**HTML & CSS programs:**

**1)**

<!DOCTYPE html>

<html>

<head>

<style type="text/css">

p{

text-align: center;

color: blue;

}

</style>

</head>

<body>

<p>This style will be applied on every paragraph.</p>

<p id="para1">Me too!</p>

<p>And me!</p>

</body>

</html>

**2)**

<!DOCTYPE html>

<html>

<head>

<style>

#para1 {

text-align: center;

color: blue;

}

</style>

</head>

<body>

<p>This style will be applied on every paragraph.</p>

<p id="para1">Me too!</p>

<p>And me!</p>

</body>

</html>

**3)**

<!DOCTYPE html>

<html>

<head>

<style>

.center {

text-align: center;

color: blue;

}

</style>

</head>

<body>

<p class="center">This style will be applied on every paragraph.</p>

<p id="para1">Me too!</p>

<p class="center">And me!</p>

</body>

</html>

**4)**

<!DOCTYPE html>

<html>

<head>

<style>

p.center {

text-align: center;

color: blue;

}

</style>

</head>

<body>

<p class="center">This style will be applied on every paragraph.</p>

<p id="para1">Me too!</p>

<p class="center">And me!</p>

<h1 class="center">Welcome</h1>

</body>

</html>

**5)**

<!DOCTYPE html>

<html>

<head>

<style>

\*{

color: #0033ff;

font-size: 20px;

}

</style>

</head>

<body>

<h2>This is heading</h2>

<p>This style will be applied on every paragraph.</p>

<p id="para1">Me too!</p>

<p>And me!</p>

</body>

</html>

**SQL QUERIES**

create table employee(eid number(4),ename varchar2(15),salary number(6),city varchar2(15),state varchar2(15));

insert into employee values(1,'viju',50000,'cbe','tn');

insert into employee values(2,'vini',64000,'trichy','tn');

insert into employee values(3,'nithi',70000,'cbe','tn');

insert into employee values(4,'vasu',66000,'salem','tn');

insert into employee values(5,'yamu',80000,'che','tn');

delete from employee;

insert into employee values(&eid,'&ename',&salary,'&city','&state');

select \* from employee;

select \* from employee where eid = 1;

select \* from employee where city = 'cbe';

select \* from employee where salary > 65000;

update employee set city='banglore' where eid=1;

update employee set city='&city' where eid=&eid;

delete from employee where eid=&eid;

delete from employee where salary > &salary;

delete from employee;

select \* from employee where city = 'cbe';

select \* from employee where city != 'cbe';

select \* from employee where city <> 'cbe';

select \* from employee where city in ('cbe');

select \* from employee where city in ('cbe','che','salem');

select \* from employee where state='kn' and city in ('cbe','che','bang');

select \* from employee where city not in ('cbe');

select \* from employee where eid between 1 and 5;

select \* from employee where eid not between 4 and 7;

select \* from employee where eid > 3 and eid al< 7 ;

select \* from employee where ename = 'hari' and (city='cbe' or city='che' or city='salem');

select \* from employee where ename = 'hari' and (city='cbe' and state='tn');

select \* from employee where ename = 'hari' and ((city='cbe' and state='tn') or city = 'salem' );

select count(city) ,city ,state from employee group by city,state;

select count(city) ,city from employee group by city;

**ENCAPSULATION:**

1.class Area {

int length;

int breadth;

// constructor to initialize values

Area(int length, int breadth) {

this.length = length;

this.breadth = breadth;

}

public void getArea() {

int area = length \* breadth;

System.out.println("Area: " + area);

}

}

class Main {

public static void main(String[] args) {

Area rectangle = new Area(2, 16);

rectangle.getArea();

}

}

2.class Name {

private int age; // Private is using to hide the data

public int getAge() {

return age;

}

public void setAge(int age)

{

this.age = age;

}

}

class GFG {

public static void main(String[] args)

{

Name n1 = new Name();

n1.setAge(19);

System.out.println("The age of the person is: "+ n1.getAge());

}

}

3.

class Account {

private long acc\_no;

private String name,email;

private float amount;

public long getAcc\_no() {

return acc\_no;

}

public void setAcc\_no(long acc\_no) {

this.acc\_no = acc\_no;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public float getAmount() {

return amount;

}

public void setAmount(float amount) {

this.amount = amount;

}

}

public class GFG {

public static void main(String[] args) {

Account acc=new Account();

acc.setAcc\_no(7310805450L);

acc.setName("MD FAIZ");

acc.setEmail("mdfaiz689@gmail.com");

acc.setAmount(100000f);

