboBox = new JComboBox<>(departments);

JList<String> hobbiesList = new JList<>(new String[]{"Reading", "Sports", "Traveling"});

feedbackArea.addKeyListener(new KeyAdapter() {

@Override

public void keyTyped(KeyEvent e) {

System.out.println("Key Typed: " + e.getKeyChar());

}

});

addWindowListener(new WindowAdapter() {

@Override

public void windowClosing(WindowEvent e) {

System.out.println("Window is closing");

}

});

checkBox.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

System.out.println("Mouse Clicked on CheckBox");

}

});

setLayout(new GridLayout(10, 2));

add(nameLabel);

add(nameField);

add(feedbackLabel);

add(scrollPane);

add(checkBox);

add(new JLabel());

add(ratingLabel);

add(excellentRadioButton);

add(new JLabel());

add(goodRadioButton);

add(new JLabel());

add(fairRadioButton);

add(new JLabel());

add(poorRadioButton);

add(new JLabel());

add(new JLabel("Department:"));

add(departmentComboBox);

add(new JLabel("Hobbies:"));

add(new JScrollPane(hobbiesList));

departmentComboBox.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String selectedDepartment = (String) departmentComboBox.getSelectedItem();

System.out.println("Selected Department: " + selectedDepartment);

}

});

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

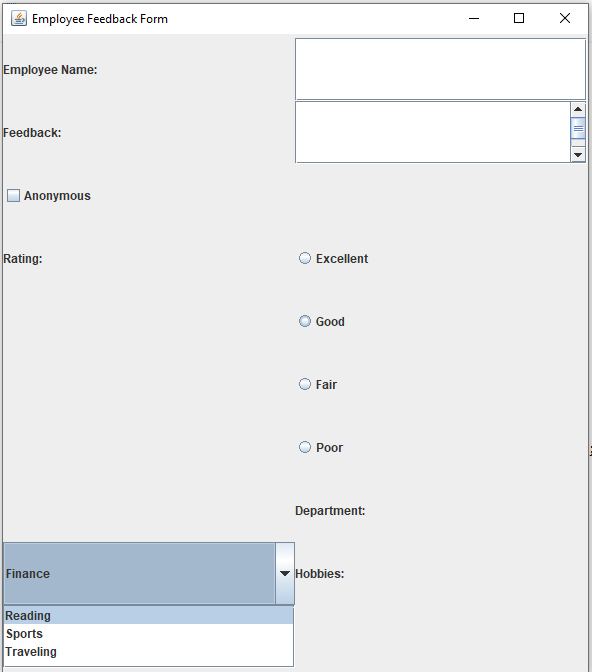
new EmpFeedbackForm().setVisible(true);

}

});

}

**OUTPUT:**



**RESULT:**  
 Hence, the GUI using Employee Feedback Form implemented successfully.

**Ex.No:12 IMPLEMENTATION OF DATABASE CONNECTION**

**DATE:10/10/2023**

**AIM:**

To implement the database connection.

**SOURCE CODE:**

**DATABASE CODE:**

import java.sql.\*;

class MysqlCon{

public static void main(String args[]){

try{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost/COLLEGE","root","root");

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from STUDENT");

while(rs.next())

System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));

con.close();

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

}

}

**INSERT CODE:**

<%@ page import="java.sql.\*" %>

<html>

<body>

<form >

Enter rollno: <input type="text" name="n1"/><br/>

Enter name: <input type="text" name="n2"/><br/>

Enter course: <input type="text" name="n3"/><br/>

<br/>

<input type ="submit" value="Insert"/>

<%!

Connection con = null;

Statement stmt = null;

ResultSet rs = null;

String url = "jdbc:mysql://localhost/college?";

String user = "root";

String pass = "root";

PreparedStatement ps=null;

%>

<%

try {

Class.forName("com.mysql.jdbc.Driver").newInstance();

con = DriverManager.getConnection(url, user, pass);

if(con!=null) {

stmt = con.createStatement();

String inp1=request.getParameter("n1");

String inp2=request.getParameter("n2");

String inp3=request.getParameter("n3");

if((inp1!=null) && (inp2!=null))

{

int id=Integer.parseInt(inp1);

String name=inp2;

String course=inp3;

String sql="insert into student values("+id+", '"+name+"', '"+course+"')";

ps=con.prepareStatement(sql);

ps.execute();

}

}

}

catch (Exception e) {

out.println("Exception: " + e.getMessage());

}

finally {

try {

if (con != null) {

con.close();

}

}

catch (SQLException e) { }

}

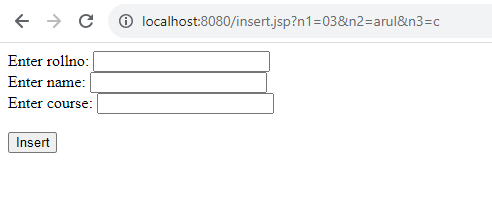
%>

</form>

</body>

</html>

**OUTPUT:**



**READ CODE:**

<%@ page import="java.sql.\*" %>

<html>

<body>

<table border="1">

<th>Roll</th><th>Name</th><th>Course</th>

<%!

