1. **Write a program to demonstrate creating a object and accessing its member:**

class rectangle

{

int length,width;

void getData(int x,int y)

{

length=x;

width=y;

}

int rectarea()

{

int area;

area=length\*width;

return(area);

}

}

class rectArea

{

public static void main(String args[])

{

int area1,area2;

rectangle rect1=new rectangle();

rectangle rect2=new rectangle();

rect1.length=15;

rect1.width=10;

area1=rect1.length\*rect1.width;

rect2.getData(10,20);

area2=rect2.rectarea();

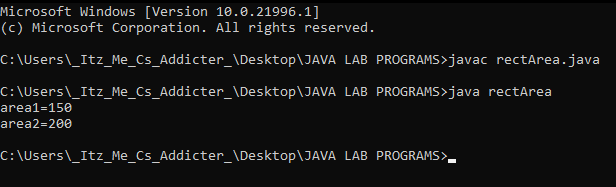
System.out.println("area1="+area1);

System.out.println("area2="+area2);

}

}

**Output:**

****

1. **Write a program to demonstrate default, parameter and copy constructer:**

import java.lang.\*;

class Employee

{

int empno;

String name;

String desig;

float salary;

Employee()

{

empno=1;

name="Nithin";

desig="Software Engineer";

salary=45000;

}

Employee(int eno,String n,String d,float sal)

{

empno=eno;

name=n;

desig=d;

salary=sal;

}

Employee(Employee e)

{

empno=e.empno;

name=e.name;

desig=e.desig;

salary=e.salary;

}

void display()

{

System.out.println("Employee no:"+empno);

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

System.out.println("Designation:"+desig);

System.out.println("Salary:"+salary);

}

}

class EmployeeDemo

{

public static void main(String args[])

{

Employee e1=new Employee();

e1.display();

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

Employee e2=new Employee(2,"koosappa","Doctor",50000);

e2.display();

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

Employee e3=new Employee(e2);

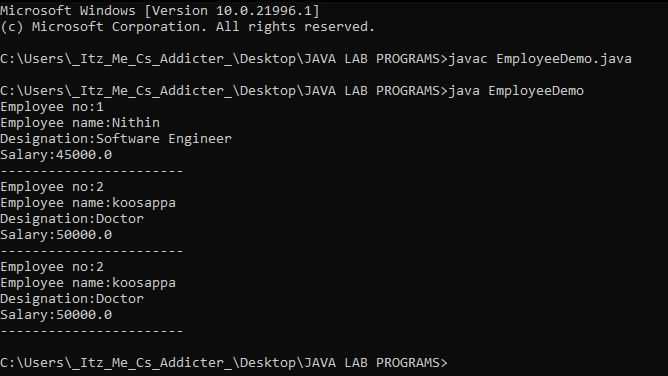
e3.display();

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

}

}

**Output:**

****

1. **Write a program to demonstrate this keyword:**

class Student

{

int rollno;

String name;

float fee;

Student(int rollno,String name,float fee)

{

this.rollno=rollno;

this.name=name;

this.fee=fee;

}

void display()

{

System.out.println(rollno+" "+name+" "+fee);}

}

class TestThis2

{

public static void main(String args[]){

Student s1=new Student(111,"ankit",5000f);

Student s2=new Student(112,"sumit",6000f);

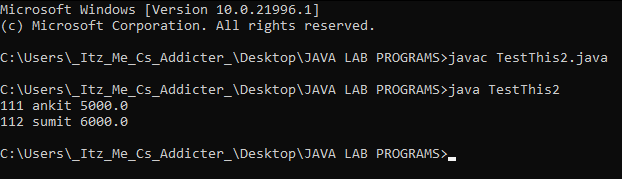
s1.display();

s2.display();

}

}

**Output:**



1. **Write a program to demonstrate Autoboxing:**

public class AutoBoxingExample {

public static void main(String[] args){

byte b = 100;

Byte B = b;

System.out. println(B);

short s = 100;

Short S = s;

System.out.println(S);

int i = 200;

Integer I = i;

System.out.println(I);

long l= 250;

Long L = l;

System.out.println(L);

float f =120L;

Float F = f;

System.out.println(F);

double d = 18.58;

Double D = d;

System.out.println(D);

boolean bln = false;

Boolean BLN = bln;

System.out.println(BLN);

char c = 'C';

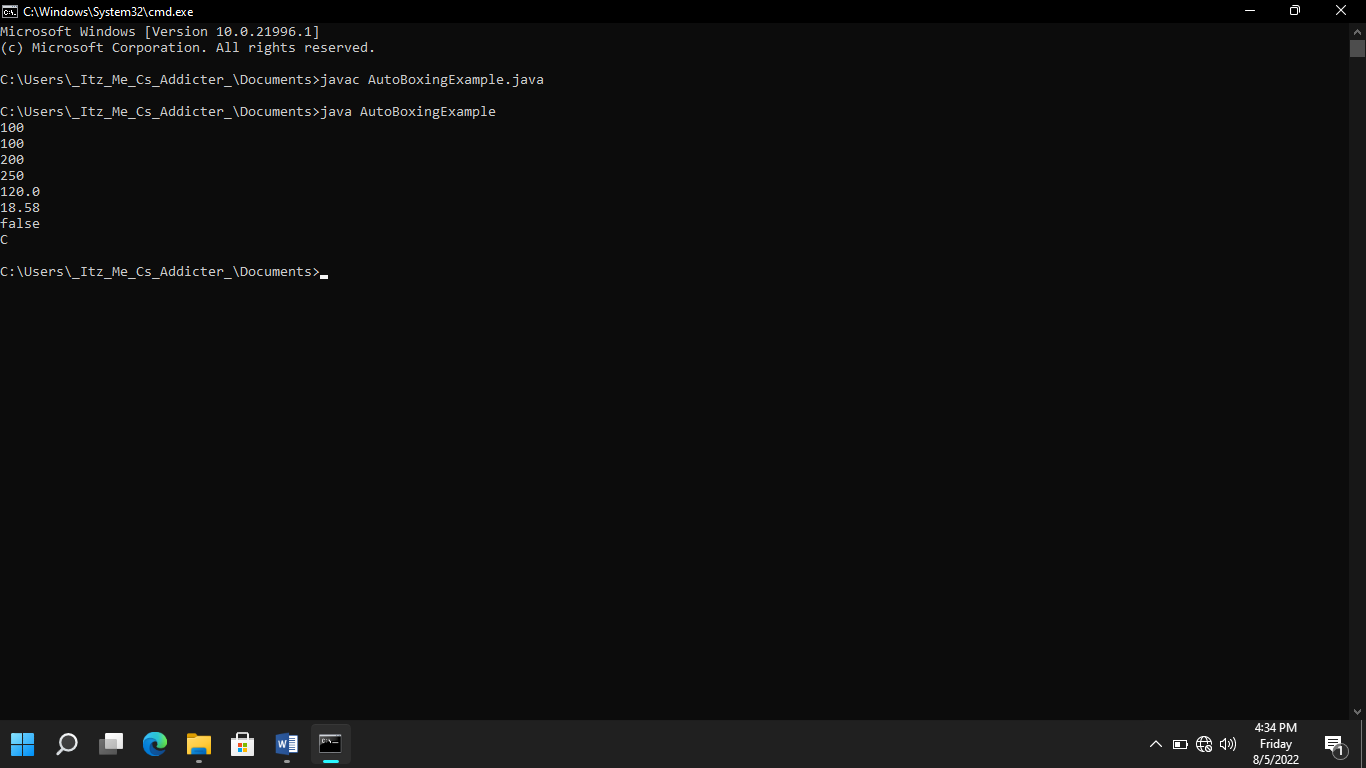
Character C = c;

System.out.println(C);

}

}

**Output:**

****

1. **Write a program to demonstrate AutoUnboxing:**

class AutoUnboxingExample {

public static void main(String[] args) {

Byte B = new Byte((byte) 10);

byte b = B;

System.out.println(b);

Short S = new Short((short) 20);

short s = S;

