## 

: 25/06/2024

## 1

**Practical Number:- 01**

**Q1) Write a java program to find reverse of a number.**

**Code:**

import java.util.Scanner; public class Reverse

{

public static void main(String args[])

{

Scanner Sc= new Scanner (System.in); int n,x,r,sum=0;

System.out.println("Enter a positive integer: "); x=Sc.nextInt();

while(x<0)

{

}

n=x;

System.out.println("Invalid positive integer"); x=Sc.nextInt();

while(x>0)

{

r=x%10;

sum=(sum\*10)+r; x=x/10;

}

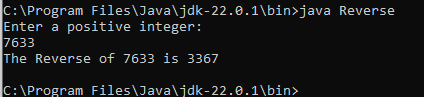
System.out.println("The Reverse of "+ n +" is "+ sum);

}

}

## Output:

Correct Input:



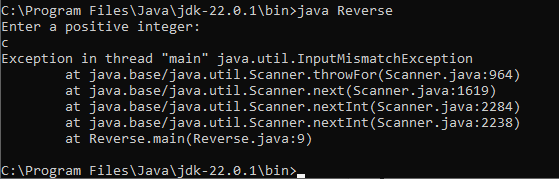
## 

: 25/06/2024

## 2

**Practical Number:- 01**

Incorrect Input:



**Q2) Implement a Java function that calculates the sum of digits for a given char array consisting of the digits '0' to '9'. The function should return the digit sum as a long value.**

**Code:**

import java.util.Scanner; public class SumOfDigits

{

public static long calc()

{

int i;

long x =0L,sum=0L;

char c[]={'0','1','2','3','4','5','6','7','8','9'};

for(i=c.length-1;i>=0;i--)

{

x=c[i]-'0';

sum=sum+x;

}

return sum;

}

public static void main(String args[])

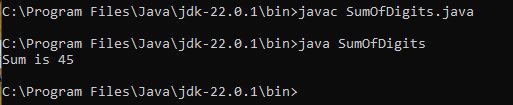
{

}

}

## Output:

long s=calc(); System.out.println("Sum is "+ s);



## 

: 25/06/2024

## 3

**Practical Number:- 01**

**Q3) Find the smallest and largest element from the array.**

**Code:**

import java.util.\*; class LargeSmallArray

{

public static void main(String args[])

{

int num[]=new int[]{13,22,27,11,4}; int s=num[0];

int l=num[0]; int i;

System.out.println("Array Elements are: "); for(i=0;i<num.length-1;i++)

{

System.out.println(num[i]);

}

for(i=1;i<num.length-1;i++)

{

if(num[i]>l)

l=num[i]; else if(num[i]<s)

s=num[i];

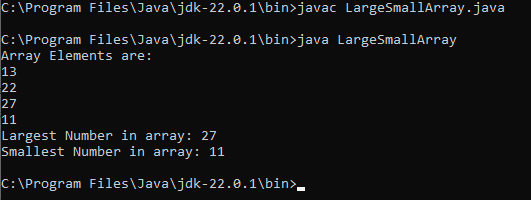
}

}

}

## Output:

System.out.println("Largest Number in array: "+ l); System.out.println("Smallest Number in array: "+ s);



## 

: 25/06/2024

## 4

**Practical Number:- 01**

**Q4) Find the smallest and largest element from the array by taking input from the user. Code:**

import java.util.Scanner; public class LargeSmallArray2

{

public static void main(String args[])

{

Scanner scan = new Scanner(System.in); System.out.print("Enter the number of elements in the array: "); int n = scan.nextInt();

int[] array = new int[n];

System.out.println("Enter the elements of the array:"); for (int i = 0; i < n; i++)

{

array[i] = scan.nextInt();

}

int smallest = array[0]; int largest = array[0]; for (int i = 1; i < n; i++)

{

if (array[i] < smallest)

{

smallest = array[i];

}

if (array[i] > largest)

{

largest = array[i];

}

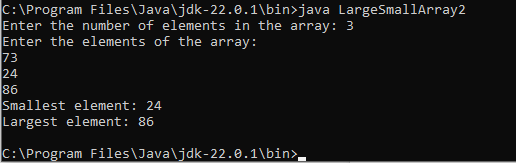
}

}

}

## Output:

System.out.println("Smallest element: " + smallest); System.out.println("Largest element: " + largest); scan.close();



## 

: 27/06/2024

## 1

**Practical Number:- 02**

## Q1) Write a Java Program to Print the area and perimeter of a circle. CODE:

import java.util.Scanner; class Circle

{

public static void main(String args[])

{

double r,cp,ca;

Scanner st=new Scanner(System.in); System.out.print("Enter Raidus of a cricle Number:-"); r=st.nextDouble();

