**CODE:**

import java.util.Scanner;

public class Dog

{

String name;

String breed;

int age;

String color;

public Dog(String name, String breed, int age, String color)

{

this.name = name;

this.breed = breed;

this.age = age;

this.color = color;

}

void print(int age)

{

System.out.println("My dog is " + age + " old and its charming.");

}

void print(String breed)

{

System.out.println("My dog is " + breed + " and its lovely.");

}

public String display()

{

return("Hi my name is "+ this.name + " .\n" + "I am a " + this.breed + ". \nI am " + this.color + " in color and just " + this.age + " age old.");

}

public static void main(String[] args)

{

Scanner myObj = new Scanner(System.in);

System.out.println("Enter name, breed, color and age:");

String name1 = myObj.nextLine();

String breed1 = myObj.nextLine();

String color1 = myObj.nextLine();

int age1 = myObj.nextInt();

Dog dog1 = new Dog(name1, breed1, age1, color1);

System.out.println(dog1.display());

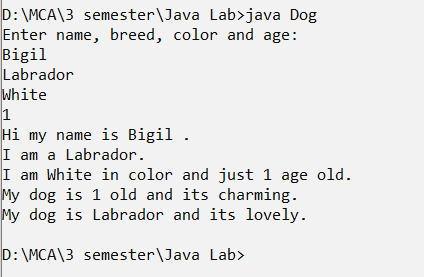
dog1.print(age1);

dog1.print(breed1);

}

}

**OUTPUT:**



**CODE:**

interface A //creating interface

{

void Adisplay();

}

interface B

{

void Bdisplay();

}

class AB implements A,B {

public void Adisplay()

{

System.out.println(" CALCULATIONS");

}

public void Bdisplay()

{

System.out.println("THE GIVEN NUMBERS IS 10 AND 20");

}

}

class Calculation { //base class

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);

}

}

class Division extends Calculation{ //derived class

public void calc(int x, int y){ //method overriding

z = x / y;

System.out.println(" The division of the given number is:"+z);

}

}

class Modulo extends Division{

public void calc(int x , int y){ //method overriding

z = x % y;

System.out.println(" Modulus of the given Number is:"+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;

//object creation

AB obj= new AB();

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

Division demo1 = new Division();

Modulo demo2 = new Modulo();

obj.Adisplay();

obj.Bdisplay();

demo.addition(a, b);

demo.Subtraction(a, b);

demo1.calc(a, b);

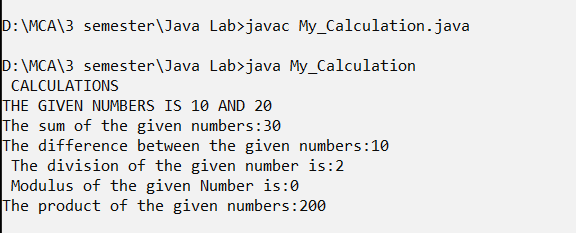
demo2.calc(a, b);

demo.multiplication(a, b);

}

}

**OUTPUT:**

****

**CODE:**

import java.util.Scanner;

public class Main {

private final static int MAX = 100;

private volatile static int maxDivisorCount = 0;

private volatile static int intWithMaxDivisorCount;

synchronized private static void report(int maxCountFromThread,

int intWithMaxFromThread) {

if (maxCountFromThread > maxDivisorCount) {

maxDivisorCount = maxCountFromThread;

intWithMaxDivisorCount = intWithMaxFromThread;

}

}

private static class CountDivisorsThread extends Thread {

int min, max;

public CountDivisorsThread(int min, int max) {

this.min = min;

this.max = max;

}

public void run() {

System.out.println("Thread " + this + " testing range " +

min + " to " + max); // For testing.

int maxDivisors = 0;

int whichInt = 0;

for (int i = min; i < max; i++) {

int divisors = countDivisors(i);

if (divisors > maxDivisors) {

maxDivisors = divisors;

whichInt = i;

}

}

report(max''Divisors,whichInt);

}

}

private static void countDivisorsWithThreads(int numberOfThreads) {

System.out.println("\nCounting divisors using " +

numberOfThreads + " threads...");

long startTime = System.currentTimeMillis();

CountDivisorsThread[] worker = new CountDivisorsThread[numberOfThreads];

int integersPerThread = MAX/numberOfThreads;

int start = 1; // Starting point of the range of ints for first thread.

int end = start + integersPerThread - 1; // End point of the range of ints.