System.out.println(s);

Integer I = new Integer(15);

int i = I;

System.out.println(i);

Long L = new Long(50);

long l = L;

System.out.println(l);

Float F = new Float(20);

float f = F;

System.out.println(f);

Double D = new Double(20.5);

double d = D;

System.out.println(d);

Boolean BLN = new Boolean(true);

boolean bln = BLN;

System.out.println(bln);

Character C = new Character('C');

char c = C;

System.out.println(c);

}

}

1. **Write a program to demonstrate packages:**
2. package p1;

public class A

{

public void displayA()

{

System.out.println("CLASS A");

}

}

1. package p2;

public class B

{

protected int m=10;

public void displayB()

{

System.out.println("CLASS B");

System.out.println("m="+m);

}

}

class A

{

void display()

{

System.out.println("This is from class A");

}

}

class B extends A

{

void display()

{

System.out.println("This is from class B");

}

}

class AB

{

public static void main(String args[])

{

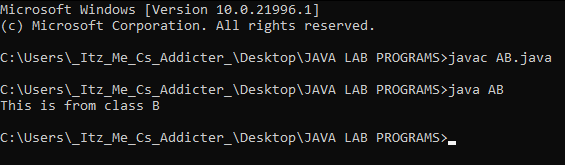
B obj= new B();

obj.display();

}

}

**Output:**

****

1. **Write a program to demonstrate Encapsulation:**

class encapsulationdemo

{

private int ssn;

private String empname;

private int empage;

public int getempssn()

{

return ssn;

}

public String getempname()

{

return empname;

}

public int getempage()

{

return empage;

}

public void setempage(int newvalue)

{

empage=newvalue;

}

public void setempname(String newvalue)

{

empname=newvalue;

}

public void setempssn(int newvalue)

{

ssn=newvalue;

}

}

public class encapstest

{

public static void main(String args [])

{

encapsulationdemo obj=new encapsulationdemo();

obj.setempname("mario");

obj.setempage(32);

obj.setempssn(112233);

System.out.println("employee name:"+obj.getempname());

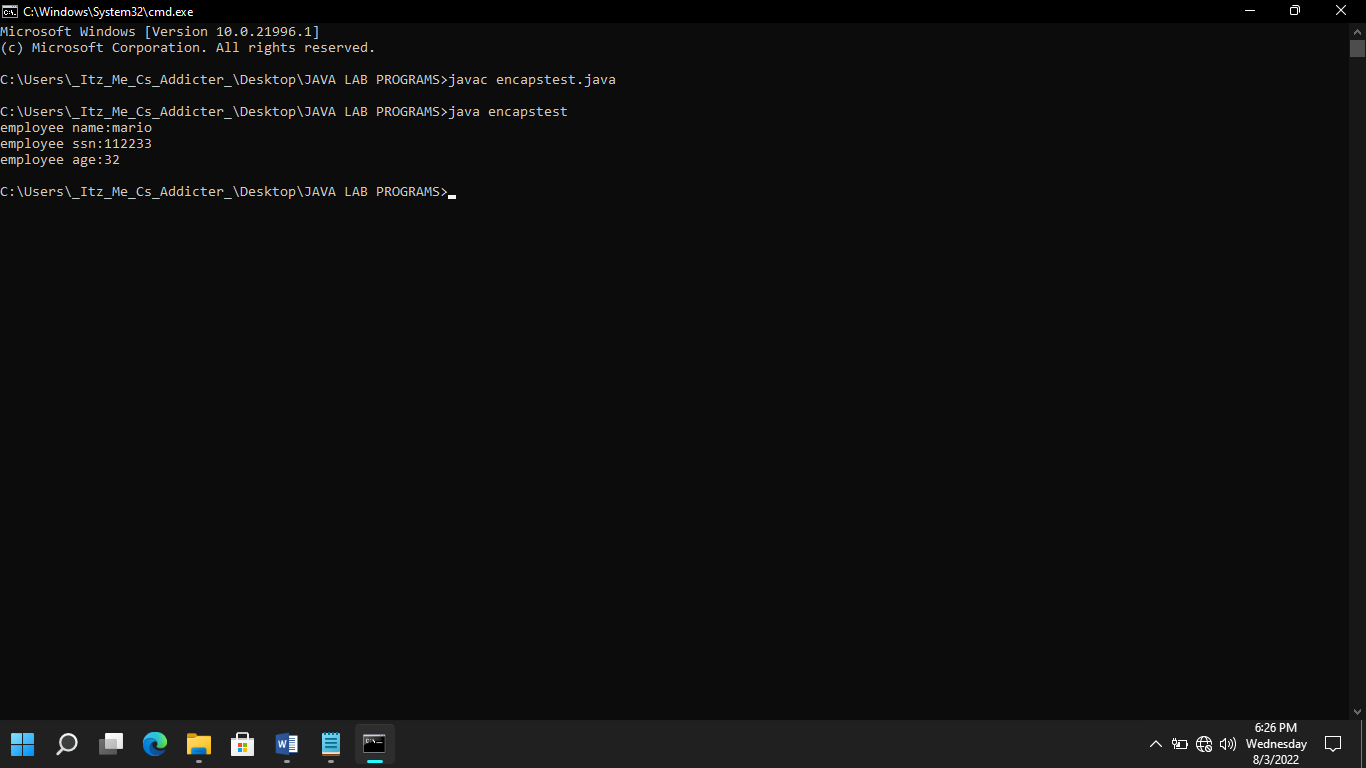
System.out.println("employee ssn:"+obj.getempssn());

System.out.println("employee age:"+obj.getempage());

}

}

**Output:**

****

1. **Write a program to demonstrate Inheritance:**

class Animal

{

void eat()

{

System.out.println("eating...");}

}

class Dog extends Animal

{

void bark()

{

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

}

}

class BabyDog extends Dog

{

void weep()

{

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

}

}

class Inheritance

{

public static void main(String args[])

{

BabyDog d=new BabyDog();

d.weep();

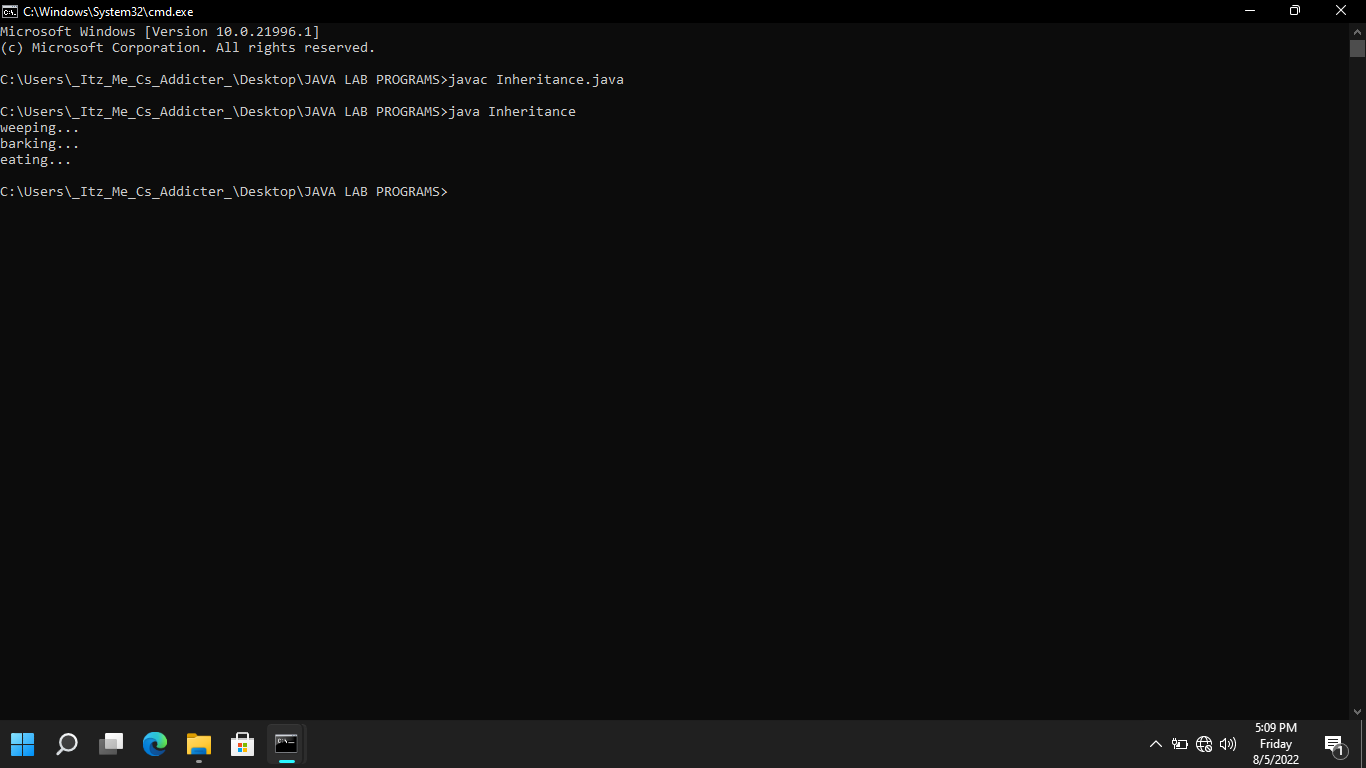
d.bark();

