**EXPERIMENT 1: Constructor Overloading**

**SOURCE CODE:**

class Box{

double width;

double height;

double depth;

//This is constructor for Box

Box(double w, double h, double d){

width=w;

height=h;

depth=d;

}

Box(){

width=-1;

height=-1;

depth=-1;

}

Box(double len){

width=height=depth=len;

}

double volume(){

return width\*height\*depth;

}

}

class OverloadCons{

public static void main(String args[]){

Box mybox1=new Box(10,20,15);

Box mybox2=new Box();

Box mybox3=new Box(7);

double vol;

vol=mybox1.volume();

System.out.println("Volume of mybox1 is " + vol);

vol=mybox2.volume();

System.out.println("Volume of mybox2 is " + vol);

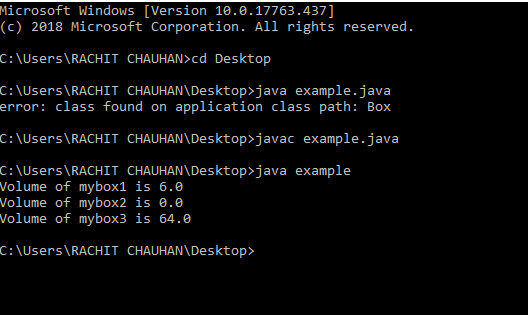
vol=mybox3.volume();

System.out.println("Volume of mybox3 is " + vol);

}

}

**OUTPUT:**



**EXPERIMENT 2: Method Overloading and Overriding**

1. **Method Overloading**

**SOURCE CODE:**

class example {

int sum(int x, int y)

{

return (x + y);

}

int sum(int x, int y, int z)

{

return (x + y + z);

}

}

class hello{

public static void main(String args[])

{

example obj = new example();

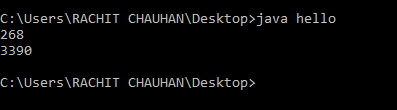
System.out.println(obj.sum(123,145));

System.out.println(obj.sum(1120, 1140, 1130));

System.out.println(obj.sum(105.5, 205.5));

}

}



1. **Method Overriding**

**SOURCE CODE:**

class A{

void show(int a){

System.out.println("hello");

}

}

class B extends A{

void show(int b){

System.out.println("world");

}

}

class hello{

public static void main(String args[]){

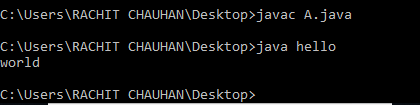
A obj= new B();

obj.show(10);

}

}

**OUTPUT:**



**EXPERIMENT 3: Encapsulation and Inheritance**

1. **Encapsulation**

**SOURCE CODE:**

public class Encap {

private String name;

private String idNum;

private int age;

public int getAge() {

return age;

}

public String getName() {

return name;

}

public String getIdNum() {

return idNum;

}

public void setAge( int newAge) {

age = newAge;

}

public void setName(String newName) {

name = newName;

}

public void setId( String newId) {

id = newId;

}

}

public class run {

public static void main(String args[]) {

Encap obj = new Encap();

obj.setName("Rachit");

obj.setAge(19);

obj.setId("384728732");

System.out.print("Name : " + obj.getName() + " Age : " + obj.getAge() ); }

**OUTPUT:**



1. **Inheritance**

**SOURCE CODE:**

class Season{

void seasonal()

{

System.out.println("allseason");

}

}

class Fruits extends Season{

void Frui()

{

System.out.println("Fruitsseason");

}

}

class Vegetables extends Season{

void Veges()

{

System.out.println("Vegesseason");

}

}

class Inheritance{

public static void main(String args[]){

Vegetables d = new Vegetables();

Fruits d1 = new Fruits();

d.seasonal();

d1.Frui();

d.Veges();

}

}

**OUTPUT:**

**EXPERIMENT 5: Exception and Thread**

1. **Exception**

**SOURCE CODE:**

import java.util.\*;

class Main

{

public static void main(String args[])

{

Scanner reader = new Scanner(System.in);

int a,d;

try

{

a = reader.nextInt();

d = a/0;

}

catch(ArithmeticException e)

{

e.printStackTrace(); //prints exception as if the error message if no exception handling

}

catch(InputMismatchException e)

{

//e.getMessage(); =>Only works for User Defined Exceptions

System.out.println(e); //best way => prints only the type of exception

}

}

}

1. **THREAD**
2. **Through Extending Thread class**

**SOURCE CODE:**

class Multi extends Thread{

public void run(){

System.out.println("thread is running...");

}

public static void main(String args[]){

Multi t1=new Multi();

t1.start();

}

}

1. **Through Runnable Interface**

**SOURCE CODE:**

class Multi3 implements Runnable{

public void run(){

System.out.println("thread is running...");

}

public static void main(String args[]){

Multi3 m1=new Multi3();

Thread t1 =new Thread(m1);

t1.start();

}

}

**EXPERIMENT 7: Input Output in Java**

**SOURCE CODE:**

1. **FileWriter**

**SOURCE CODE:**

import java.io.FileWriter;

import java.io.IOException;

class CreateFile

{

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

{

// Accept a string

String str = "File Handling in Java using "+

" FileWriter and FileReader";

// attach a file to FileWriter

FileWriter fw=new FileWriter("output.txt");

// read character wise from string and write

// into FileWriter

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

fw.write(str.charAt(i));

System.out.println("Writing successful");

//close the file

fw.close();

}

}

1. **FileReader**

**SOURCE CODE:**

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

class ReadFile

{

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

{

// variable declaration

int ch;

// check if File exists or not

FileReader fr=null;

try

{

fr = new FileReader("text");

}

catch (FileNotFoundException fe)

{

System.out.println("File not found");

}

// read from FileReader till the end of file

while ((ch=fr.read())!=-1)

System.out.print((char)ch);

// close the file

fr.close();

}

}