System.out.println(acc.getAcc\_no()+" "+acc.getName()+" "+acc.getEmail()+" "+acc.getAmount());

}

}

POLYMORPHISM

class Helper {

static int Multiply(int a, int b)

{

// Returns product of integer numbers

return a \* b;

}

static double Multiply(double a, double b)

{

// Returns product of double numbers

return a \* b;

}

}

class GFG {

public static void main(String[] args)

{

System.out.println(Helper.Multiply(2, 4));

System.out.println(Helper.Multiply(5.5, 6.3));

}

}

class Helper {

static int Multiply(int a, int b)

{

// Return product

return a \* b;

}

static int Multiply(int a, int b, int c)

{

// Return product

return a \* b \* c;

}

}

class GFG {

public static void main(String[] args)

{

System.out.println(Helper.Multiply(2, 4));

System.out.println(Helper.Multiply(2, 7, 3));

}

}

**Method Overriding**

class Parent {

void Print()

{

// Print statement

System.out.println("parent class");

}

}

class subclass1 extends Parent {

void Print() {

System.out.println("subclass1");

}

}

class subclass2 extends Parent {

void Print()

{

System.out.println("subclass2");

}

}

class GFG {

public static void main(String[] args)

{

Parent a;

a = new subclass1();

a.Print();

a = new subclass2();

a.Print();

}

}

ABSTRACTION AND INTERFACE

//Abstract Class

abstract class Car

{

abstract void door();

abstract void glass();

}

class Benz extends Car

{

void door()

{

System.out.println(" Benz door");

}

void glass()

{

System.out.println(" Benz glass");

}

}

class Demo1

{

public static void main(String aa[])

{

Car ob1=new Benz();

ob1.door();

ob1.glass();

}

}

//Abstract Class

abstract class Car

{

abstract void door();

abstract void glass();

void wheel()

{

System.out.println(" Wheel");

}

}

class Benz extends Car

{

void door()

{

System.out.println(" Benz door");

}

void glass()

{

System.out.println(" Benz glass");

}

}

class Lancer extends Car

{

void door()

{

System.out.println(" Lancer door");

}

void glass()

{

System.out.println(" Lancer glass");

}

}

class Demo2

{

public static void main(String aa[])

{

Car ob1=new Benz();

ob1.door();

ob1.glass();

ob1.wheel();

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

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

Car ob2=new Lancer();

ob2.door();

ob2.glass();

ob2.wheel();

}

}

######################################################################

abstract class Car

{

abstract void door();

abstract void glass();

}

abstract class Benz extends Car

{

abstract void test();

}

class Sample extends Benz

{

void door()

{

System.out.println(" Sample door");

}

void glass()

{

System.out.println(" Sample glass");

}

void test()

{

System.out.println(" Sample Test");

}

}

class Demo3

{

public static void main(String aa[])

{

Sample ob=new Sample();

ob.door();

ob.glass();

ob.test();

}

}

######################################################################

interface Mail

{

void register();

void valid();

}

class Yahoo implements Mail

{

public void register()

{

System.out.println(" Yahoo registration");

}

public void valid()

{

System.out.println(" Yahoo validation");

}

}

class Demo4

{

public static void main(String aa[])

{

Mail ob1=new Yahoo();

ob1.register();

ob1.valid();

}

}

######################################################################

interface Mail1

{

void register();

}

interface Mail2

{

void register();

}

class Yahoo implements Mail1,Mail2

{

public void register()

{

System.out.println(" Yahoo registration");

}

}

class Demo5

{

public static void main(String aa[])

{

Mail1 ob1=new Yahoo();

Mail2 ob2=new Yahoo();

ob1.register();

ob2.register();

}

}

######################################################################

interface Mail

{

void register();

}

class Test

{

void display()

{

Mail ob = new Mail()

{

public void register()

{

System.out.println(" Anonymous Sub-Class ");

}

};

ob.register();

}

}

class Demo6

{

public static void main(String aa[])

{

new Test().display();

}

}

###################################################

interface Mail

{

void register();

void valid();

}

class Demo6

{

public static void main(String aa[])

{

Mail ob1=new Mail()

{

public void register()

{

System.out.println(" Unknown registration");

}

public void valid()

{

System.out.println(" Unknown validation");

}

};

ob1.register();

ob1.valid();

}

}

######################################################################

interface Mail1

{

void register();

}

class Sample

{

void test()

{

new Mail1()

{

public void register()

{

System.out.println(" Anonymous registration ");

}

}.register();

}

}

interface Mail2

{

void register();

}

class Yahoo implements Mail1,Mail2

{

public void register()

{

System.out.println(" Yahoo registration ");