d.eat();

}

}

**Output:**



1. **Write a program to demonstrate sort an Array:**

import java.util.Arrays;

public class sortArrayExample1

{

public static void main(String[]args)

{

int[]array=new int[]{90,23,5,109,12,22,67,34};

Arrays.sort(array);

System.out.println("element of array sorted in ascending order:");

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

{

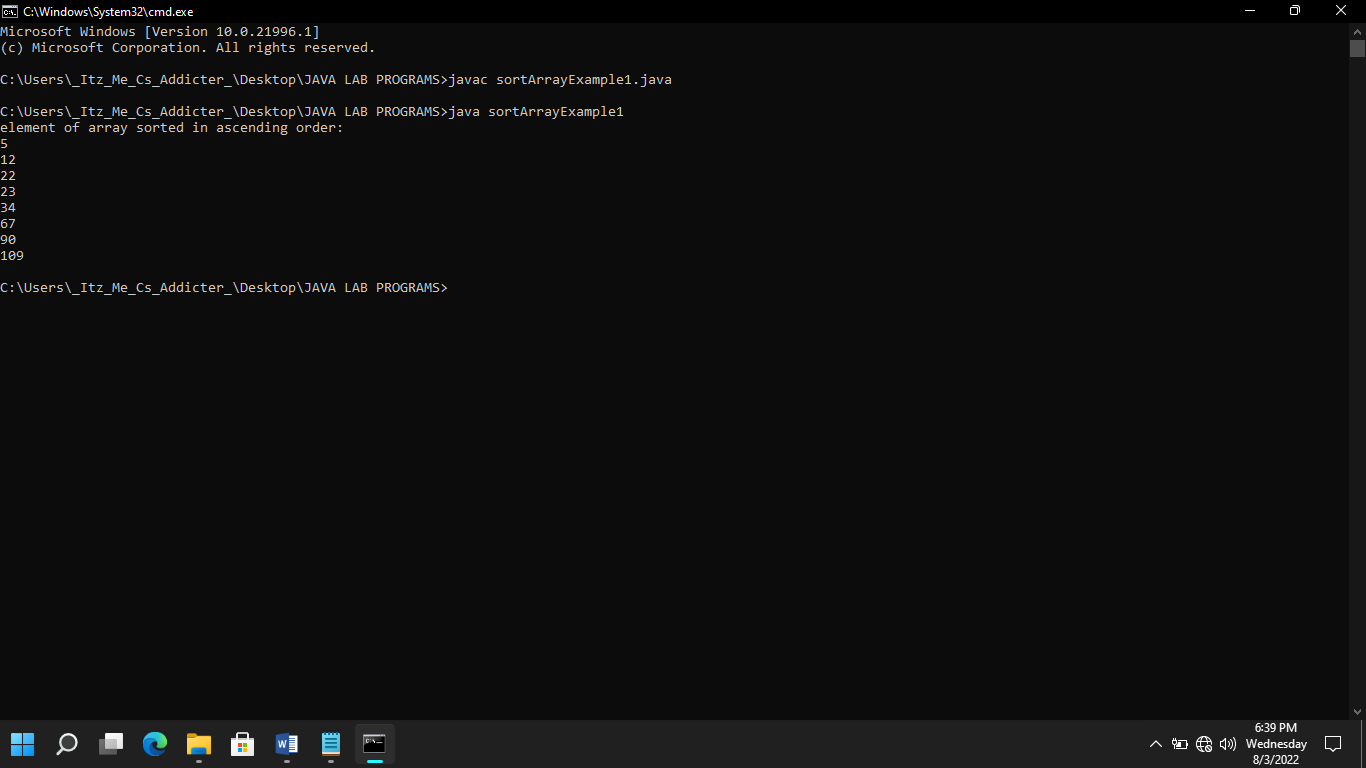
System.out.println(array[i]);

}

}

}

**Output:**

****

1. **Write a program to demonstrate String methods:**

public class demo1

{

public static void main(String[] args)

{

String myStr1 = "Hello";

String myStr2 ="HELLO";

String myStr3 =" welCome";

System.out.println(myStr1.compareTo(myStr2));

System.out.println(myStr1.charAt(0));

System.out.println(myStr1.compareToIgnoreCase(myStr2));

System.out.println(myStr1.concat(myStr2));

System.out.println(myStr1.length());

System.out.println(myStr1.replace( 'l','p'));

System.out.println(myStr1.toUpperCase());

System.out.println(myStr2.toLowerCase());

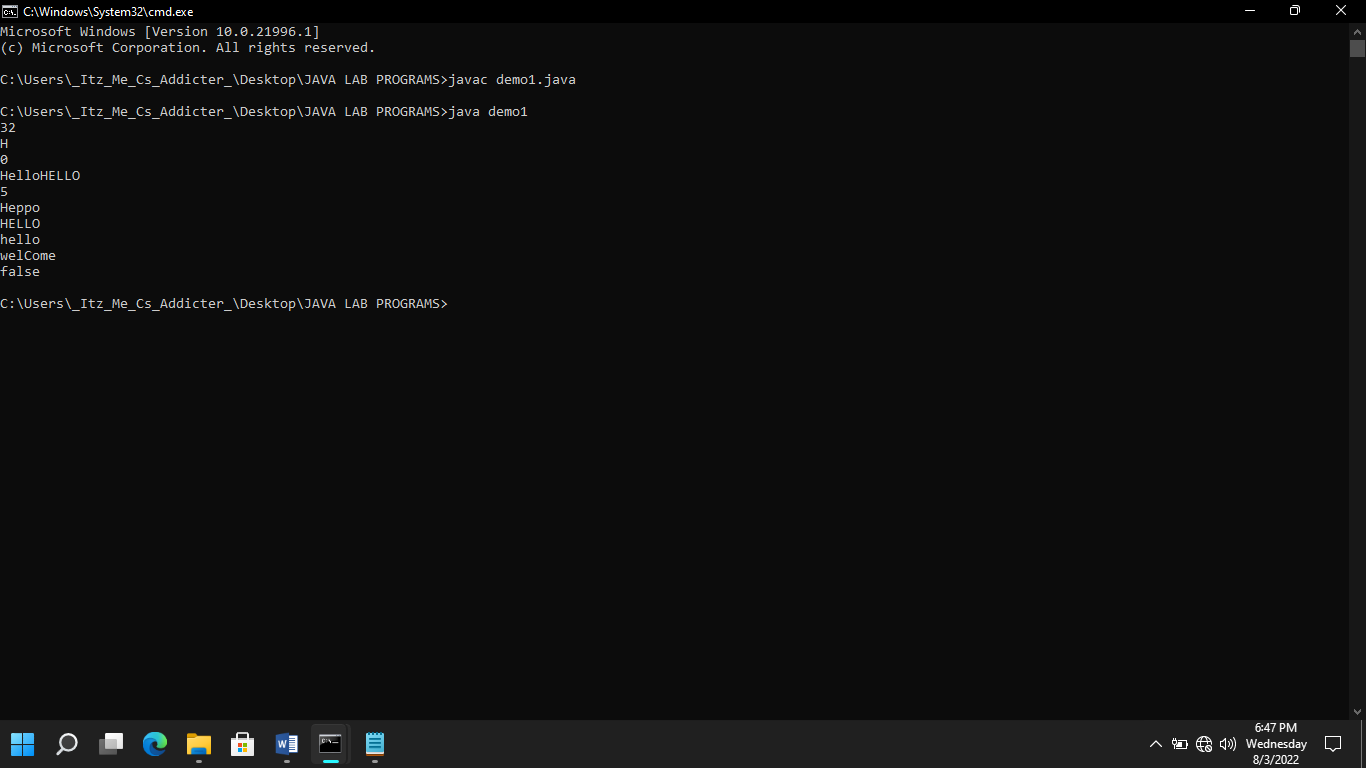
System.out.println(myStr3.trim());

