Write a program in Java to create a thread by extending the ‘Thread’ class and by implementing the “Runnable” interface ?

Thread with extends class :-

class A extends Thread{

public void run(){

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

}

}

class B extends Thread{

public void run() {

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

}

}

class demo{

public static void main(String args[]){

A t1 = new A();

B t2 = new B();

t1.start();

t2.start();

}

}

Thread with interface :-

class A implements Runnable{

public void run(){

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

}

}

class B implements Runnable{

public void run() {

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

}

}

class demo{

public static void main(String[] args){

A obj1 = new A();

B obj2 = new B();

Thread t1 = new Thread(obj1);

Thread t2 = new Thread(obj2);

t1.start();

t2.start();

}

}

Write a program in Java to demonstrate sleep() and wait() ?

class demo{

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

System.out.println("Hello Java");

Thread.sleep(7000);

System.out.println("Nice Java");

}

}

Write a program in Java to demonstrate synchronization ?

import java.io.\*;

import java.util.\*;

class Sender

{

public void send(String msg)

{

System.out.println("Sending\t" + msg );

try

{

Thread.sleep(1000);

}

catch (Exception e)

{

System.out.println("Thread interrupted.");

}

System.out.println("\n" + msg + "Sent");

}

}

class ThreadedSend extends Thread

{

private String msg;

Sender sender;

ThreadedSend(String m, Sender obj)

{

msg = m;

sender = obj;

}

public void run()

{

synchronized(sender)

{

sender.send(msg);

}

}

}

class demo

{

public static void main(String args[])

{

Sender send = new Sender();

ThreadedSend S1 =

new ThreadedSend( " Hi " , send );

ThreadedSend S2 =

new ThreadedSend( " Bye " , send );

S1.start();

S2.start();

try

{

S1.join();

S2.join();

}

catch(Exception e)

{

System.out.println("Interrupted");

}

}

}

Write a program in Java to demonstrate try and catch ?

class demo{

public static void main(String args[]) {

try {

int n = 2/0;

}

catch(ArithmeticException e){

System.out.println("It gives an error because you divide by zero");

System.out.println("Error "+ e);

}

}

}

Writing a program in Java to throws, throw, finally, and custom exceptions in Java. ?

public class demo {

public static void inside\_test1\_error() throws Exception{

System.out.println("Error Starting");

throw new Exception("%+5");

}

public static void inside\_test2\_error() throws Exception {

try {

inside\_test1\_error();

}

catch(Exception e) {

System.out.println("I catch a eror of test1\_insdie\_error");

System.out.println("Error "+ e);

}

}

public static void main(String args[]) {

try {

inside\_test2\_error();

}

catch(Exception e){

System.out.println("It gives an error in main");

System.out.println("Error "+ e);

}

finally {

System.out.println("It runs always");

}

}

}

Write a program in Java to demonstrate exception handling ?

class demo{

public static void main(String args[]) {

try {

int n = 2/0;

}

catch(ArithmeticException e){

System.out.println("It gives an error because you divide by zero");

System.out.println("Error "+ e);

}

}

}

Writing a program in Java to create, read, update, and delete operations on the files in Java.?

import java.io.File;

import java.io.IOException;

import java.io.RandomAccessFile;

import java.lang.NumberFormatException;

class AddFriend {

public static void main(String data[])

{

try {

String newName = data[0];

long newNumber = Long.parseLong(data[1]);

String nameNumberString;

String name;

long number;

int index;

File file = new File("friendsContact.txt");

if (!file.exists()) {

file.createNewFile();

}

RandomAccessFile raf

= new RandomAccessFile(file, "rw");

boolean found = false;

while (raf.getFilePointer() < raf.length()) {

nameNumberString = raf.readLine();

String[] lineSplit

= nameNumberString.split("!");

name = lineSplit[0];

number = Long.parseLong(lineSplit[1]);

if (name == newName

|| number == newNumber) {

found = true;

break;

}

}

if (found == false) {

nameNumberString

= newName + "!"

+ String.valueOf(newNumber);

raf.writeBytes(nameNumberString);

raf.writeBytes(System.lineSeparator());

System.out.println(" Friend added. ");

raf.close();

}

else {

raf.close();

System.out.println(" Input name"

+ " does not exists. ");

}

}

catch (IOException ioe) {

System.out.println(ioe);

}

catch (NumberFormatException nef) {

System.out.println(nef);

}

}

}

Write a program in Java to demonstrate the uses of classes, objects, and the object-oriented pillars in Java ?

Fours pillars of Java :-

Abstraction

Polymorphism

Inheritance

Encapsulation

All other exmples are include in this examples.

Example of Inheritance :-

class Employee{

public void name() {

System.out.println("RAMU");

}

}

class Employee1{

public void name1() {

System.out.println("Krrish");

}

}

class demo{

public static void main(String args[]) {

Employee1 obj = new Employee1();

obj.name1();

Employee obj1 = new Employee();

obj1.name();

}

}

Example of Abstraction:-

interface Employee{

public void name() ;

}

interface Employee1{

public void name1();

}

class salary implements Employee,Employee1{

public void name() {

System.out.println("RAMU");

}

public void name1() {

System.out.println("Krrish");

}

}

class demo{

public static void main(String args[]) {

salary obj = new salary();

obj.name();

obj.name1();

}

}

Write a program in Java to resolve the diamond problem using OOPs’ concepts

interface Employee{

public void name() ;

}

interface Employee1{

public void name1();

}

class salary implements Employee,Employee1{

public void name() {

System.out.println("RAMU");

}

public void name1() {

System.out.println("Krrish");

}

}

class demo{

public static void main(String args[]) {

salary obj = new salary();

obj.name();

obj.name1();

}

}