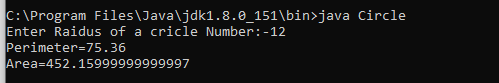
cp=2\*3.14\*r; ca=3.14\*r\*r;

System.out.println("Perimeter="+cp); System.out.println("Area="+ca);

}

}

## Output:



Q2) **Write a Java program That takes a number as Inpuit and prints its multiplication table up to 10**. **CODE:**

import java.util.Scanner; class Multi

{

public static void main(String args[])

{

int num,i;

Scanner st=new Scanner(System.in); System.out.print("Enter Number:-"); num=st.nextInt(); System.out.println("Table of"+num);

## 

: 27/06/2024

## 2

**Practical Number:- 02**

for(i=1;i<=10;i++)

{

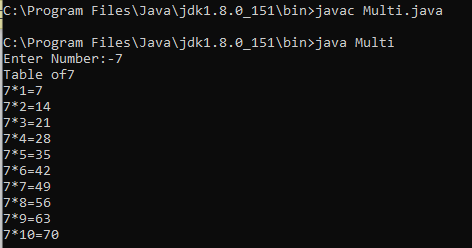
System.out.println(num+"\*"+i+"="+num\*i);

}

}

}

## OUTPUT:



Q3) **Write a Java Program to add to binary numbers CODE:**

import java.util.Scanner; class Binary

{

public static void main(String args[])

{

long b1,b2; int i=0,rem=0;

int[] sum=new int[20];

Scanner in=new Scanner(System.in); System.out.print("Enter a first binary Number:-"); b1=in.nextLong();

System.out.print("Enter Second Binary Number:-"); b2=in.nextLong();

while(b1!=0||b2!=0)

{

## 

: 27/06/2024

## 3

**Practical Number:- 02**

sum[i++]=(int)((b1%10+b2%10+rem)%2); rem=(int)((b1%10+b2%10+rem)/2); b1=b1/10;

b2=b2/10;

}

if(rem!=0)

{

sum[i++]=rem;

}

--i;

System.out.print("Sum of b1+b2= "); while(i>=0)

{

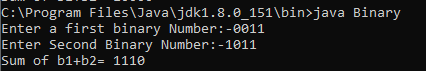
System.out.print(sum[i--]);

}

}

}

## OUTPUT:



## 

## Practical Number:- 01

Q1. Write a Java program to count the letters, spaces, numbers and other characters of an input string.

**Code:**

import java.util.Scanner; class Pract3a {

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

Scanner in = new Scanner(System.in); System.out.print("Enter a string: ");

st = in.nextLine(); count(st);

}

public static void count(String x) { char[] ch = x.toCharArray(); int i, l = 0, s = 0, d = 0, a = 0;

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

if (Character.isLetter(ch[i])) { l++;

}

if (Character.isDigit(ch[i])) { d++;

}

if (Character.isSpaceChar(ch[i])) { s++;

}

else {

}

}

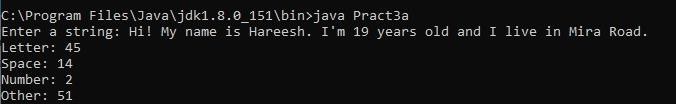
a++;

}

}

**Output:**

System.out.println("Letter: " +l); System.out.println("Space: " +s); System.out.println("Number: " +d); System.out.println("Other: " +a);



## 

09-07-2024

## 9

**Practical Number:- 01**

Q2. Write a Java Program to demonstrate the use of constructor and destructor.

**Code:**

public class Pract4b { public Pract4b() {

System.out.println("Hi Hareesh!");

}

public void finalize() { System.out.println("Sayonara");

}

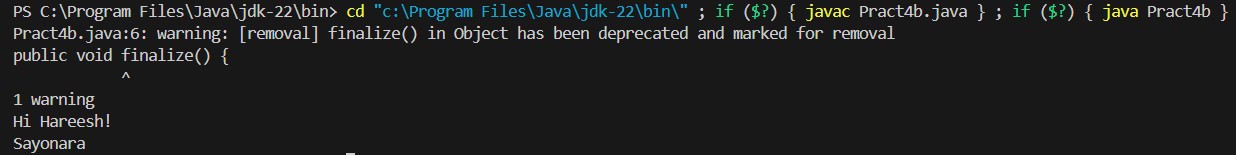
public static void main(String[] args) { Pract4b obj = new Pract4b();

obj = null; System.gc();

}

}

**Output:**



## 

09-07-2024

## 10

**Practical Number:- 01**

Q3. Design a class SortData that contains the method asec and desc.