System.out.println(myStr1.contains("Mysore"));

}

}

**Output:**



1. **Write a program to demonstrate Method Overloading:**

class Overload

{

void display(String s)

{

System.out.println(s);

}

void display(int a)

{

System.out.println(a);

}

void display(int a,float b)

{

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

}

void display(double d)

{

System.out.println(d);

}

}

class OverloadDemo

{

public static void main(String args[])

{

Overload obj=new Overload();

obj.display("Welcome to method overloading");

obj.display(123);

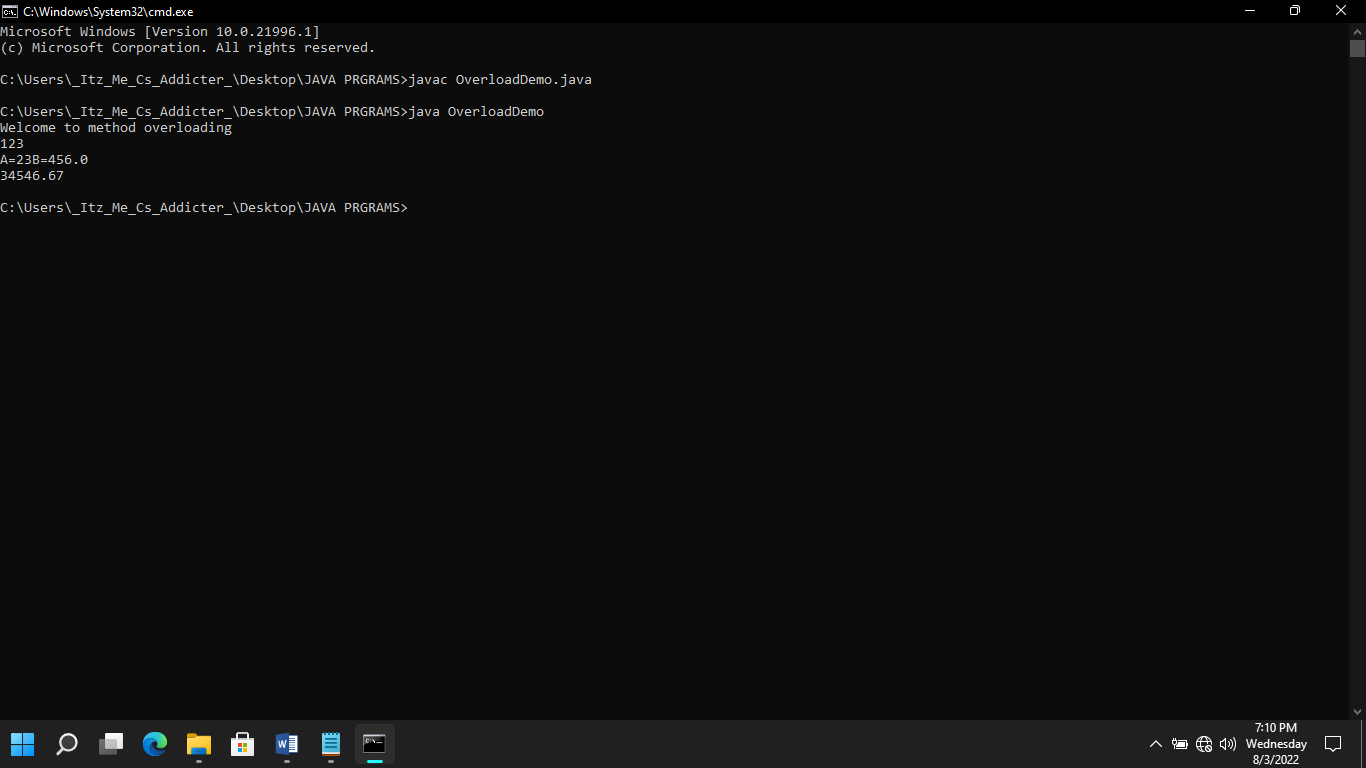
obj.display(23,456.0f);

obj.display(34546.67d);

}

}

**Output:**

****

1. **Write a program to demonstrate Method Overriding:**

class A

{

void display()

{

System.out.println("This is from class A ");

}

}

class B extends A

{

void display()

{

System.out.println("This is from class B ");

}

}

class AB

{

public static void main(String arg[])

{

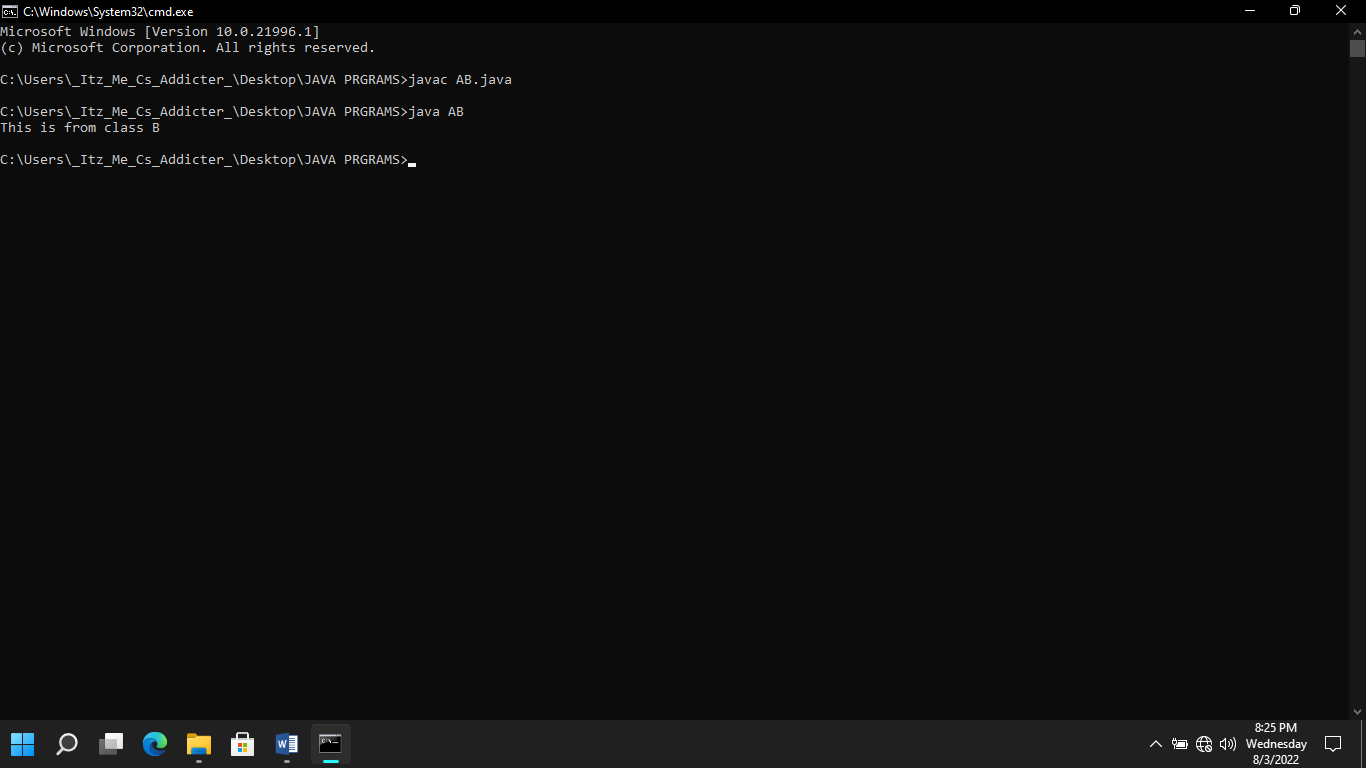
B obj=new B();

obj.display();