}

}

class Demo5

{

public static void main(String aa[])

{

Mail1 ob1=new Yahoo();

Mail2 ob2=new Yahoo();

ob1.register();

ob2.register();

new Sample().test();

}

}

INHERITANCE

//INHERITANCE

class Animal {

String name;

public void eat() {

System.out.println("I can eat");

}

}

class Dog extends Animal {

public void display() {

System.out.println("My name is " + name);

}

}

class Main {

public static void main(String[] args) {

Dog labrador = new Dog();

labrador.name = "Rohu";

labrador.display();

labrador.eat();

}

}

class Animal {

public void eat() {

System.out.println("I can eat");

}

}

class Dog extends Animal {

@Override

public void eat() {

System.out.println("I eat dog food");

}

public void bark() {

System.out.println("I can bark");

}

}

class Main {

public static void main(String[] args) {

Dog labrador = new Dog();

labrador.eat();

labrador.bark();

}

}

class Animal {

public void eat() {

System.out.println("I can eat");

}

}

class Dog extends Animal {

@Override

public void eat() {

super.eat();

System.out.println("I eat dog food");

}

public void bark() {

System.out.println("I can bark");

}

}

class Main {

public static void main(String[] args) {

Dog labrador = new Dog();

labrador.eat();

labrador.bark();

}

}

class Animal {

protected String name;

protected void display() {

System.out.println("I am an animal.");

}

}

class Dog extends Animal {

public void getInfo() {

System.out.println("My name is " + name);

}

}

class Main {

public static void main(String[] args) {

Dog labrador = new Dog();

labrador.name = "Rocky";

labrador.display();

labrador.getInfo();

}

}

class Calculation {

int z;

public void addition(int x, int y) {

z = x + y;

System.out.println("The sum of the given numbers:"+z);

}

public void Subtraction(int x, int y) {

z = x - y;

System.out.println("The difference between the given numbers:"+z);

}

}

public class My\_Calculation extends Calculation {

public void multiplication(int x, int y) {

z = x \* y;

System.out.println("The product of the given numbers:"+z);

}

public static void main(String args[]) {

int a = 20, b = 10;

My\_Calculation demo = new My\_Calculation();

demo.addition(a, b);

demo.Subtraction(a, b);

demo.multiplication(a, b);

}

}

EXCEPTION HANDLING

##############################################################################

class EDemo1

{

public static void main(String args[])

{

int a=100/0;

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

}

}

##############################################################################

class EDemo2

{

public static void main(String args[])

{

try

{

int a=100/0;

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

}

catch(ArithmeticException ee)

{

System.out.println(" Can't Divide by zero ");

}

}

}

##############################################################################

class EDemo3

{

public static void main(String args[])

{

try

{

System.out.println(" Connection Opened ");

int a=100/2;

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

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

int b[]={10,20,30,40};

System.out.println(" B : "+b[2]);

System.out.println(" Connection Closed ");

}

catch(ArithmeticException ee)

{

System.out.println(" Can't Divide by zero");

}

catch(ArrayIndexOutOfBoundsException ee)

{

System.out.println(" Array Index Out Of Range");

}

}

}

##############################################################################

class EDemo4

{

public static void main(String args[])

{

try

{

System.out.println(" Connection Opened ");

int a=100/2;

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

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

int b[]={10,20,30,40};

System.out.println(" B : "+b[1]);

}

catch(ArithmeticException ee)

{

System.out.println(" Can't Divide by zero");

}

catch(ArrayIndexOutOfBoundsException ee)

{

System.out.println(" Array Index Out Of Range");

}

finally

{

System.out.println(" Connection Closed ");

System.out.println("Finally Block");

}

}

}

##############################################################################

class EDemo04{

public static void main(String args[]){

try{

System.out.println(" Connection Opened \n\n");

int a=100/2;

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

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

int b[]={10,20,30,40};

System.out.println(" B : "+b[11]);

System.out.println(" \n\n Connection Closed ");

}

catch(Exception ex){

if(ex instanceof ArithmeticException){

System.out.println(" Can't Divide by zero");

}

if(ex instanceof ArrayIndexOutOfBoundsException){

System.out.println(" Array Index Out Of Range");

}

}

}

}

##############################################################################

class AgeException extends Exception //user defined Exception

{

String getException()

{

return "Age Should not > 25";

}

}

class Registration

{

void validation(int x)throws AgeException

{

if(x>25)

{

throw new AgeException();

}

else

{

System.out.println(" Validation Success!");

}

}

}

class EDemo5

{

public static void main(String args[])

{

Registration s1=new Registration();