Connection con = null;

Statement stmt = null;

ResultSet rs = null;

String url = "jdbc:mysql://localhost/college?";

String user = "root";

String pass = "root";

%>

<%

try {

Class.forName("com.mysql.jdbc.Driver").newInstance();

con = DriverManager.getConnection(url, user, pass);

if(con!=null) {

stmt = con.createStatement();

rs = stmt.executeQuery("select \* from student");

}

while (rs.next()) {

%>

<tr>

<td><%=rs.getInt(1)%></td>

<td><%=rs.getString(2)%></td>

<td><%=rs.getString(3)%></td>

</tr>

<%

}

}

catch (Exception e) {

out.println("Exception: " + e.getMessage());

}

finally {

try {

if (con != null) {

con.close();

}

}

catch (SQLException e) { }

}

%>

</table>

</body>

</html>

OUTPUT:



**RESULT:**

Thus, the database connection is implemented successfully.

**Ex.No:13 IMPLEMENTATION OF BASIC JSP PROGRAM**

**DATE:31/10/2023**

**AIM:**

To implement a basic JSP program to add two numbers.

**SOURCE CODE:**

**ADD TWO NUMBERS:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Simple Calculator</title>

</head>

<body>

<h2>Simple Calculator</h2>

<form action="calculate.jsp" method="post">

<label for="num1">Number 1:</label>

<input type="text" id="num1" name="num1" required>

<br>

<label for="num2">Number 2:</label>

<input type="text" id="num2" name="num2" required>

<br>

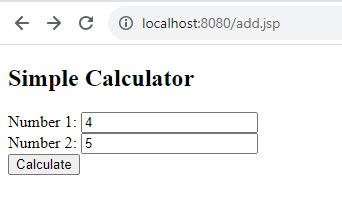
<button type="submit">Calculate</button>

</form>

</body>

</html>

**OUTPUT:**



**CALCULATE:**

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%@ page import="java.io.\*, java.util.\*" %>

<%@ page import="java.text.\*" %>

<%

// Get input parameters

int num1 = Integer.parseInt(request.getParameter("num1"));

int num2 = Integer.parseInt(request.getParameter("num2"));

// Perform calculation

int result = num1 + num2;

%>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Calculation Result</title>

</head>

<body>

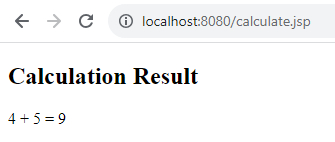
<h2>Calculation Result</h2>

<p><%= num1 %> + <%= num2 %> = <%= result %></p>

</body>

</html>

**OUTPUT:**



**RESULT:**

Thus, the basic JSP program

**Ex.No:14 IMPLEMENTATION OF MULTITHREADING**

**DATE:07/11/2023**

**AIM:**

To implement Multithreading program in java.

**SOURCE CODE:**

**1.MULTI THREADING WITH SYNCHRONIZATION**

class Account

{

int balance;

synchronized void balInquiry(int bal,int wd)

{

balance = bal-wd;

try

{

Thread.sleep(400);

}

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

System.out.println("THREAD ID="+Thread.currentThread().getId()+"BALANCE="+balance);

}

}

class MyThread1 extends Thread

{

Account a;

int balance ;

MyThread1(Account a,int bal)

{

this.a=a;

this.balance=bal;

}

public void run()

{

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

{

a.balInquiry(balance,30);

balance=a.balance;

}

}

}

class MyThread2 extends Thread

{

Account a;

int balance ;

MyThread2(Account a,int bal)

{

this.a=a;

this.balance=bal;

}

public void run()

{

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

{

a.balInquiry(balance,20);

balance=a.balance;

}

}

}

public class MulThreadWithSynchronization

{

public static void main(String args[])

{

Account a=new Account ();

MyThread1 t1=new MyThread1(a,2000);

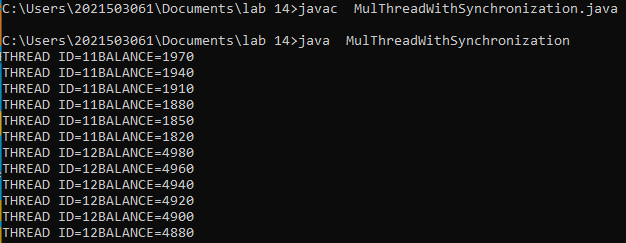
MyThread2 t2=new MyThread2(a,5000);

t1.start();

t2.start();

}

}

**OUTPUT:**  


**2.MULTI THREADING WITHOUT SYNCRONIZATION**

class Account

{

int balance;

void balInquiry(int bal,int wd)

{

balance = bal-wd;

try

{

Thread.sleep(400);

}

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

System.out.println("THREAD ID="+Thread.currentThread().getId()+"BALANCE="+balance);