}

}

**Output:**

****

1. **Write a program to demonstrate Abstract class and method:**

abstract class Bike

{

abstract void run();

}

class Honda4 extends Bike

{

void run()

{

System.out.println("running safely");

}

public static void main(String args[])

{

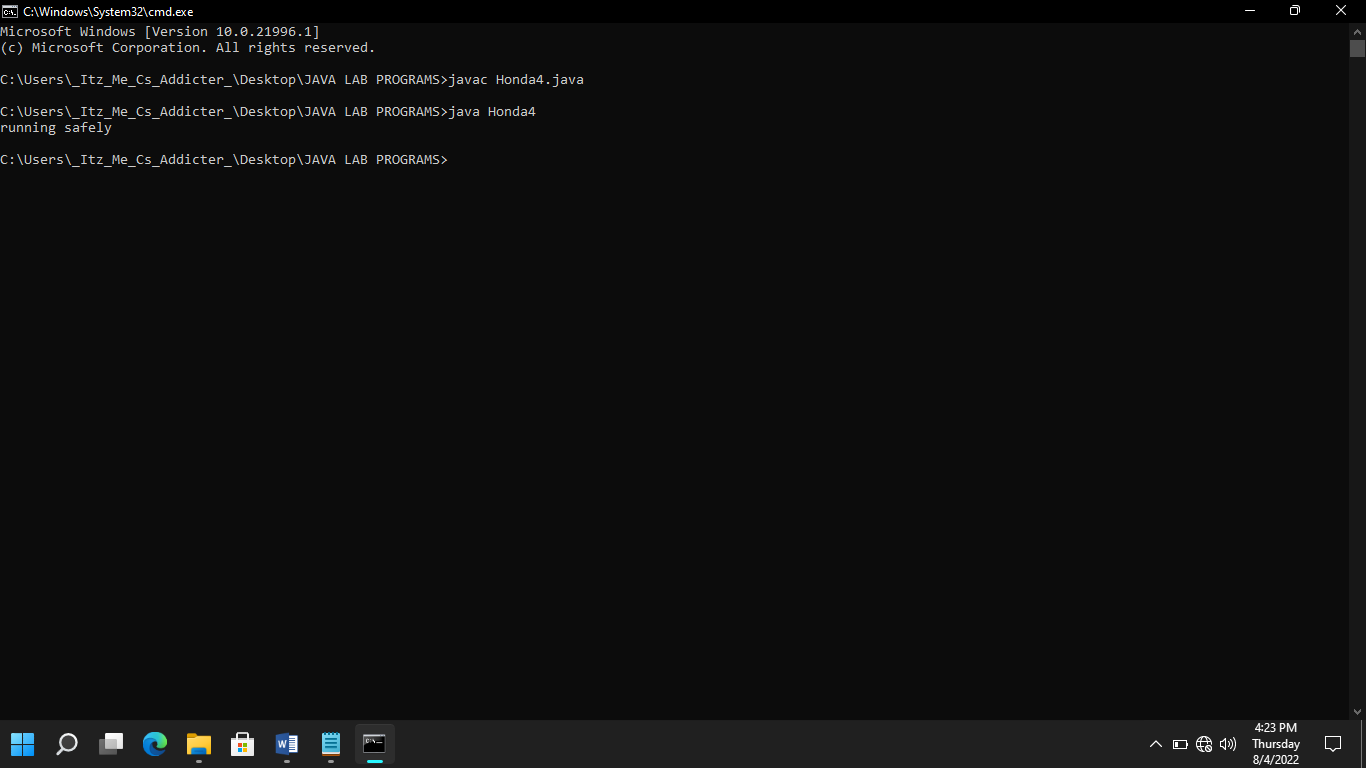
Bike obj = new Honda4();

obj.run();

}

}

**Output:**



1. **Write a program to demonstrate Interfaces:**

interface Animal

{

public void Animalsound();

public void sleep();

}

class pig implements Animal

{

public void Animalsound()

{

System.out.println("the pig says:wee wee");

}

public void sleep()

{

System.out.println("zzz");

}

}

class main

{

public static void main(String args[])

{

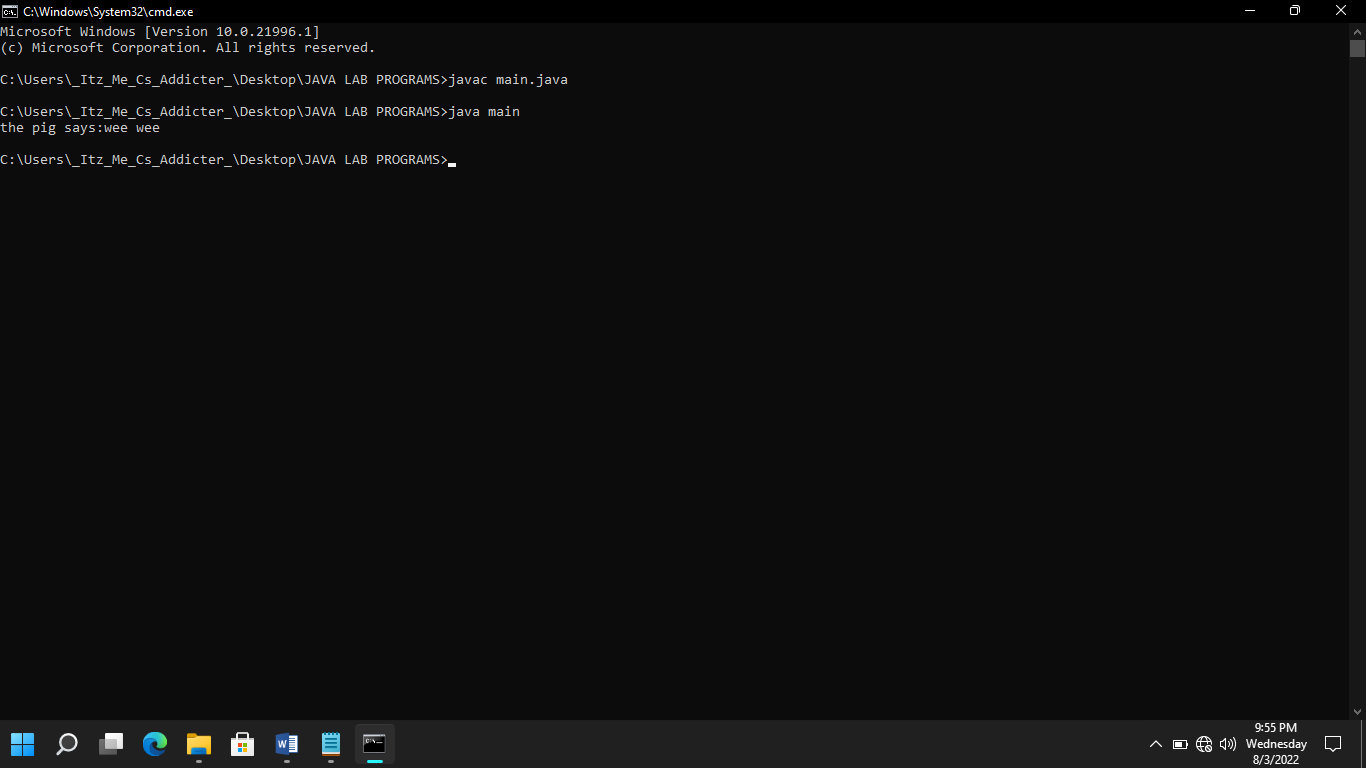
pig Mypig=new pig();