**Code:**

public class Pract4a {

public static void main(String[] args) { SortData obj = new SortData();

int arr[] = {12,44,45,28,9,87};

obj.asec(arr); obj.desc(arr);

}

}

class SortData {

public void asec(int num[]) { int temp;

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

for (int j = i + 1; j < num.length; j++) { if (num[i] > num[j]) {

temp = num[i] ; num[i] = num[j]; num[j] = temp;

}

}

}

System.out.println("Ascending Order"); for (int i = 0; i < num.length; i++) {

System.out.println(num[i] + " ");

}

}

public void desc(int num[]) { int temp;

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

for (int j = i + 1; j < num.length; j++) { if (num[i] < num[j]) {

temp = num[i] ; num[i] = num[j]; num[j] = temp;

}

}

}

System.out.println("Descending Order"); for (int i = 0; i < num.length; i++) {

System.out.println(num[i] + " ");

}

System.out.println();

}

}

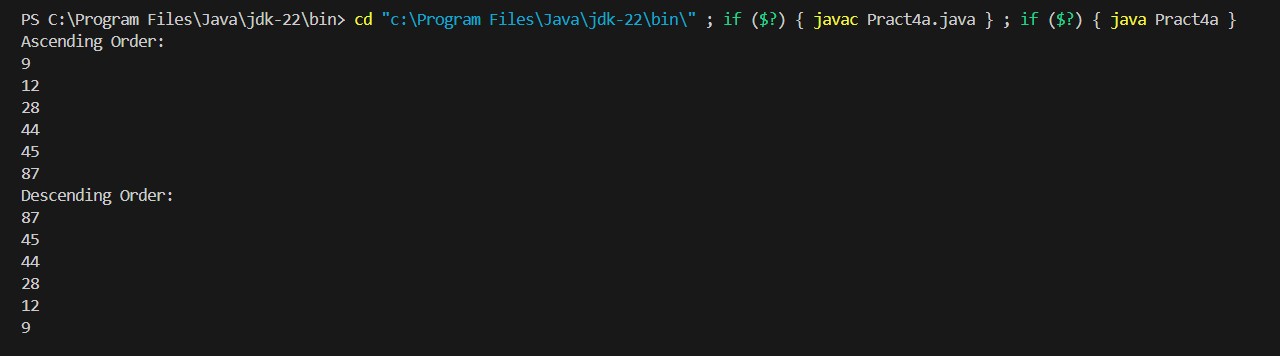
## 

09-07-2024

## 11

**Practical Number:- 01**

**Output:**



## 

11-07-2024

## 12

**Practical Number:- 04**

Q1. Write java program to demonstrate implementation of abstract class.

**Code:**

abstract class Clit {

public abstract int sqrt(int n1); public abstract int cube(int n1);

public void show() {

System.out.println("Im hareesh");

}

}

public class Pract4c extends Clit { @Override

public int sqrt(int n1){

return(n1\*n1);

}

@Override

public int cube(int n1){

return(n1\*n1\*n1);

}

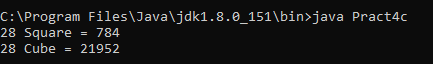
public static void main(String[] args) { Pract4c obj = new Pract4c(); int sq = obj.sqrt(28);

System.out.println("28 Square = " + sq); int cube = obj.cube(28); System.out.println("28 Cube = " + cube);

}

}

**Output:**



## 

11-07-2024

## 13

**Practical Number:- 04**

Q2. Write java program to implement single level inheritance.

**Code:**

class demo {

float pi = 3.14f;

void show() {

System.out.println("Area of circle: ");

}

}

public class Pract5a extends demo { float r = 2.0f;

void area() {

System.out.println(pi\*r\*r);

}

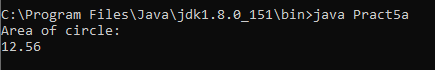
public static void main(String[] args) { Pract5a obj = new Pract5a(); obj.show();

obj.area();

}

}

**Output:**



## 

11-07-2024

## 14

**Practical Number:- 04**

Q3. How interface are implemented with Annotations.

**Code:** @FunctionalInterface interface Drawable {

public void age();

}

public class Hareesh{

public static void main(String[] args) { int age = 19;

Drawable ddd = () -> {

System.out.println("Hareesh age: " + age);

};

ddd.age();

}

}

**Output:**



## 

11-07-2024

## 15

**Practical Number:- 04**

Q4. Write java program to implement method overriding.

