Unit4: java multithreading

Q1:- 1.     Create a class **MyThread** derived from **Thread** class and override the run method.

 Create a class **ThreadDemo** having main method.  Create 2 objects of MyThread class and observe the behavior of threads

class ThreadDemo extends Thread

{

public void run()

{

for(int i=0;i<10;i++)

{

System.out.println("child thread");

}

}

}

class Test

{

public static void main(String[] args)

{

ThreadDemo t=new ThreadDemo();

Thread t1=new Thread(t);

t1.start();

for(int i=0;i<10;i++)

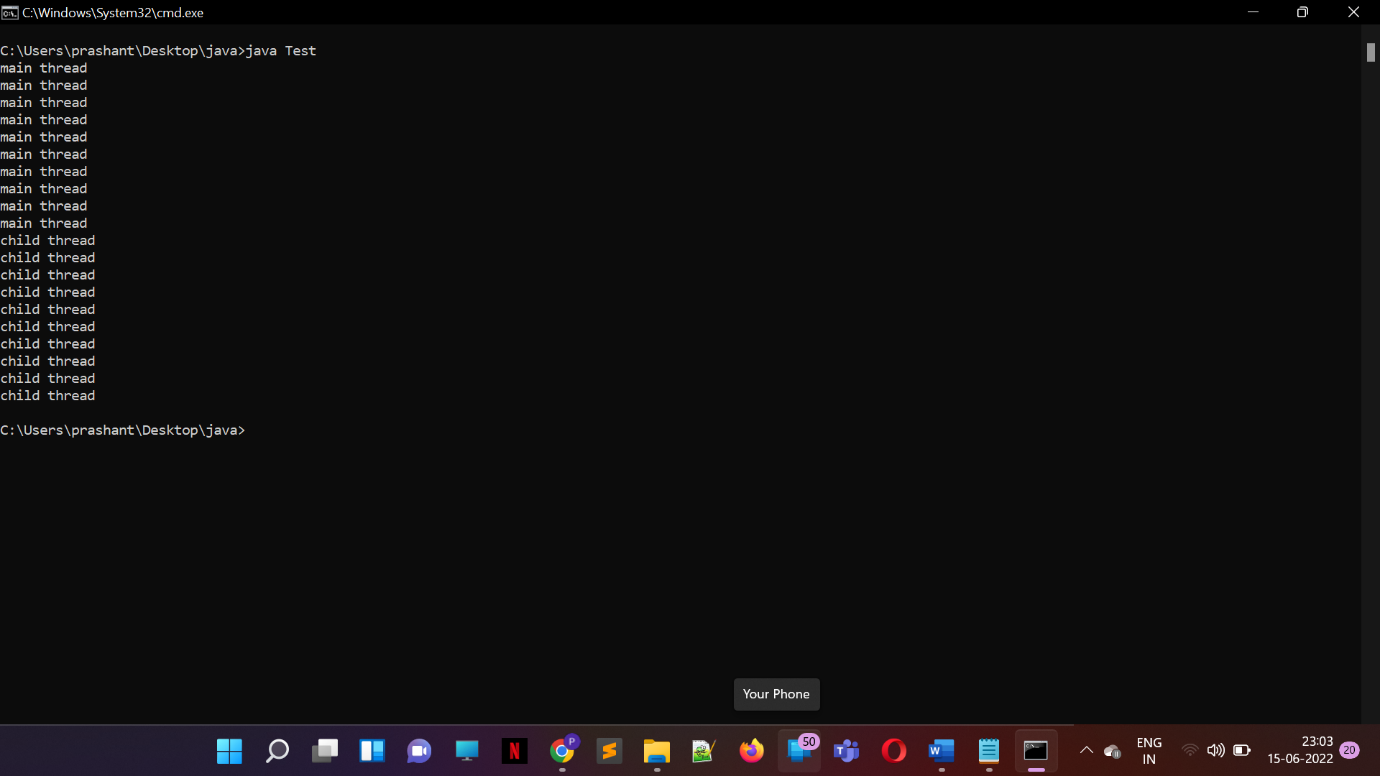
{

System.out.println("main thread");

}

}

}



Q2:- Modify the above to create MyThread class by implementing **Runnable** interface and observe the behavior of threads.

class MyRunnable implements Runnable

{

public void run()

{

for(int i=0;i<10;i++)

{

System.out.println("child thread");

}

}

}

class Test

{

public static void main(String[] args)

{

MyRunnable r=new MyRunnable();

Thread t=new Thread(r);

t.start();

for(int i=0;i<10;i++)