Mypig.Animalsound();

}

}

**Output:**

****

1. **Write a program to demonstrate Create a File:**

import java.io.File;

import java.io.IOException;

class CreateFile

{

public static void main(String args[])

{

try

{

File fo=new File("C:FileOperationExample.txt");

if(fo.createNewFile())

{

System.out.println("File"+fo.getName() +"is created successfully.");

}

else

{

System.out.println("File is already exits in the directory.");

}

}

catch(IOException exception)

{

System.out.println("An unexpected error is occured.");

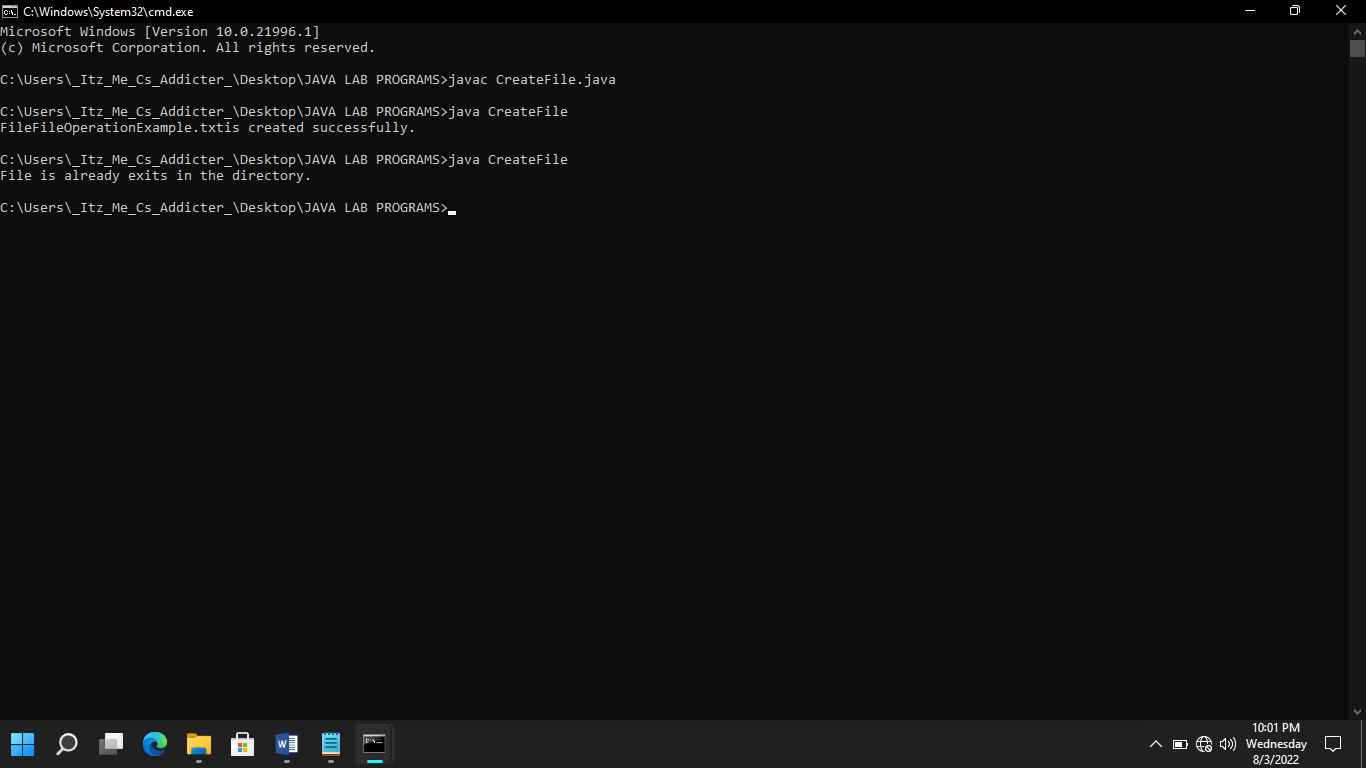
exception.printStackTrace();

}

}

}

**Output:**



1. **Write a program to demonstrate get the information of the file:**

import java.io.File;

class FileInfo

{

public static void main(String[] args)

{

File fo=new File("C:FileOperationExample.txt");

if(fo.exists())

{

System.out.println("The name of the file is:"+fo.getName());

System.out.println("The absolute path of the file is:" + fo.getAbsolutePath());

System.out.println("Is file writeable?:"+fo.canWrite());

System.out.println("Is file readable" + fo.canRead());

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

}

else

{

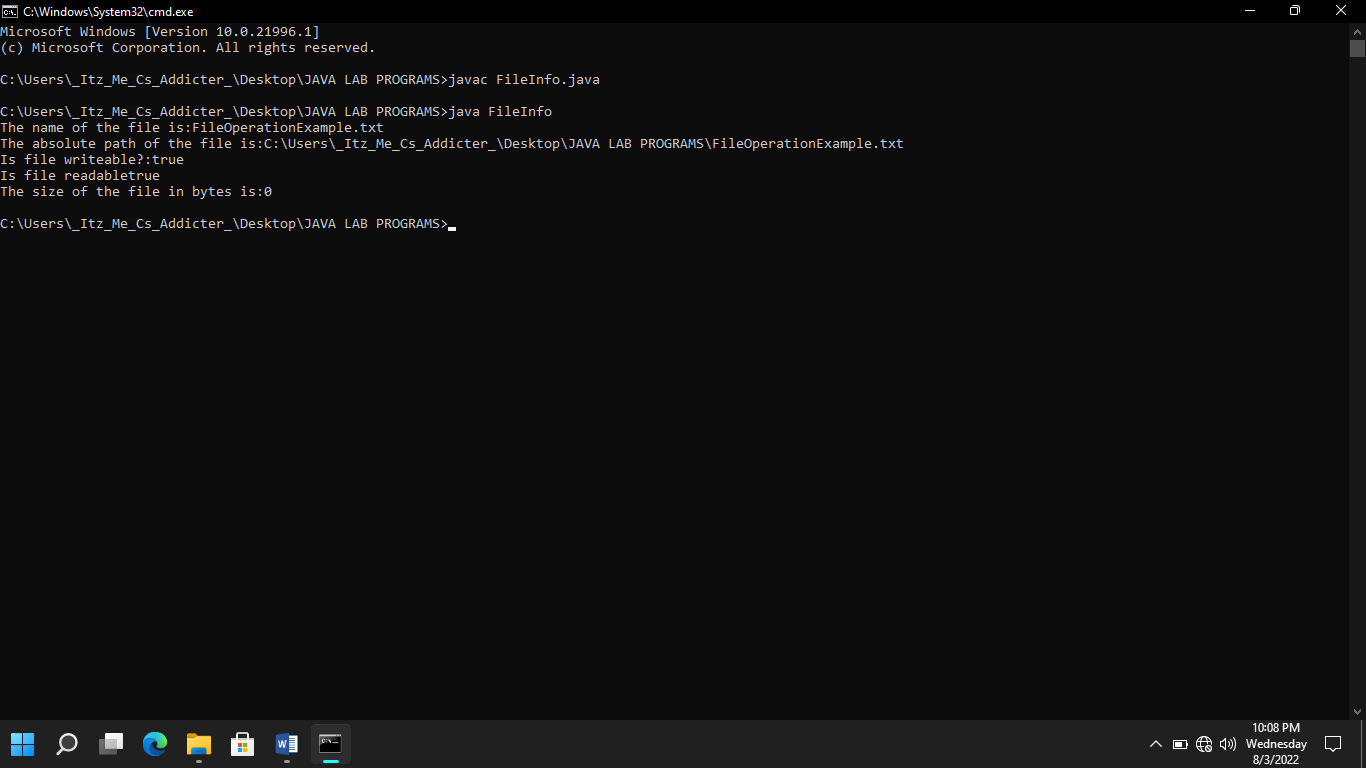
System.out.println("The file does not exists.");

