1.Write the programme to open a text file named input 2, and copy its contents to an output text file output 2.

# **Code:-**

package MyPackage; //importing classes

Import java.io.BufferedReader;

Import java.io.BufferedWriter;

Import java.io.FileReader;

Import java.io.FileWriter;

Import java.io.IOException;

Import java.io.\*;

Public class SimpleBufferedFileCopy

{

Public static void main(String[] args)

{

//input file path String inputFile = “input2.txt”;

//output file path

String outputFile = “output2.txt”;

//reading input file

FileReader fr = new FileReader(inputFile);

//writing in output file

FileWriter fw = new FileWriter(outputFile);

Try (BufferedReader br = new BufferedReader(fr);

BufferedWriter bw = new BufferedWriter(fw)) {

//creat line

String line;

While ((line = br.readLine()) != null) {

//write line of input file

Bw.write(line);

//create new line

Bw.newLine();

}

//message after successful copy

System.out.println(“File copied successfully!”);

} catch (IOException e) {

//message when some error occurs

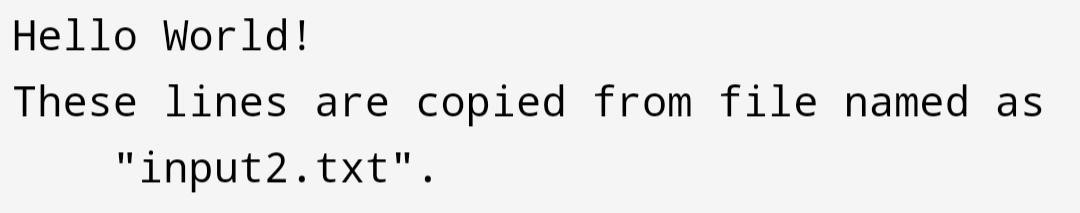
System.out.println(“An error occurred: “ + e.getMessage());

}

}

}

# Output:-



Write the programme to show multithreading for the string “multi threads”. Show the resulting output.

## Code:-

package MyPackage;

class MultiThread implements Runnable {

String word;

//constructor

Public MultiThread(String word) {

This.word = word;

}

//method

Public void run() {

// Print the word

System.out.print(word + “ “);

}

}

Public class MultiThreadExample

{

Public static void main(String[] args)

{

String str = “multi threads”;

String[] words = str.split(“ “); // Split the string into words

Thread[] threads = new Thread[words.length];

// Create and start a thread for each word in the string

For (int I = 0; I < words.length; i++) { Threads[i] = new Thread(new MultiThread(words[i]));

Threads[i].start();

}

// Wait for all threads to finish

For (int I = 0; I < threads.length; i++) {

Try {

Threads[i].join();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

# Output:-



Implement a Java program that creates a thread using the Runnable interface. The thread should print numbers from 1 to 10 with a delay of 1 second between each number.

## Code:-

package MyPackage;

//creating classs

Public class PrintNumber implements Runnable {

//method

Public void run() {

//loop for printing number from 1 to 5

For (int I = 1; I <= 10; i++) {

System.out.println(i);

Try {

//print number with 1s difference

Thread.sleep(1000);

} catch (InterruptedException e) { //message when some error occurs

System.err.println(“Thread was interrupted: “ + e.getMessage());

}

}

}

Public static void main(String[] args)

{

//creating object of class

PrintNumber pn = new PrintNumber();

//creating thread

Thread thread = new Thread(pn);

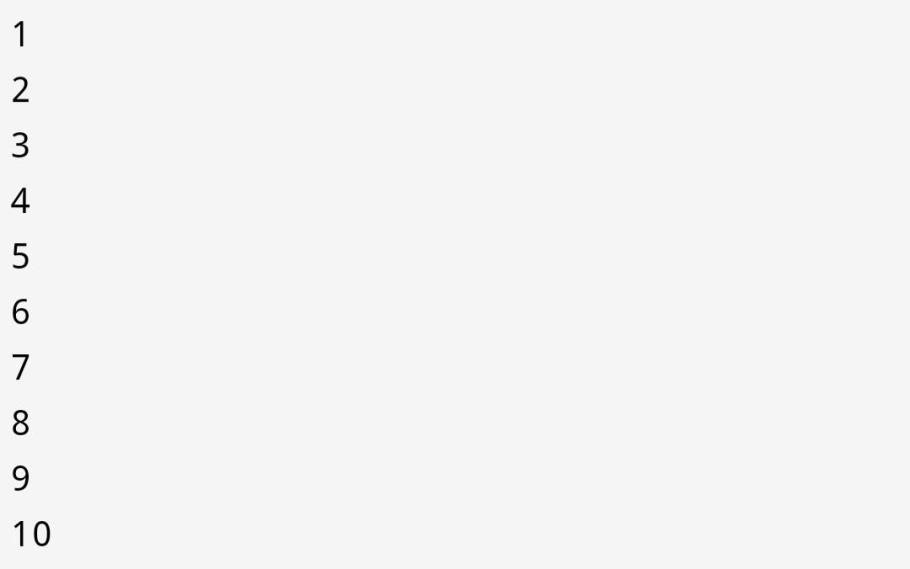
//starting thread

Thread.start();

}

}

# Output:-



Write a Java program that creates and starts three threads. Each thread should print its name and count from 1 to 5 with a delay of 500 milliseconds between each count.