//s1.validation(21);

try

{

s1.validation(14);

}

catch(AgeException ee)

{

String msg=ee.getException();

System.out.println("------------------> "+msg);

}

}

}

package exceptionhandling;

import java.util.Scanner;

class AgeException extends Exception

{

String getException()

{

return "Age Should not > 25";

}

}

class Registration

{

void validation(int x) throws AgeException

{

if(x>25)

{

throw new AgeException();

}

else

{

System.out.println(" Validation Success!");

}

}

}

##############################################################################

class EDemo6

{

public static void main(String args[])

{

Registration s1 = new Registration();

//s1.validation(10);

try

{

Scanner s=new Scanner(System.in);

System.out.print(" Enter the Value ");

int x = s.nextInt();

s1.validation(x);

}

catch(Exception ee)

{

if(ee instanceof AgeException){

AgeException age=(AgeException)ee;

String msg=age.getException();

System.out.println("------------------> "+msg);

}

}

}

}

##############################################################################

import java.io.\*;

class Sample

{

Sample(int x)throws FileNotFoundException

{

if(x<20)

{

throw new FileNotFoundException();

}

else

{

System.out.println(" Validation Success!");

}

}

}

class EDemo6

{

public static void main(String args[])

{

//new Sample(31);

try

{

new Sample(22);

}

catch(FileNotFoundException ee)

{

System.out.println("Value should not < 20");

}

}

}

##############################################################################

import java.io.\*;

class EDemo7

{

public static void main(String args[])

{

FileReader fr=new FileReader("EDemo19.java");

}

}

##############################################################################

import java.io.\*;

class EDemo8

{

public static void main(String args[])

{

try

{

FileReader fr=new FileReader("EDemo19.java");

System.out.println("Success!");

}

catch(FileNotFoundException ee)

{

System.out.println("File is not available");

}

}

}

##############################################################################

class Sample1

{

void test(int x)throws ArithmeticException

{

if(x<20)

{

ArithmeticException ob=new ArithmeticException();

throw ob;

}

else

{

System.out.println(" Validation Success!");

}

}

}

class EDemo9

{

public static void main(String args[])

{

Sample1 s1=new Sample1();

s1.test(10);

}

}

##############################################################################

class Sample2

{

void test(int x)throws ArithmeticException

{

if(x<20)

{

ArithmeticException ob=new ArithmeticException();

throw ob;

}

else

{

System.out.println(" Validation Success!");

}

}

}

class EDemo10

{

public static void main(String args[])

{

try

{

Sample2 s1=new Sample2();

s1.test(10);

}

catch(ArithmeticException ee)

{

System.out.println("Value should not < 20");

}

}

}

##############################################################################

import java.util.Scanner;

class AssertionExample

{

public static void main( String args[] )

{

Scanner input= new Scanner(System.in);

System.out.print("Enter ur age ");

int value = input.nextInt();

assert value>=18:" Not valid";

System.out.println("value is "+value);

}

}

// javac AssertionExample.java

// java -ea AssertionExample

=================================================================================

JavaBean

-------------

package datas;

public class StudentDetails {

private int rno;

private String name;

private String city;

public int getRno() {

return rno;

}

public void setRno(int rno) {

this.rno = rno;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

}

THREADS

1) 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();

 }

}

2) 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();

 }

}

3) public class MyThread2 implements Runnable

{

public void run()

{

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

}

// main method

public static void main(String argvs[])

{

// creating an object of the class MyThread2

Runnable r1 = new MyThread2();

// creating an object of the class Thread using Thread(Runnable r, String name)

Thread th1 = new Thread(r1, "My new thread");

// the start() method moves the thread to the active state

th1.start();

// getting the thread name by invoking the getName() method

String str = th1.getName();

System.out.println(str);

}

}

JDBC

import java.io.\*;

import java.sql.\*;

class GFG {

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

    {

        String url

            = "jdbc:<mysql://localhost:3306/table_name>"; // table details

        String username = "rootgfg"; // MySQL credentials

        String password = "gfg123";

        String query

            = "select \*from students"; // query to be run

        Class.forName(

            "com.mysql.cj.jdbc.Driver"); // Driver name

        Connection con = DriverManager.getConnection(

            url, username, password);

        System.out.println(

            "Connection Established successfully");

        Statement st = con.createStatement();

        ResultSet rs

            = st.executeQuery(query); // Execute query

        rs.next();

        String name

            = rs.getString("name"); // Retrieve name from db

        System.out.println(name); // Print result on console

        st.close(); // close statement

        con.close(); // close connection

        System.out.println("Connection Closed....");

    }

}