**Code:**

class A {

void show(){

System.out.println("Im hareesh");

}

}

class B extends A {

void show(){

System.out.println("Im also hareesh");

}

}

public class Pract5b{

public static void main(String[] args) { B ob = new B();

ob.show();

}

}

**Output:**



## 

: 16-07-2024

## (CJ) Page no:

**Practical Number:- 05**

## Design an abstract class to find or check if the number is Armstrong number or not and generate result.

**Code:**

import java.util.Scanner; abstract class Check {

public abstract boolean ams(int n);

}

public class Armstrong extends Check { @Override

public boolean ams(int n) { int x = n;

int sum = 0; while (x > 0) {

int r = x % 10; sum += r \* r \* r; x = x / 10;

}

return sum == n;

}

public static void main(String[] args) { Scanner sc = new Scanner(System.in); int x;

System.out.print("Enter a positive number: "); x = sc.nextInt();

while (x <= 0) { System.out.println("Invalid input");

System.out.print("Enter a positive number: "); x = sc.nextInt();

}

Armstrong armst = new Armstrong(); if (armst.ams(x))

{

System.out.println(x + " is an Armstrong number");

} else

{

System.out.println(x + " is not an Armstrong number");

}

}

}

**Outputs:**

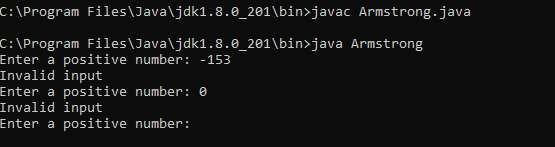
*Incorrect Input*:

## 

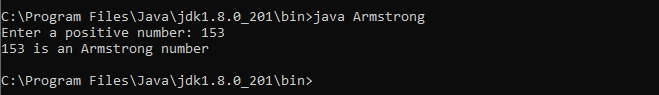
: 16-07-2024

## (CJ) Page no:

**Practical Number:- 05**



*Correct Input*:



## Write a java program to implement multiple inheritance. Code:

interface S

{

public void show();

}

interface T extends S

{

public void display();

}

public class Prac5c implements T

{

@Override

public void display()

{

System.out.println("From interface T");

}

@Override

public void show()

{

System.out.println("From interface S");

}

public static void main(String[] args)

{

}

}

**Output:**

Prac5c ob=new Prac5c(); ob.show();

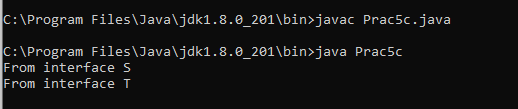
ob.display();

## 

: 16-07-2024

## (CJ) Page no:

**Practical Number:- 05**



## Write a java program for Adding two matrices and print the result for the same. Code:

import java.util.Scanner; public class MatrixAddition

{

public static void main(String args[])

{

int i,j;

int mat1[][] = new int[2][2];

int mat2[][] = new int[2][2];

int mat3[][] = new int[2][2];

Scanner sc = new Scanner(System.in); System.out.print("Enter Matrix 1 Elements: "); for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

mat1[i][j] = sc.nextInt();

}

}

System.out.print("Enter Matrix 2 Elements: "); for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

mat2[i][j] = sc.nextInt();

}

}

System.out.print("Adding both Matrix to from the Third Matrix...\n"); for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

mat3[i][j] = mat1[i][j] + mat2[i][j];

}

}

System.out.print("The Two Matrix Added Successfully..!!\n"); System.out.print("The New Matrix will be :\n"); for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

## 

: 16-07-2024

## (CJ) Page no:

**Practical Number:- 05**

{

System.out.print(mat3[i][j]+" ");

}

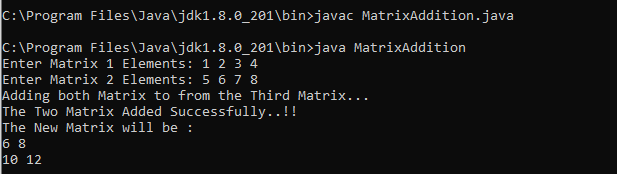
System.out.println();

}

}

}

## Output:



* 1. **Write a java program for multiplying two matrices and print the product for the same.**

## Code:

import java.util.Scanner; public class MatrixMulti

{

public static void main(String args[])

{

int n;

Scanner input = new Scanner(System.in); System.out.print("Enter the base of squared matrices: "); n = input.nextInt();

int[][] a = new int[n][n];

int[][] b = new int[n][n];

int[][] c = new int[n][n];

System.out.print("Enter the elements of 1st martix row wise: \n"); for(int i=0;i<n;i++)

{

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

{

a[i][j] = input.nextInt();

}

}

System.out.print("Enter the elements of 2nd martix row wise: \n"); for(int i=0;i<n;i++)

{

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

{

## 

: 16-07-2024

## (CJ) Page no:

**Practical Number:- 05**

b[i][j] = input.nextInt();

}

}

System.out.println("Multiplying the matrices..."); for(int i=0;i<n;i++)

{

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

{

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

{

c[i][j] = c[i][j] + a[i][k] \* b[k][j];

}

}

}

System.out.println("The product is: "); for(int i=0;i<n;i++)