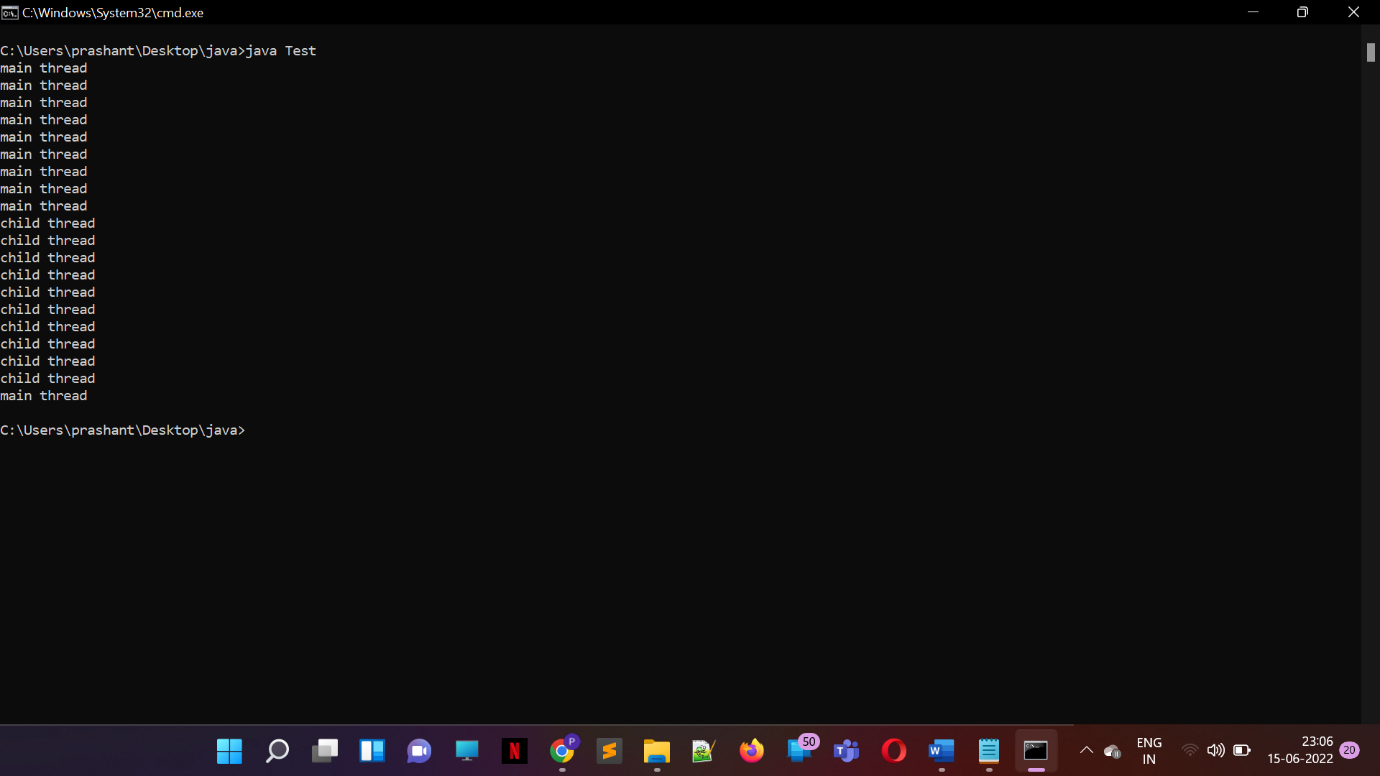
{

System.out.println("main thread");

}

}

}



Q3:- Assign different priorities to the 2 threads and observe the behaviour

// Java Program to Illustrate Priorities in Multithreading

// via help of getPriority() and setPriority() method

// Importing required classes

import java.lang.\*;

// Main class

class ThreadDemo extends Thread {

// Method 1

// run() method for the thread that is called

// as soon as start() is invoked for thread in main()

public void run()

{

// Print statement

System.out.println("Inside run method");

}

// Main driver method

public static void main(String[] args)

{

// Creating random threads

// with the help of above class

ThreadDemo t1 = new ThreadDemo();

ThreadDemo t2 = new ThreadDemo();

ThreadDemo t3 = new ThreadDemo();

// Thread 1

// Display the priority of above thread

// using getPriority() method

System.out.println("t1 thread priority : "

+ t1.getPriority());

// Thread 1

// Display the priority of above thread

System.out.println("t2 thread priority : "

+ t2.getPriority());

// Thread 3

System.out.println("t3 thread priority : "

+ t3.getPriority());

// Setting priorities of above threads by

// passing integer arguments

t1.setPriority(2);

t2.setPriority(5);

t3.setPriority(8);

// t3.setPriority(21); will throw

// IllegalArgumentException

// 2

System.out.println("t1 thread priority : "

+ t1.getPriority());

// 5

System.out.println("t2 thread priority : "

+ t2.getPriority());

// 8

System.out.println("t3 thread priority : "

+ t3.getPriority());