}

}

class MyThread1 extends Thread

{

Account a;

int balance ;

MyThread1(Account a,int bal)

{

this.a=a;

this.balance=bal;

}

public void run()

{

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

{

a.balInquiry(balance,30);

balance=a.balance;

}

}

}

class MyThread2 extends Thread

{

Account a;

int balance ;

MyThread2(Account a,int bal)

{

this.a=a;

this.balance=bal;

}

public void run()

{

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

{

a.balInquiry(balance,20);

balance=a.balance;

}

}

}

public class MulThreadWithoutSynchronization

{

public static void main(String args[])

{

Account a=new Account ();

MyThread1 t1=new MyThread1(a,2000);

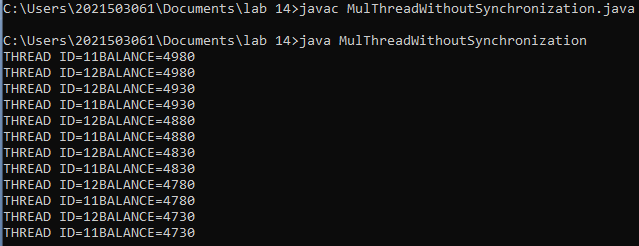
MyThread2 t2=new MyThread2(a,5000);

t1.start();

t2.start();

}

}

**OUTPUT:**

**RESULT:**

Thus, the multithreading program is executed successfully.

**Ex.No:15 IMPLEMENTATION OF READING AND WRITTING**

**DATE:07/11/2023**

**AIM:**

To implement a java programthat reads from console and write it file and to read a file and print it on console.

**SOURCE CODE:**

import java.io.\*;

public class FileReadWrite {

public static void main(String[] args) {

try {

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

System.out.print("Enter content to write to file: ");

String content = reader.readLine();

BufferedWriter writer = new BufferedWriter(new FileWriter("output.txt"));

writer.write(content);

writer.close();

System.out.println("Content written to file 'output.txt'.");

BufferedReader fileReader = new BufferedReader(new FileReader("output.txt"));

System.out.println("Content read from file:");

String line;

while ((line = fileReader.readLine()) != null) {

System.out.println(line);

}

fileReader.close();

} catch (IOException e) {

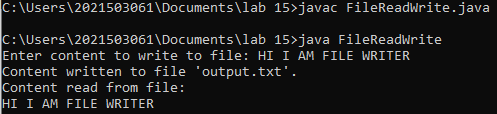
e.printStackTrace();

}

}

}

**OUTPUT:**



**RESULT:**  
 Thus, the reading and writing are implemented successfully.

**Ex.No:16 IMPLEMENTATION OF CHAT APPLICATION**

**DATE:18/11/2023**

**AIM:**

To implement chat application program between server and client.

**SOURCE CODE:**

**CHAT SERVER:**

import java.net.\*;

import java.io.\*;

import java.util.Scanner;

public class Server{

public static void main(String args[]) throws Exception{

ServerSocket ssoc=new ServerSocket(2002);

Socket soc=ssoc.accept();

BufferedReader InFromClient = new BufferedReader(new InputStreamReader(soc.getInputStream()));

Scanner sc=new Scanner(System.in);

PrintStream OutToClient = new PrintStream(soc.getOutputStream());

String msg;

while(true){

msg=InFromClient.readLine();

if(msg.equalsIgnoreCase("END")){

OutToClient.println("BYE");

break;

}

System.out.print("Client:"+msg+"\n");

System.out.print("Server:");

msg=sc.nextLine();

OutToClient.println(msg);

}

ssoc.close();

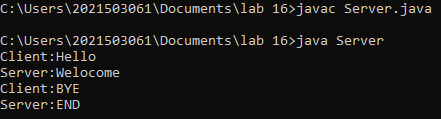
soc.close();

OutToClient.close();

InFromClient.close();

}

**}  
OUTPUT:**



**CHAT CLIENT:**

import java.net.\*;

import java.io.\*;

import java.util.Scanner;

public class Client{

public static void main(String args[]) throws Exception{

Socket soc=new Socket("127.0.0.1",2002);

Scanner sc=new Scanner(System.in);

PrintStream OutToServer = new PrintStream(soc.getOutputStream());

BufferedReader InFromServer = new BufferedReader(new InputStreamReader(soc.getInputStream()));

String msg;

while(true){

System.out.print("Client:");

msg=sc.nextLine();

OutToServer.println(msg);

msg=InFromServer.readLine();

System.out.print("Server:"+msg+"\n");

if(msg.equalsIgnoreCase("BYE"))

break;

}

soc.close();

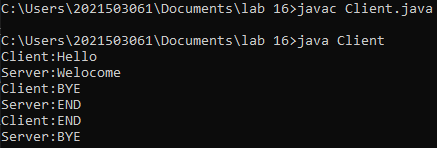
OutToServer.close();

InFromServer.close();

}

}

**OUTPUT:**

****

**RESULT:**

Thus, the chat program between client and server are executed successfully.