{

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

{

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

}

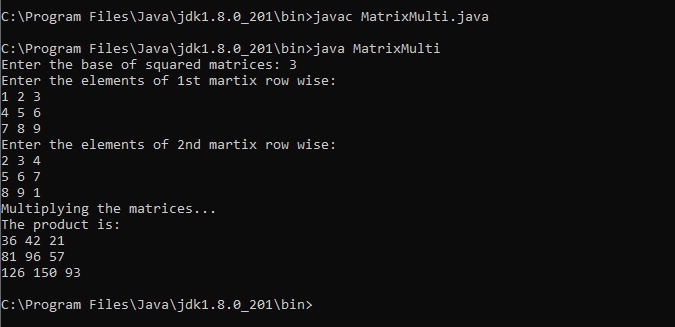
System.out.println();

}

}

}

**Output:**



## : 29-07-2024

**(CJ) Page no:**

**Practical Number:- 6**

# Write a Java program to demonstrate Private access modifier.

## CODE:

class A

{

private int totalValue=50; private int Multiply(int a, int b)

{

return a\*b;

}

}

public class B

{

public static void main(String args[])

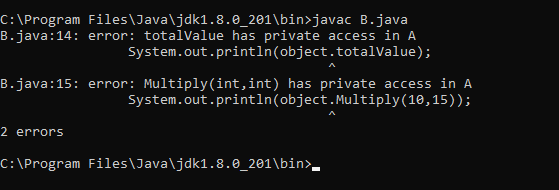
{

}

}

**OUTPUT:**

A object = new A(); System.out.println(object.totalValue); System.out.println(object.Multiply(10,15));



## 

: 29-07-2024

## (CJ) Page no:

**Practical Number:- 6**

# Write a Java program to demonstrate Public access modifier.

## CODE:

class A

{

public int totalValue=50; public int Multiply(int a, int b)

{

return a\*b;

}

}

public class B2

{

public static void main(String args[])

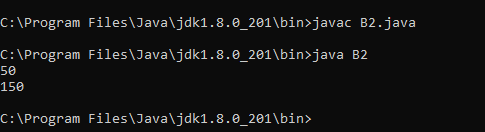
{

}

}

**OUTPUT:**

A object = new A(); System.out.println(object.totalValue); System.out.println(object.Multiply(10,15));



### 

: 29-07-2024

### (CJ) Page no:

**Practical Number:- 07**

# Q1). Write a java program to implement a thread life cycle.

### CODE:

import java.util.Vector; class Mayuresh7B

{

public static void main(String args[])

{

System.out.println(Thread.currentThread().getName()); for(int i=0; i<10;i++)

{

new Thread(" "+i)

{

public void run()

{

}

}

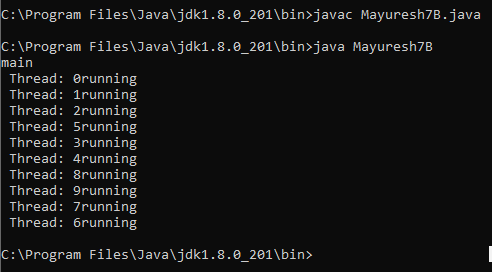
}

**OUTPUT:**

}

}.start();

System.out.println(" Thread:"+ getName()+"running ");



### 

: 29-07-2024

### (CJ) Page no:

**Practical Number:- 07**

# Q2). Write a java program to implement Vector.

### CODE:

import java.util.Vector; class Mayuresh7a1

{

public static void main(String[] args)

{

Vector<String> v = new Vector<String>();

v.add("Red");

v.add("Green");

v.add("Blue");

System.out.println("Vector Elements are: " + v);

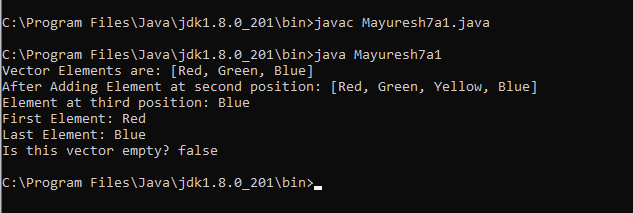
}

}

**OUTPUT:**

v.add(2, "Yellow");

System.out.println("After Adding Element at second position: " + v); System.out.println("Element at third position: " + v.get(3)); System.out.println("First Element: " + v.firstElement()); System.out.println("Last Element: " + v.lastElement()); System.out.println("Is this vector empty? " + v.isEmpty());



### 

: 29-07-2024

### (CJ) Page no:

**Practical Number:- 07**

# Q3). Write a java program to implement Vector and take input from user.

### CODE:

import java.util.Vector; import java.util.Scanner;

class Mayuresh7a

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); int n;