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

if (i == numberOfThreads - 1) {

end = MAX; // Make sure that the last thread's range goes all

// the way up to MAX. Because of rounding, this

// is not automatic.

}

worker[i] = new CountDivisorsThread( start, end );

start = end+1; // Determine the range of ints for the NEXT thread.

end = start + integersPerThread - 1;

}

maxDivisorCount = 0;

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

worker[i].start();

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

while (worker[i].isAlive()) {

try {

worker[i].join();

}

catch (InterruptedException e) {

}

}

}

long elapsedTime = System.currentTimeMillis() - startTime;

System.out.println("\nThe largest number of divisors " +

"for numbers between 1 and " + MAX + " is " + maxDivisorCount);

System.out.println("An integer with that many divisors is " +

intWithMaxDivisorCount);

System.out.println("Total elapsed time: " +

(elapsedTime/1000.0) + " seconds.\n");

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int numberOfThreads = 0;

while (numberOfThreads < 1 || numberOfThreads > 10) {

System.out.print("How many threads do you want to use (1 to 10) ? ");

numberOfThreads = in.nextInt();

if (numberOfThreads < 1 || numberOfThreads > 10)

System.out.println("Please enter a number from 1 to 10 !");

}

countDivisorsWithThreads(numberOfThreads);

}

public static int countDivisors(int N) {

int count = 0;

for (int i = 1; i <= N ; i++) {

if ( N % i == 0 )

count ++;

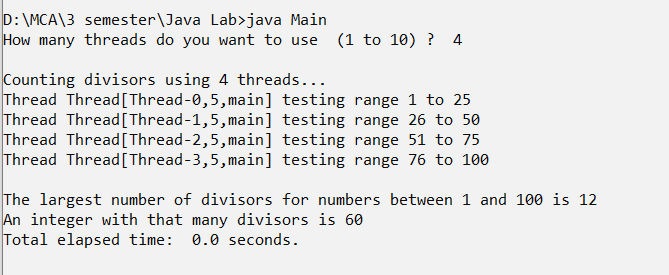
}

return count;

}

}

**OUTPUT**



**CODE:**

import java.util.\*;

import java.lang.Exception;

class InvalidAgeException extends Exception

{

public InvalidAgeException (String str)

{

super(str);

}

}

public class CustomException

{

static void validate (int age) throws InvalidAgeException{

if(age < 18){

throw new InvalidAgeException("Not Eligible to vote.");

}

else {

System.out.println("Eligible to vote.");

}

}

public static void main(String args[])

{

String name;

int age;

Scanner scan= new Scanner(System.in);

try

{

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

name=scan.nextLine();

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

age=scan.nextInt();

validate(age);

}

catch (InvalidAgeException ex)

{

System.out.println("Caught the exception");

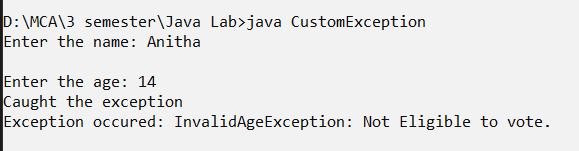
System.out.println("Exception occured: " + ex);

}

}

}

**OUTPUT:**



**CODE:**

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class ReadFile {

public static void main(String[] args) {

try {

File myObj = new File("sample.txt");

Scanner myReader = new Scanner(myObj);

while (myReader.hasNextLine()) {

String data = myReader.nextLine();

System.out.println(data);

}

myReader.close();

} catch (FileNotFoundException e) {

System.out.println("An error occurred.");

e.printStackTrace();

}

}

}

**OUTPUT:**