// Main thread

// Displays the name of

// currently executing Thread

System.out.println(

"Currently Executing Thread : "

+ Thread.currentThread().getName());

System.out.println(

"Main thread priority : "

+ Thread.currentThread().getPriority());

// Main thread priority is set to 10

Thread.currentThread().setPriority(10);

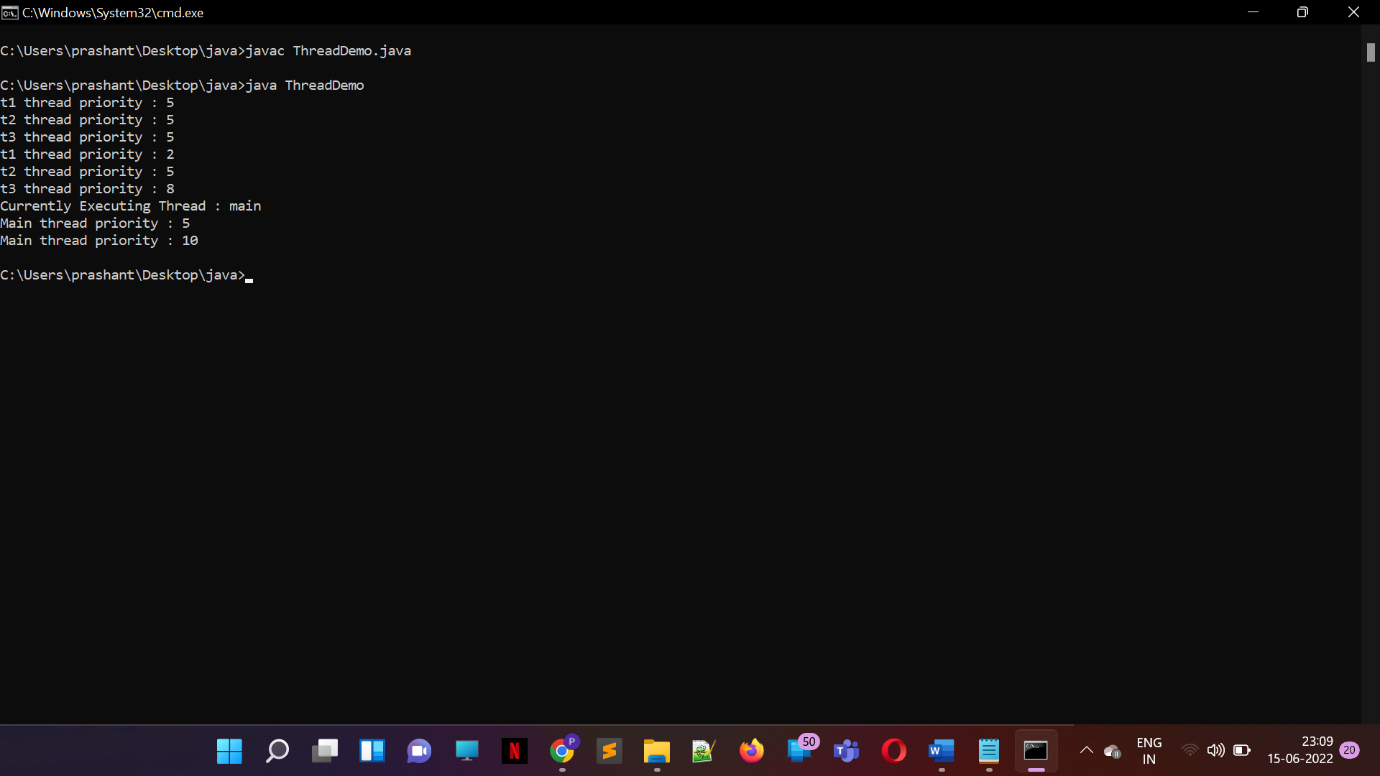
System.out.println(

"Main thread priority : "

+ Thread.currentThread().getPriority());

}

}



Q4:-   Implement three classes: **Storage, Counter** and **Printer**

public class Counter implements Runnable{

public static void main(String[] args) {

Storage store = new Storage();

Counter c1 = new Counter(store);

Printer p1 = new Printer(store);

}

Storage st;

public Counter(Storage store){

st = store;

new Thread(this, "Counter").start();

}

@Override

public void run() {

for(int i = 0 ; i < 10; i++){

st.setValue(i);

}

}

}

class Printer implements Runnable{

Storage st;

public Printer(Storage st){

this.st = st;

new Thread(this, "Printer").start();

}

@Override

public void run() {

for(int i = 0; i < 10; i++)

System.out.println(Thread.currentThread().getName() + " " + st.getValue());

}

}

class Storage{

int i;

public synchronized void setValue(int i){

this.i = i;

}

public synchronized int getValue(){

return this.i;

}

}