Vector<String> v = new Vector<String>(); System.out.println("How many vector you want to enter: "); n=sc.nextInt();

System.out.println("Enter " + n + " words"); for(int i=0;i<n;i++)

{

String num =sc.next(); v.add(num);

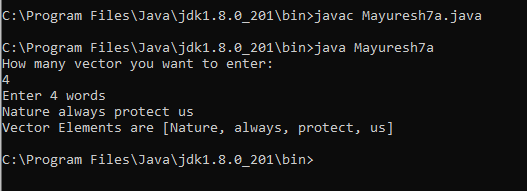
}

System.out.println("Vector Elements are "+v);

}

}

**OUTPUT:**



**(NEP)**

: 02-08-2024

## (CJ) Page no:

**Practical Number: 08**

# Q1). Write a java program to demonstrate inheritance in java.

## CODE:

class Animal

{

void eat()

{

System.out.println("Eating...");

}

}

class Dog extends Animal

{

void bark()

{

System.out.println("Barking...");

}

}

class TestInheritance

{

public static void main(String args[])

{

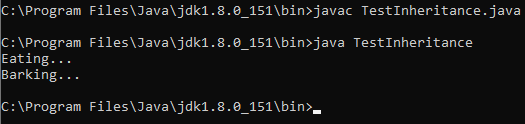
}

}

**OUTPUT:**

Dog d=new Dog(); d.eat();

d.bark();



# Q2). Write a java program to demonstrate multilevel inheritance in java.

## CODE:

class Animal

{

void eat()

{

System.out.println("Eating...");

}

}

class Dog extends Animal

## (NEP)

: 02-08-2024

## (CJ) Page no:

**Practical Number: 08**

{

void bark()

{

System.out.println("Barking...");

}

}

class BabyDog extends Dog

{

void weep()

{

System.out.println("Weeping...");

}

}

class TestInheritance2

{

public static void main(String args[])

{

}

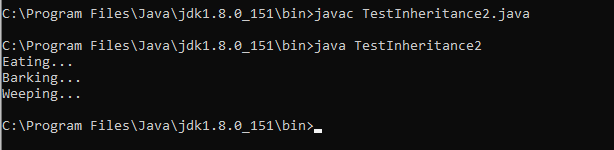
}

**OUTPUT:**

BabyDog d=new BabyDog(); d.eat();

d.bark();

d.weep();



# Q3). Write a java program to demonstrate exception handling.

## CODE:

class main

{

public static void main(String[] args)

{

try

{

}

//code that generates exception int divideByZero = 5/0;

catch(ArithmeticException e)

## (NEP)

: 02-08-2024

## (CJ) Page no:

**Practical Number: 08**

{

}

}

**OUTPUT:**

}

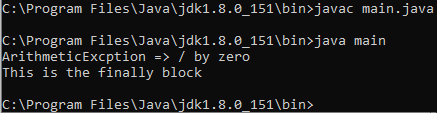
finally

{

}

System.out.println("ArithmeticExcption => "+ e.getMessage());

System.out.println("This is the finally block");



# Q4) Write a java program to demonstrate exception handling and take input from user.

## CODE:

import java.util.Scanner; class ExcepHand

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in); int dividend;

int divisor;

System.out.print("Enter the Dividend: "); dividend = sc.nextInt(); System.out.print("Enter the Divisor: "); divisor = sc.nextInt();

try

{

int result = dividend/divisor;

System.out.println("Division of "+dividend+" and "+divisor+" is "+result);

}

catch(ArithmeticException e)

{

possible");

}

finally

{

System.out.println("Arithmetic Exception -> Division "+ e.getMessage()+" is not

System.out.print("Did you wish to continue? If yes then type 'yes' and if no type 'no': ");

## (NEP)

: 02-08-2024

## (CJ) Page no:

**Practical Number: 08**

String a;

a = sc.next(); while(a.equals("yes"))

{

System.out.print("Enter the Dividend: "); dividend = sc.nextInt(); System.out.print("Enter the Divisor: "); divisor = sc.nextInt();

try

{

int result = dividend/divisor;

System.out.println("Division of "+dividend+" and "+divisor+" is "+result);

}

catch(ArithmeticException e)

{

is not possible");

type 'no': ");

}

finally

{

}

}

System.out.println("Arithmetic Exception -> Division "+ e.getMessage()+"

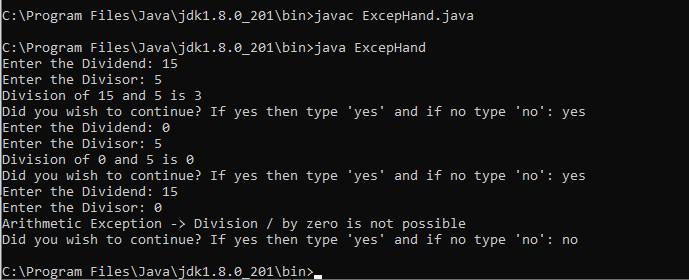
System.out.print("Did you wish to continue? If yes then type 'yes' and if no a = sc.next();