}

}

}

**Output:**



1. **Write a program to demonstrate get the information of the file:**

import java.io.FileWriter;

import java.io.IOException;

class WriteToFile{

public static void main(String[] args){

try{

FileWriter fwrite=new FileWriter("C:FileOperationExample.txt");

fwrite.write("A named location used to store related information is referred to as a File.");

fwrite.close();

System.out.println("Content is successfully wrote to the file.");

}catch(IOException e){

System.out.println("Unexpected error occured");

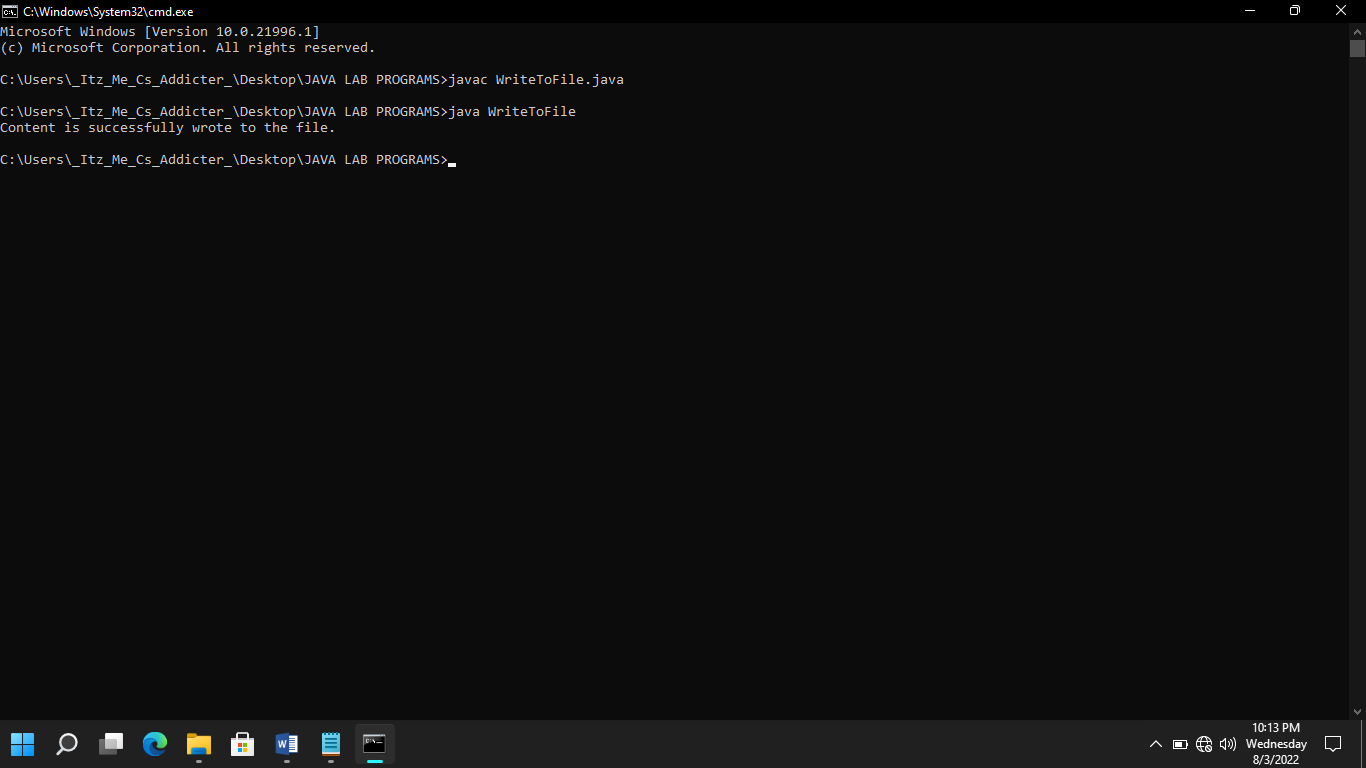
e.printStackTrace();

}

}

}

**Output:**



1. **Write a program to demonstrate Read the content of file:**

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

class ReadFromFile{

public static void main(String[] args){

try{

File f1=new File("C:FileOperationExample.txt");

Scanner dataReader=new Scanner(f1);

while(dataReader.hasNextLine()){

String fileData=dataReader.nextLine();

System.out.println(fileData);

}

dataReader.close();

}catch(FileNotFoundException exception){

System.out.println("Unexcepted error occured!");

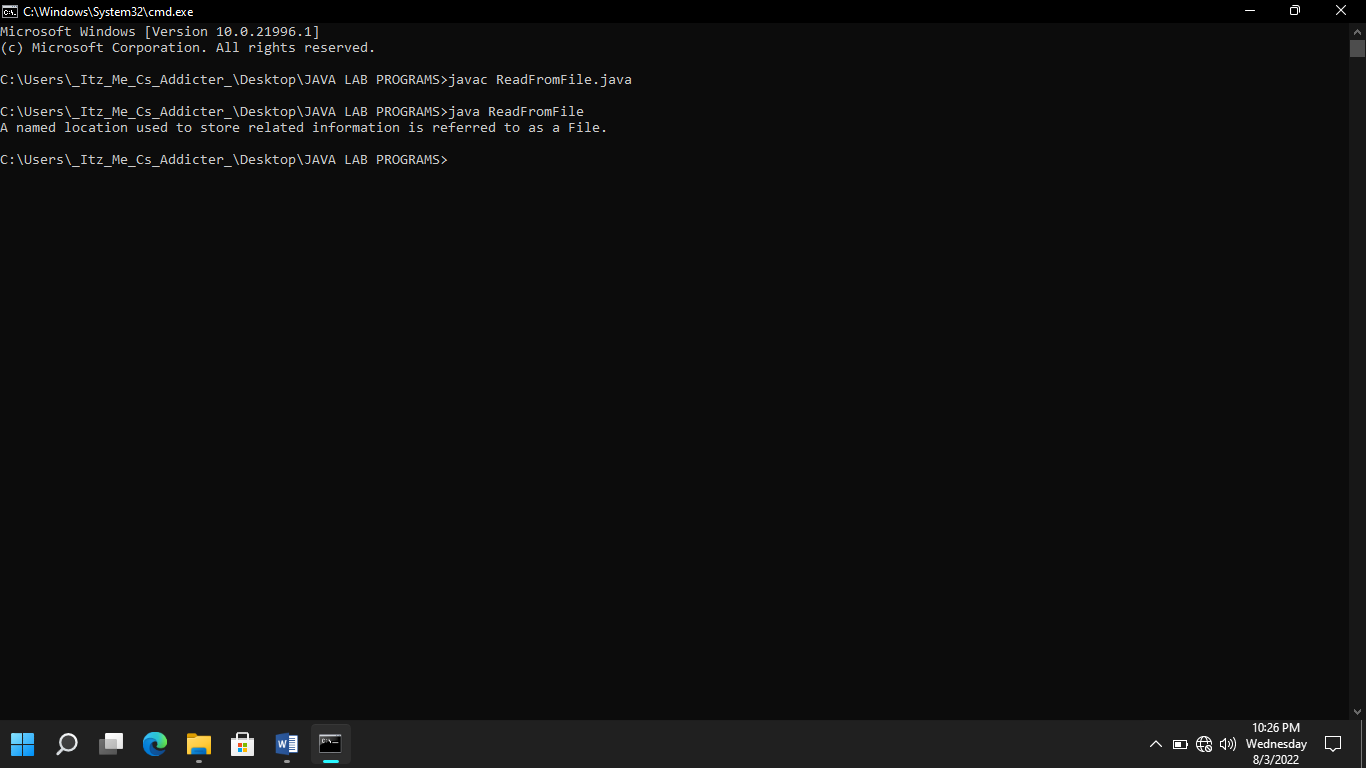
exception.printStackTrace();