Code:-

package MyPackage;

//creating thread1 class which prints numbers from 1 to 5 with time difference of 500 milliseconds

Class Thread1 extends Thread {

Public void run() {

For (int I = 1; I <= 5; i++) {

System.out.println(“First thread : “ + i);

Try {

Thread.sleep(500);

} catch (InterruptedException e) {

System.err.println(“Thread was interrupted: “ + e.getMessage()); }

}

}

}

//creating thread2 class which prints numbers from 1 to 5 with time difference of 500 milliseconds

Class Thread2 extends Thread {

Public void run() {

For (int I = 1; I <= 5; i++) {

System.out.println(“Second thread : “ + i);

Try {

Thread.sleep(500);

} catch (InterruptedException e) {

System.err.println(“Thread was interrupted: “ + e.getMessage());

}

}

}

}

//creating thread3 class which prints numbers from 1 to 5 with time difference of 500 milliseconds

Class Thread3 extends Thread { Public void run() {

For (int I = 1; I <= 5; i++) {

System.out.println(“Third thread : “ + i);

Try {

Thread.sleep(500);

} catch (InterruptedException e) {

System.err.println(“Thread was interrupted: “ + e.getMessage());

}

}

}

}

Public class ThreadDemo

{

Public static void main(String[] args)

{

//creating object of threads with reference to Thread class;

Thread thread1 = new Thread1();

Thread thread2 = new Thread2();

Thread thread3 = new Thread3();

//starting all threads

Thread1.start();

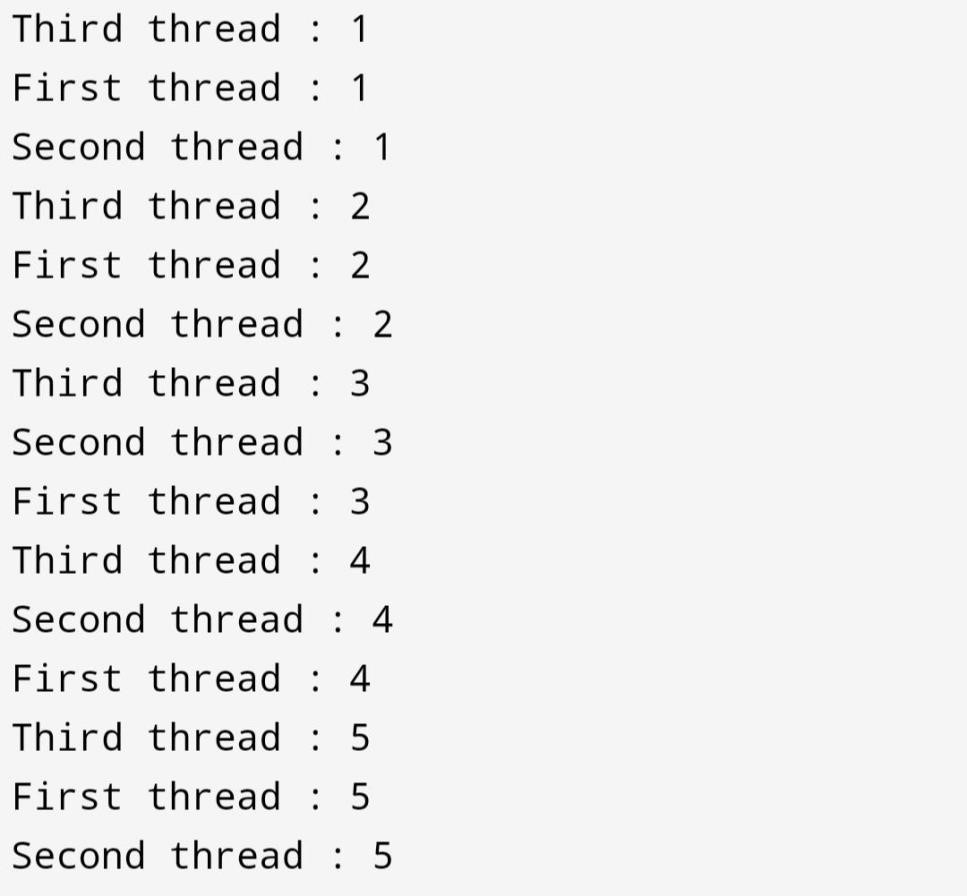
Thread2.start();

Thread3.start();

}

}

# Output:-



5.Create a Java program that demonstrates thread priorities. Create three threads with different priorities and observe the order in which they execute.

Code:-

package MyPackage;

//creating mythread class extends thread class

Class MyThread extends Thread

{

//constructor takes thread name and its priority

Public MyThread(String name, int priority) { Super(name); setPriority(priority);

}

//method

Public void run() {

Try {

System.out.println(getName());

Thread.sleep(500);

} catch (InterruptedException e) {

System.err.println€;

}

}

}

Public class ThreadPriorityDemo

{

Public static void main(String[] args)

{

//creating maximum priority thread

MyThread thread1 = new MyThread(“Maximum Priority Thread.”,

Thread.MAX\_PRIORITY);

//creating medium priority thread

MyThread thread2 = new MyThread(“Medium Priority Thread.”,

Thread.NORM\_PRIORITY);

//creating minimum priority thread

MyThread thread3 = new MyThread(“Minimum Priority Thread.”,

Thread.MIN\_PRIORITY);

//starting threads

Thread1.start();

Thread2.start();

Thread3.start();

}

}

# Output:-