}

}

}

**OUTPUT:**



## : 06-08-2024

**(CJ) 32**

**Practical Number:- 08**

# Write a Java program to demonstrate multiple inheritance using interface.

## CODE:

interface FirstInterface

{

public void myMethod();

}

interface SecondInterface

{

public void myOtherMethod();

}

class DemoClass implements FirstInterface, SecondInterface

{

public void myMethod()

{

System.out.println("Some text...");

}

public void myOtherMethod()

{

System.out.println("Some other text...");

}

}

class Interface

{

public static void main(String[] args)

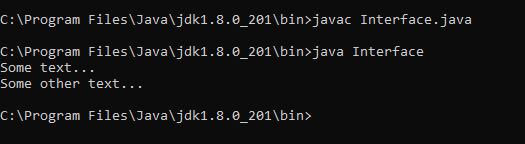
{

}

}

**OUTPUT:**

DemoClass myObj = new DemoClass(); myObj.myMethod(); myObj.myOtherMethod();



# Write a Java program to show ‘this’ and ‘super’ keywords.

## CODE:

class A

## 

: 06-08-2024

## (CJ) 33

**Practical Number:- 08**

{

public int x,y; public A(int x, int y)

{

this.x = x; this.y = y;

}

}

class B extends A

{

public int x,y; public B()

{

this(0,0);

}

public B(int x, int y)

{

super(x + 1, y + 1); this.x = x;

this.y = y;

}

public void print()

{

System.out.println("Base class: {" + x + "," + y + "}"); System.out.println("Super class: {" + super.x + "," + super.y + "}");

}

}

class Point

{

public static void main(String[] args)

{

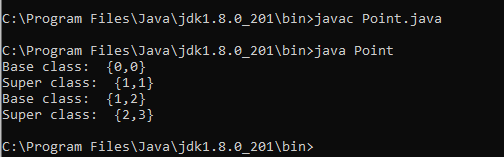
}

}

**OUTPUT:**

B obj = new B(); obj.print();

obj = new B(1, 2); obj.print();



## 

: 07-08-2024

## (CJ) 34

**Practical Number:- 10**

# Write a java program to demonstrate ‘Final Membership’.

## CODE:

class FinalVarDemo

{

final int a = 10; void show()

{

a = 20;

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

}

public static void main(String[] args)

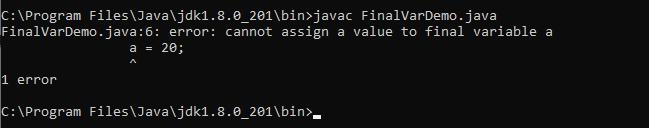
{

}

}

**OUTPUT:**

FinalVarDemo var = new FinalVarDemo(); var.show();



# Write a java program to demonstrate ‘Final Method’.

## CODE:

class Animal

{

final void eat()

{

System.out.println("Animals are eating.");

}

}

public class DeerFinalMethod extends Animal

{

void eat()

{

System.out.println("Deer is eating.");

}

public static void main(String args[])

{

DeerFinalMethod deer = new DeerFinalMethod();

## 

: 07-08-2024

## (CJ) 35

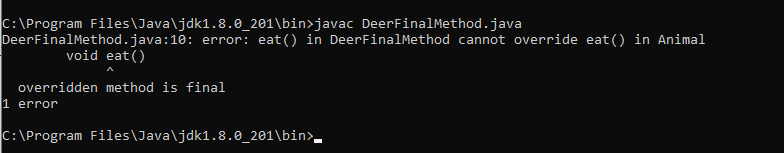
**Practical Number:- 10**

deer.eat();

}

}

**OUTPUT:**



# Write a java program to demonstrate ‘Final Class’.

## CODE:

final class Animal

{

void eat()

{

System.out.println("Animals are eating.");

}

}

class DeerFinalClass extends Animal

{

void eat()

{

System.out.println("Deer is eating.");

}

public static void main(String args[])

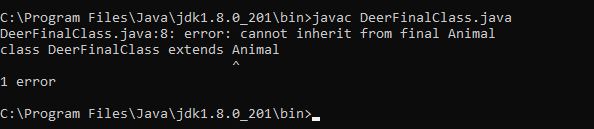
{

}

}

**OUTPUT:**

DeerFinalClass deer = new DeerFinalClass(); deer.eat();