}

}

}

**Output:**



1. **Write a program to demonstrate try and catch statement:**

class ERROR3

{

public static void main (String args[])

{

int a[]={5,10};

int b=5;

try

{

int x=a[2]/b-a[1];

}

catch (ArithmeticException e)

{

System.out.println("division by zero");

}

catch (ArrayIndexOutOfBoundsException e)

{

System.out.println("ArrayIndexError");

}

catch (ArrayStoreException e)

{

System.out.println("Wrong data type");

}

{

int y=a[1]/a[0];

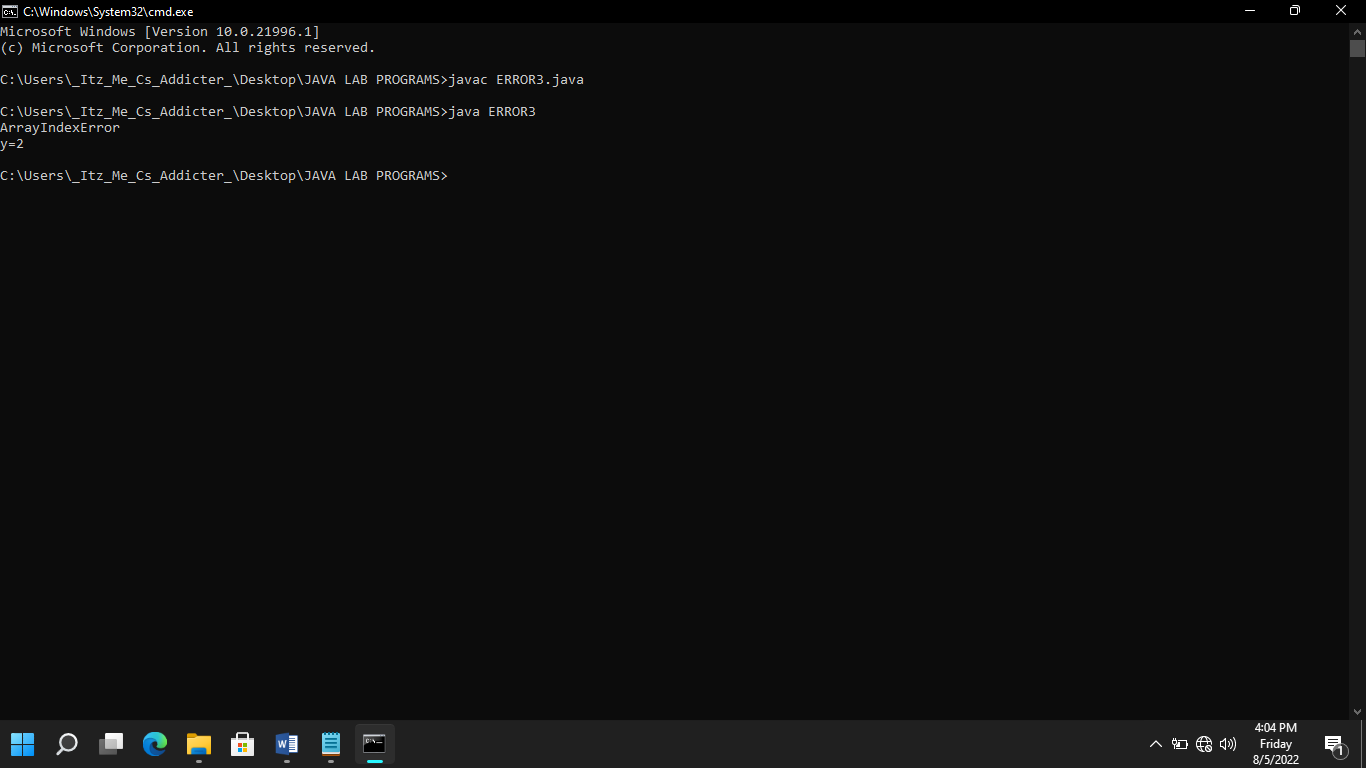
System.out.println("y="+y);

}

}

}

**Output:**



1. **Write a program to demonstrate try and catch statement:**

class ERROR4

{

public static void main (String args[])

{

int a[]={5,10};

int b=5;

try

{

int x=a[2]/b-a[1];

}

catch (ArithmeticException e)

{

System.out.println("division by zero");

}

catch (ArrayIndexOutOfBoundsException e)

{

System.out.println("ArrayIndexError");

}

catch (ArrayStoreException e)

{

System.out.println("Wrong data type");

}

finally

{

int y=a[1]/a[0];

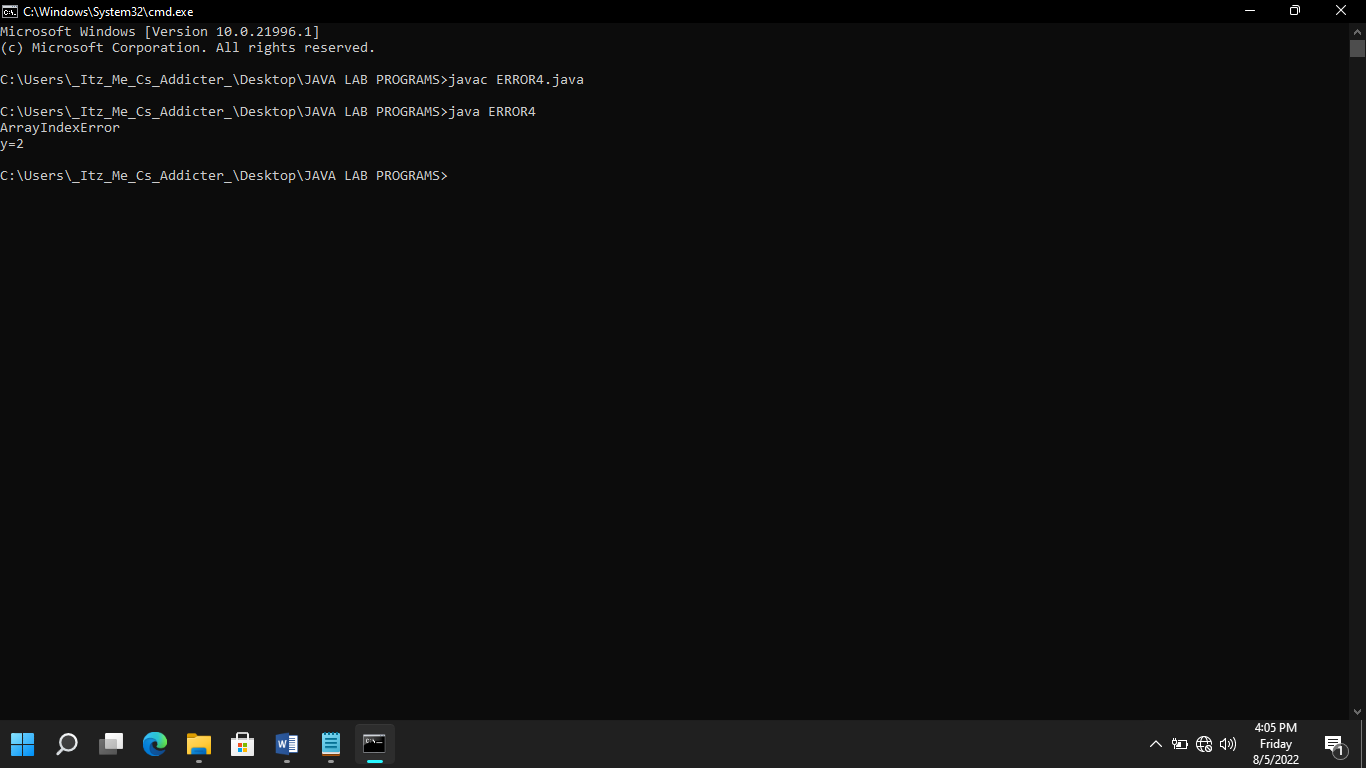
System.out.println("y="+y);

}

}

}

**Output:**

****