## 

13-08-24

## (CJ) Page no:

**Practical Number:- 11**

# Q1). Write a Java program to open a file and display the text in the console window.

## CODE:

import java.io.\*; public class InExam

{

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

{

InputStream is = new FileInputStream("C:\\Program

Files\\Java\\jdk1.8.0\_201\\bin\\Demo.txt");

DataInputStream dis = new DataInputStream(is); int count = is.available();

byte a[] = new byte[count]; is.read(a);

for(byte b:a)

{

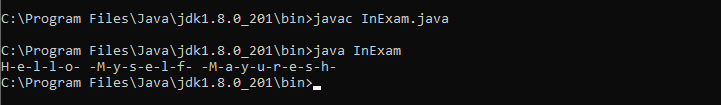
}

}

}

## Output:

char k = (char)b; System.out.print(k+"-");



## 

13-08-24

## (CJ) Page no:

**Practical Number:- 11**

# Q2). Write a Java program to copy the content from one file to another file.

## CODE:

import java.io.\*; public class FileCopy

{

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

{

FileInputStream fis=null; FileOutputStream fos= null; try

{

File inF= new File("C:\\Program Files\\Java\\jdk1.8.0\_201\\bin\\Demo1.txt"); File outF= new File("C:\\Program Files\\Java\\jdk1.8.0\_201\\bin\\Demo2.txt"); fis=new FileInputStream(inF);

fos=new FileOutputStream(outF); byte buff[]= new byte[1024];

int length; while((length=fis.read(buff))>0)

{

fos.write(buff,0,length);

}

fis.close();

fos.close();

System.out.println("The content of the file is successfully coppied...");

}

catch(Exception e)

{

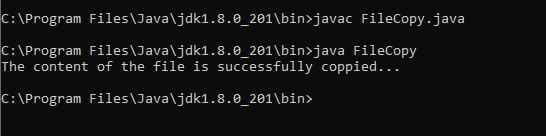
e.printStackTrace();

}

}

}

## Output:



## 13-08-24

**(CJ) Page no:**

## Practical Number:- 11

## 20/08/24

**Page no:**

## Practical Number:- 12

Q1. Demonstrate user-defined Exception with java.

## Code:

import java.lang.Exception;

class MyException extends Exception { MyException(String message) {

super(message);

}

}

class Testing {

public static void main(String[] args) { float a=60, b=10000000;

try {

float c = a/b; if(c < 0.1) {

throw new MyException("The result is too smallllll");

}

}

catch(MyException e) { System.out.println("Exception is caught"); System.out.println(e.getMessage());

}

finally {

System.out.println("Called for nothing");

}

}

}

## Output:



## 20/08/24

**Page no:**

## Practical Number:- 12

Q2. Write Java Program to check whether a number is an Armstrong or not using user-defined exceptions.

## Code:

import java.lang.Exception; import java.util.Scanner;

class MyException extends Exception { MyException(String message) {

super(message);

}

}

class chkarm {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

System.out.print("Enter any number: "); int num = sc.nextInt();

int og = num;

int remainder = 0; int sum = 0;

while(num > 0){ remainder = num % 10;

sum = sum + (remainder\*remainder\*remainder); num /= 10;

}

try {

if(og != sum) {

throw new MyException(og + " is not an Armstrong number.");

}else{

System.out.println(og + " is an Armstrong number. ");

}

}

catch(MyException e) { System.out.println("Exception is caught!"); System.out.println(e.getMessage());

}

finally {

System.out.println("Finally block executed!!!");

}

}

}

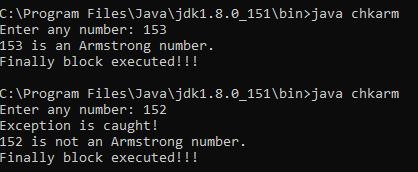
## 

20/08/24

## Page no:

**Practical Number:- 12**

## Output:



## 20/08/24

**Page no:**

## Practical Number:- 12

Q3. Write a java program to read the student data from user and store it in the file.

## Code:

import java.io.\*;

import java.util.Scanner;

public class FileHandling{

public static void main(String[] args) throws FileNotFoundException, IOException { String s1,s2,s3;

Scanner sc = new Scanner(System.in);

System.out.println("Enter name: "); s1 = sc.nextLine();

System.out.println("Enter Phone number: "); s2 = sc.nextLine(); System.out.println("Enter address: ");

s3 = sc.nextLine();

OutputStream fos = new FileOutputStream("FileHandling text.txt"); byte b1[] = s1.getBytes();

fos.write(b1 + " ");

byte b2[] = s2.getBytes(); fos.write(b2 + " ");

byte b3[] = s3.getBytes(); fos.write(b3 + " "); fos.close();

System.out.println("File Created");

}

}

## Output:

