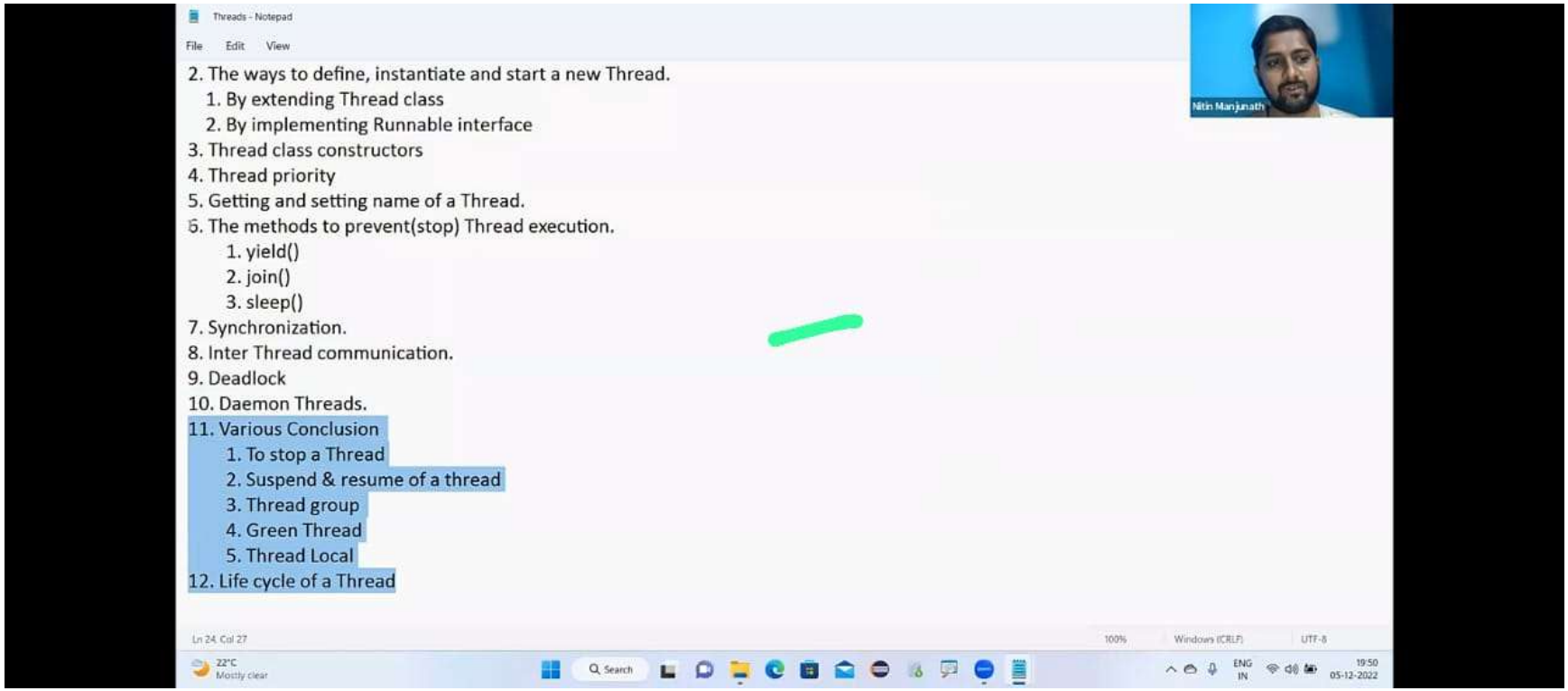
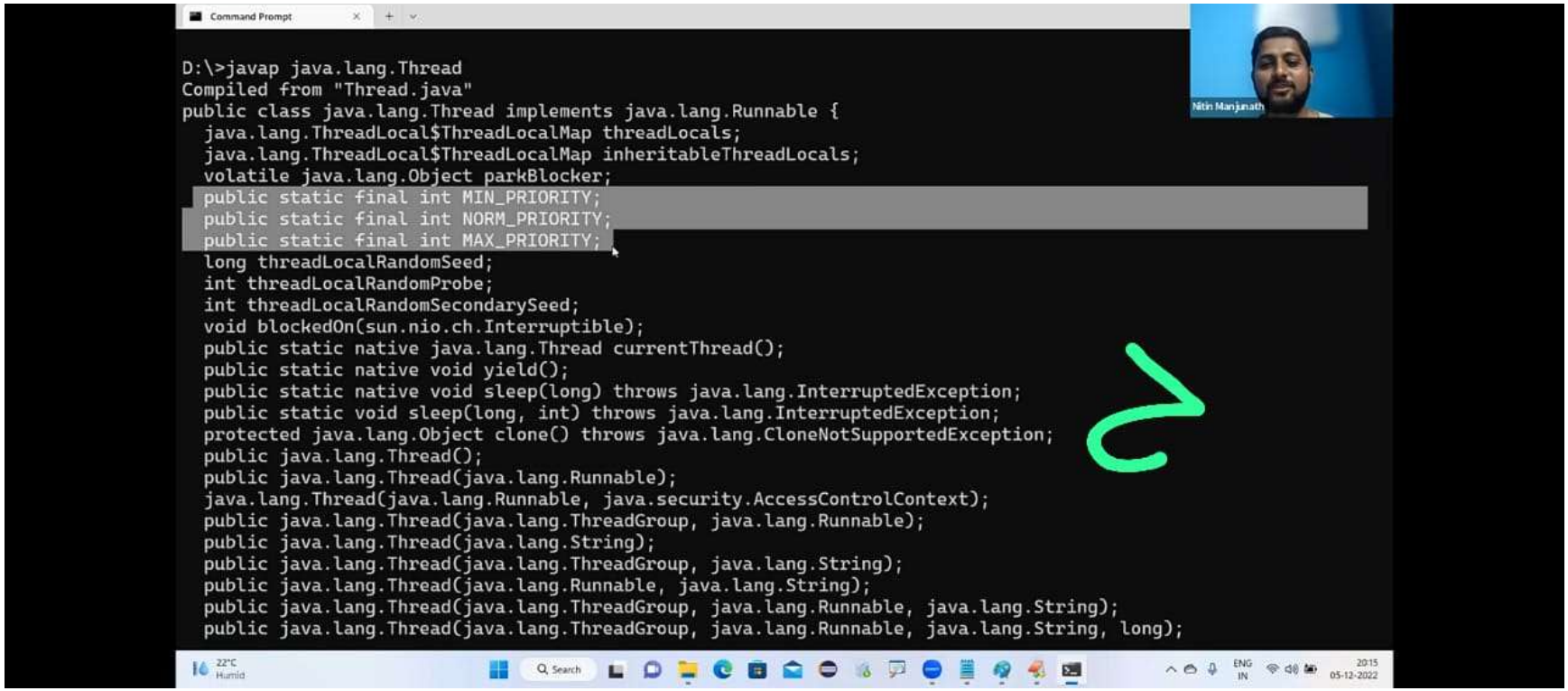
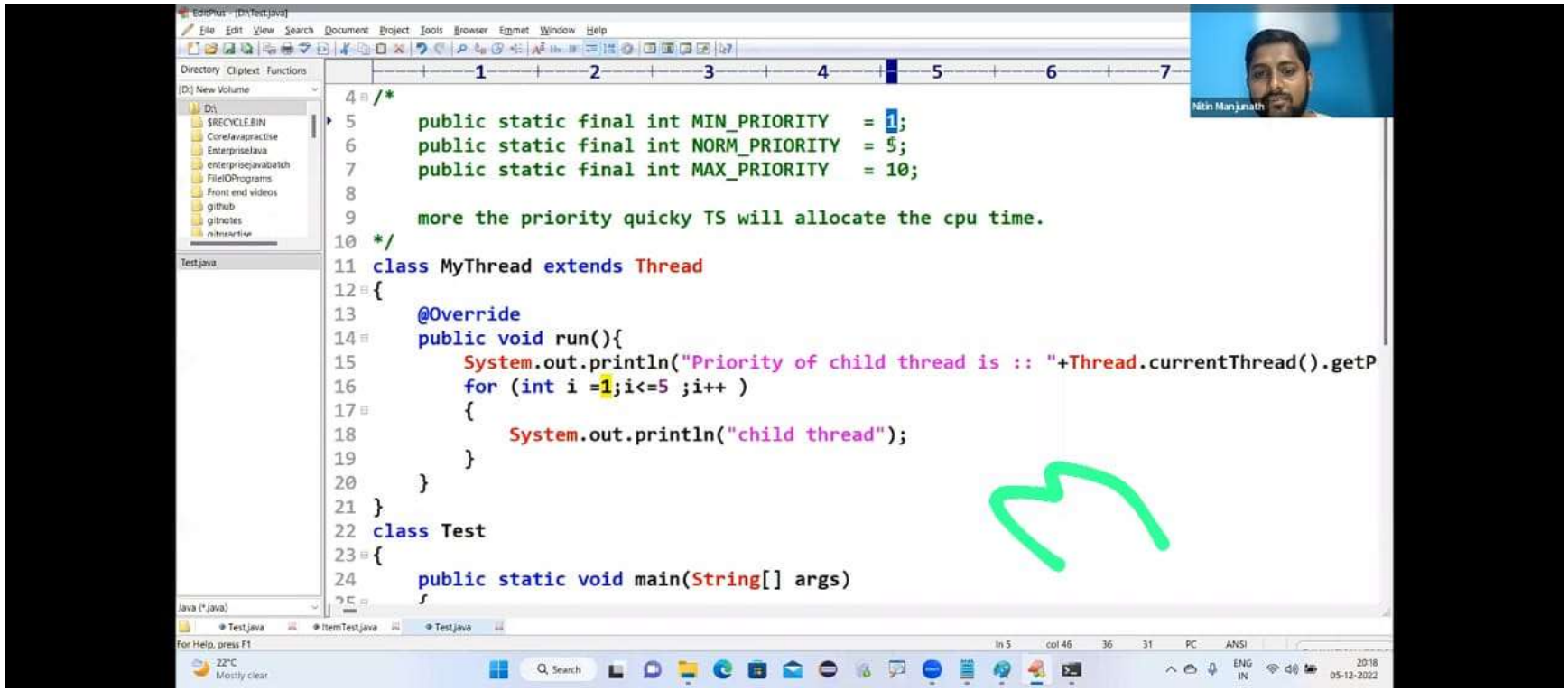
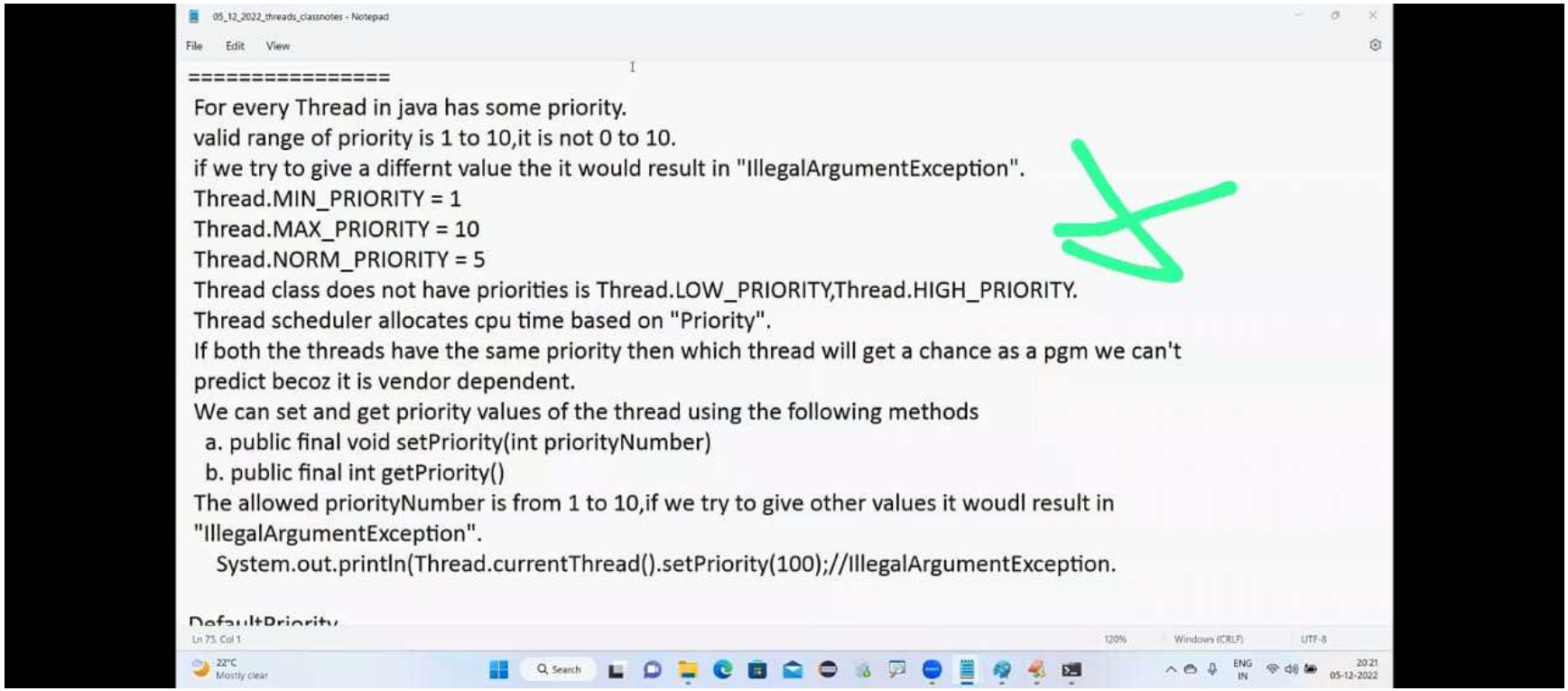
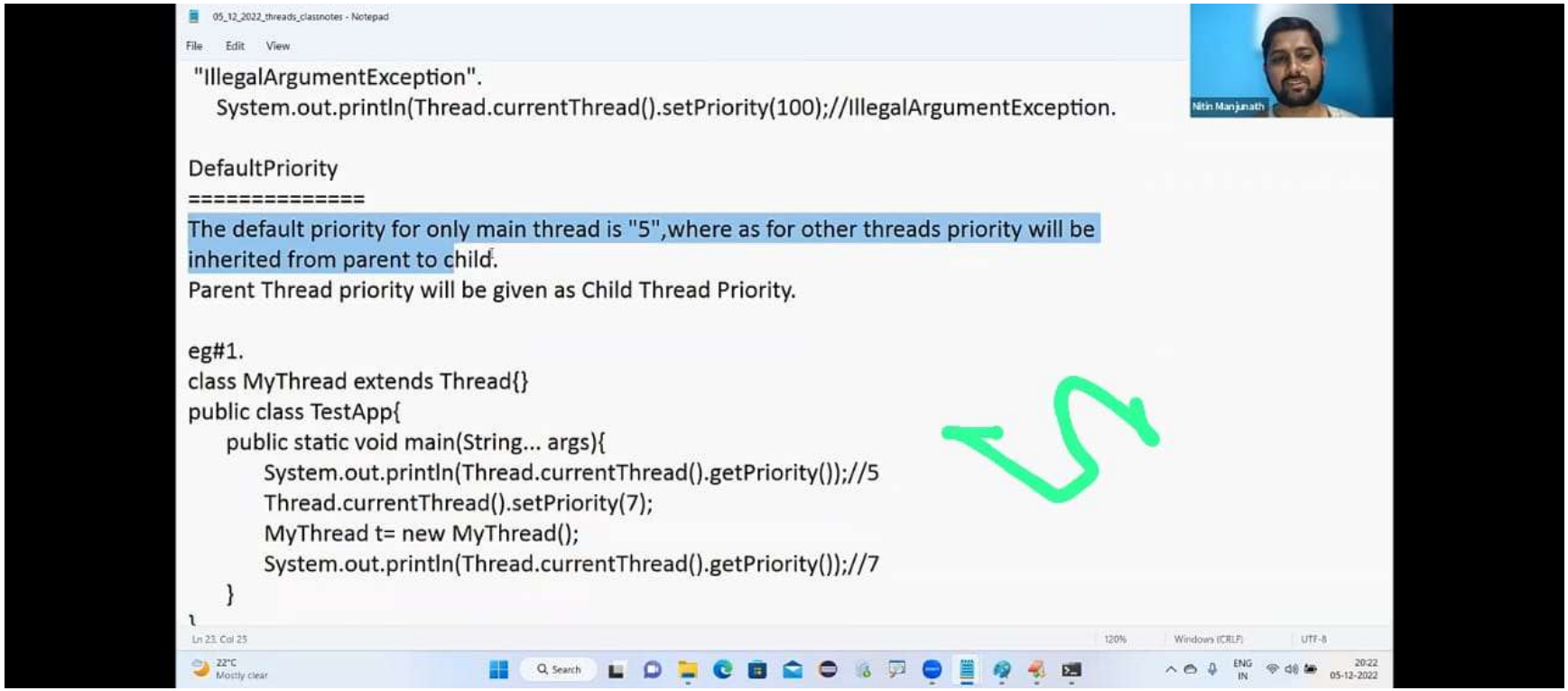
Java Multithreading Part3

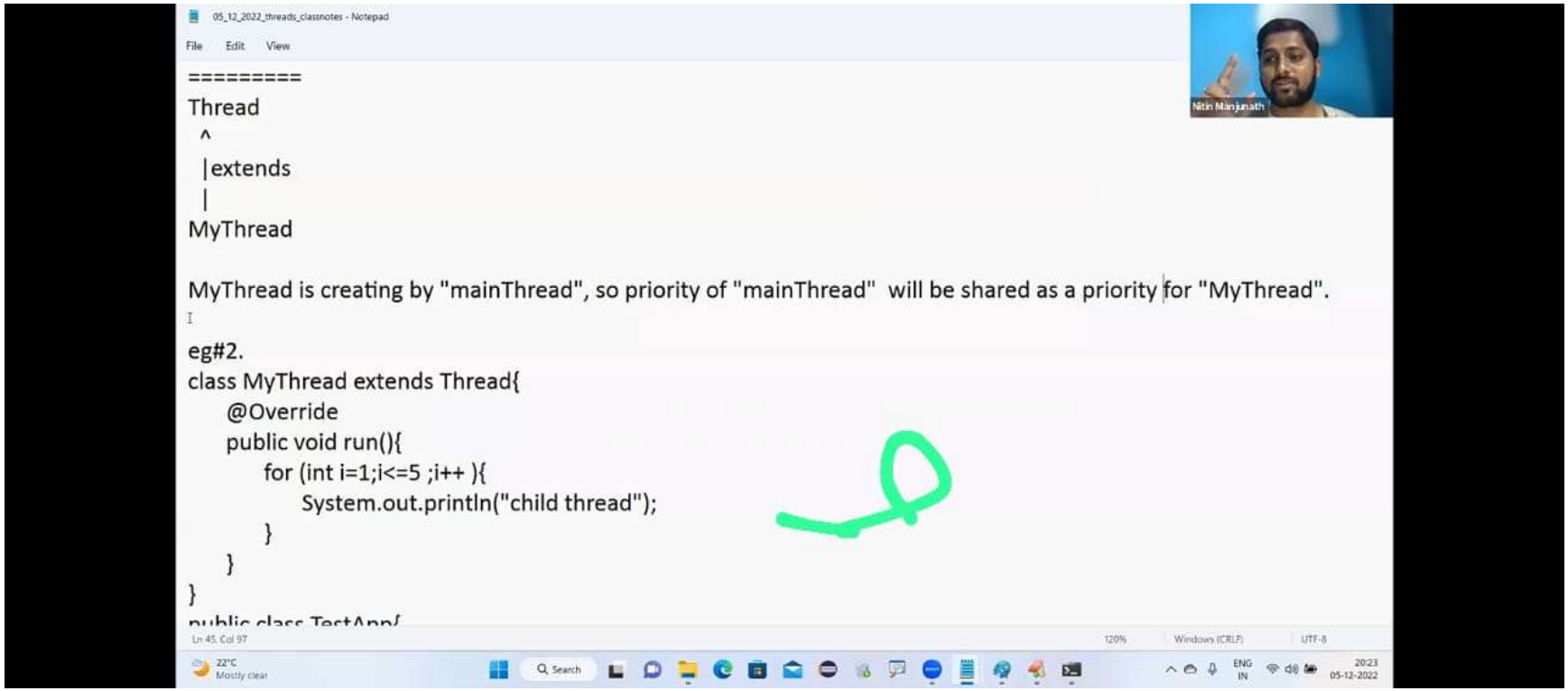


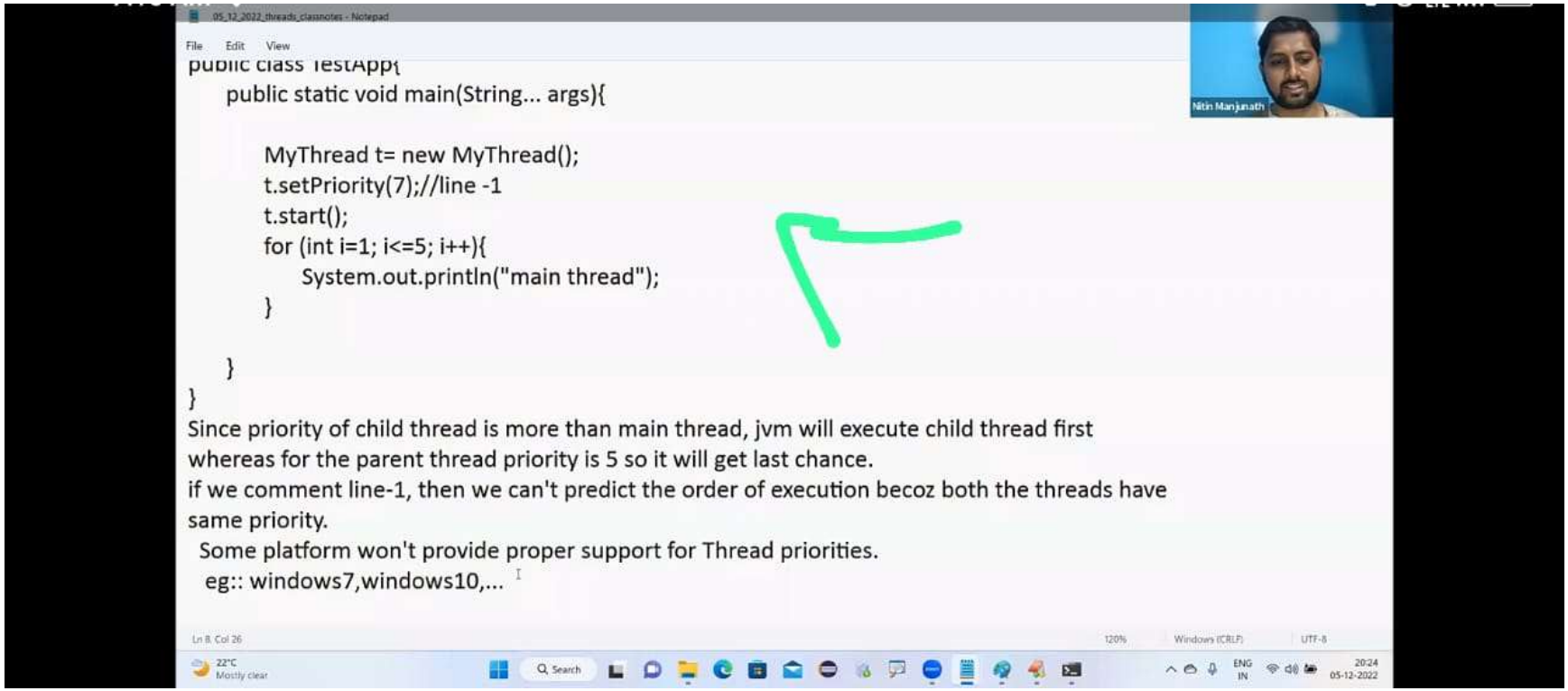

Command Prompt
D:\>javap java.lang.Thread
Compiled from "Thread.java"
public class java.lang.Thread implements java.lang.Runnable {
java.lang.ThreadLocal$ThreadLocalMap threadLocals;
java.lang.ThreadLocal$ThreadLocalMap inheritableThreadLocals;
volatile java.lang.Object parkBlocker;
public static final int MIN_PRIORITY;
public static final int NORM_PRIORITY;
public static final int MAX_PRIORITY;
long threadLocalRandomSeed;
int threadLocalRandomProbe;
int threadLocalRandomSecondarySeed;
void blockedOn(sun.nio.ch.Interruptible);
public static native java.lang.Thread currentThread();
public static native void yield();
public static native void sleep(long) throws java.lang.InterruptedException;
public static void sleep(long, int) throws java.lang.InterruptedException;
protected java.lang.Object clone() throws java.lang.CloneNotSupportedException;
public java.lang.Thread();
public java.lang.Thread(java.lang.Runnable);
java.lang.Thread(java.lang.Runnable, java.security.AccessControlContext);
public java.lang.Thread(java.lang.ThreadGroup, java.lang.Runnable);
public java.lang.Thread(java.lang.String);
public java.lang.Thread(java.lang.ThreadGroup, java.lang.String);
public java.lang.Thread(java.lang.Runnable, java.lang.String);
public java.lang.Thread(java.lang.ThreadGroup, java.lang.Runnable, java.lang.String);
public java.lang.Thread(java.lang.ThreadGroup, java.lang.Runnable, java.lang.String, long);
Nitin Manjunath
22°C
Humid
Search
ENG
IN
20:15
05-12-2022

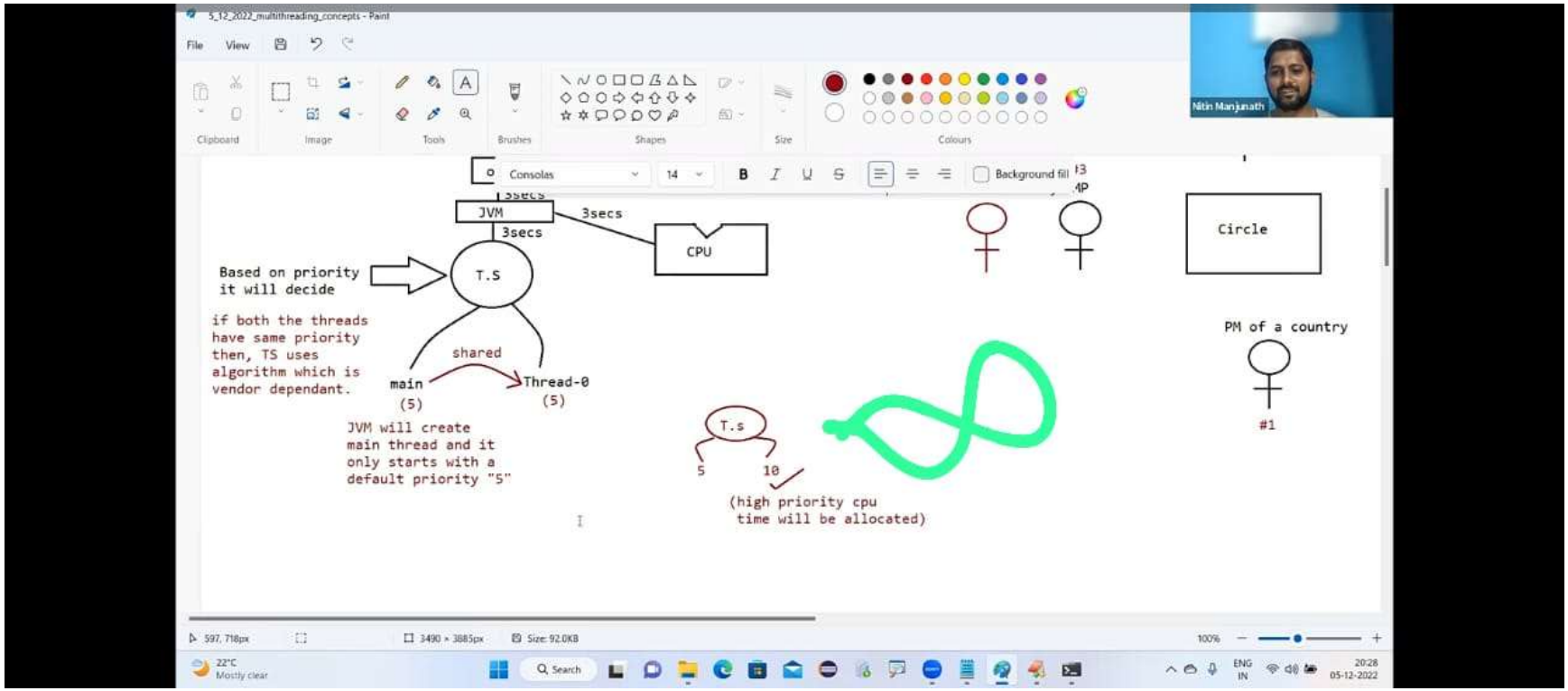

Nitin Manjunath
/*
public static final int MIN_PRIORITY = 1;
public static final int NORM_PRIORITY = 5;
public static final int MAX_PRIORITY = 10;
more the priority quicky TS will allocate the cpu time.
*/
class MyThread extends Thread
{
@Override
public void run(){
System.out.println("Priority of child thread is :: "+Thread.currentThread().getP
for (int i =1;i<=5 ;i++)
{
System.out.println("child thread");
}
}
}
class Test
{
public static void main(String[] args)

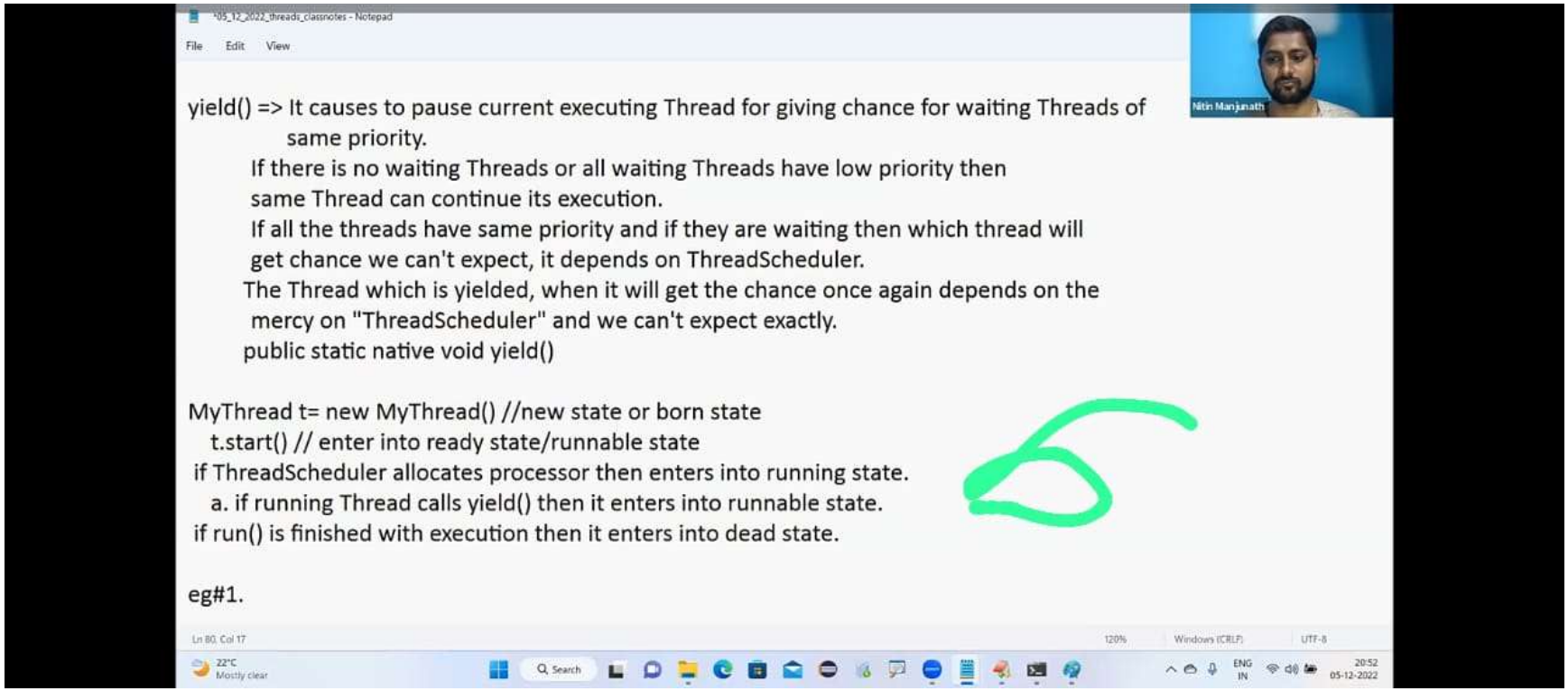

05_12_2022_threads_classnotes - Notepad
File Edit View
==========
For every Thread in java has some priority.
valid range of priority is 1 to 10,it is not 0 to 10.
if we try to give a differnt value the it would result in "IllegalArgumentException".
Thread.MIN_PRIORITY = 1
Thread.MAX_PRIORITY = 10
Thread.NORM_PRIORITY = 5
Thread class does not have priorities is Thread.LOW_PRIORITY,Thread.HIGH_PRIORITY.
Thread scheduler allocates cpu time based on "Priority".
If both the threads have the same priority then which thread will get a chance as a pgm we can't predict becoz it is vendor dependent.
We can set and get priority values of the thread using the following methods
a. public final void setPriority(int priorityNumber)
b. public final int getPriority()
The allowed priorityNumber is from 1 to 10,if we try to give other values it woudl result in "IllegalArgumentException".
System.out.println(Thread.currentThread().setPriority(100);//IllegalArgumentException.
DefaultPriority
Ln 75, Col 1
120%
Windows (CRLF)
UTF-8
22°C
Mostly clear
Search
ENG
IN
20:21
05-12-2022


05_12_2022_threads_classnotes - Notepad
File Edit View
"IllegalArgumentException".
System.out.println(Thread.currentThread().setPriority(100);//IllegalArgumentException.
DefaultPriority
==========
The default priority for only main thread is "5",where as for other threads priority will be inherited from parent to child.
Parent Thread priority will be given as Child Thread Priority.
eg#1.
class MyThread extends Thread{}
public class TestApp{
public static void main(String... args){
System.out.println(Thread.currentThread().getPriority());//5
Thread.currentThread().setPriority(7);
MyThread t= new MyThread();
System.out.println(Thread.currentThread().getPriority());//7
}
}
Nitin Manjunath
Ln 23, Col 25
120%
Windows (CRLF)
UTF-8
22°C
Mostly clear
Search
ENG
IN
20:22
05-12-2022


05_12_2022_threads_classnotes - Notepad
File Edit View
Nitin Manjunath
=========
Thread
^
|extends
|
MyThread
MyThread is creating by "mainThread", so priority of "mainThread" will be shared as a priority for "MyThread".
eg#2.
class MyThread extends Thread{
@Override
public void run(){
for (int i=1;i<=5 ;i++){
System.out.println("child thread");
}
}
}
Ln 45, Col 97
120%
Windows (CRLF)
UTF-8
22°C
Mostly clear
Search
ENG
IN
20:23
05-12-2022


05_12_2022_threads_classnotes - Notepad
File Edit View
public class TestApp{
public static void main(String... args){
MyThread t= new MyThread();
t.setPriority(7);//line -1
t.start();
for (int i=1; i<=5; i++){
System.out.println("main thread");
}
}
}
Since priority of child thread is more than main thread, jvm will execute child thread first
whereas for the parent thread priority is 5 so it will get last chance.
if we comment line-1, then we can't predict the order of execution becoz both the threads have
same priority.
Some platform won't provide proper support for Thread priorities.
eg:: windows7,windows10,...
Nitin Manjunath
Ln 8, Col 26
120%
Windows (CRLF)
UTF-8
22°C
Mostly clear
Search
ENG
IN
20:24
05-12-2022


5_12_2022_multithreading_concepts - Paint
File
View
Nitin Manjunath
Consolas
14
Background fill
JVM
3secs
3secs
CPU
Based on priority
it will decide
T.S
if both the threads
have same priority
then, TS uses
algorithm which is
vendor dependant.
shared
main
(5)
Thread-0
(5)
JVM will create
main thread and it
only starts with a
default priority "5"
T.s
5
10
(high priority cpu
time will be allocated)
Circle
PM of a country
#1
597, 718px
3490 × 3885px
Size: 92.0KB
100%
22°C
Mostly clear
Search
ENG
IN
20:28
05-12-2022


*05_12_2022_threads_classnotes - Notepad
File Edit View
Nitin Manjunath
yield() => It causes to pause current executing Thread for giving chance for waiting Threads of same priority.
If there is no waiting Threads or all waiting Threads have low priority then same Thread can continue its execution.
If all the threads have same priority and if they are waiting then which thread will get chance we can't expect, it depends on ThreadScheduler.
The Thread which is yielded, when it will get the chance once again depends on the mercy on "ThreadScheduler" and we can't expect exactly.
public static native void yield()
MyThread t= new MyThread() //new state or born state
t.start() // enter into ready state/runnable state
if ThreadScheduler allocates processor then enters into running state.
a. if running Thread calls yield() then it enters into runnable state.
if run() is finished with execution then it enters into dead state.
eg#1.
Ln 80, Col 17
120%
Windows (CRLF)
UTF-8
22°C
Mostly clear
Search
ENG
IN
20:52
05-12-2022

05_12_2022_threads_classnotes - Notepad

File Edit View

```
        }
    }
}
```

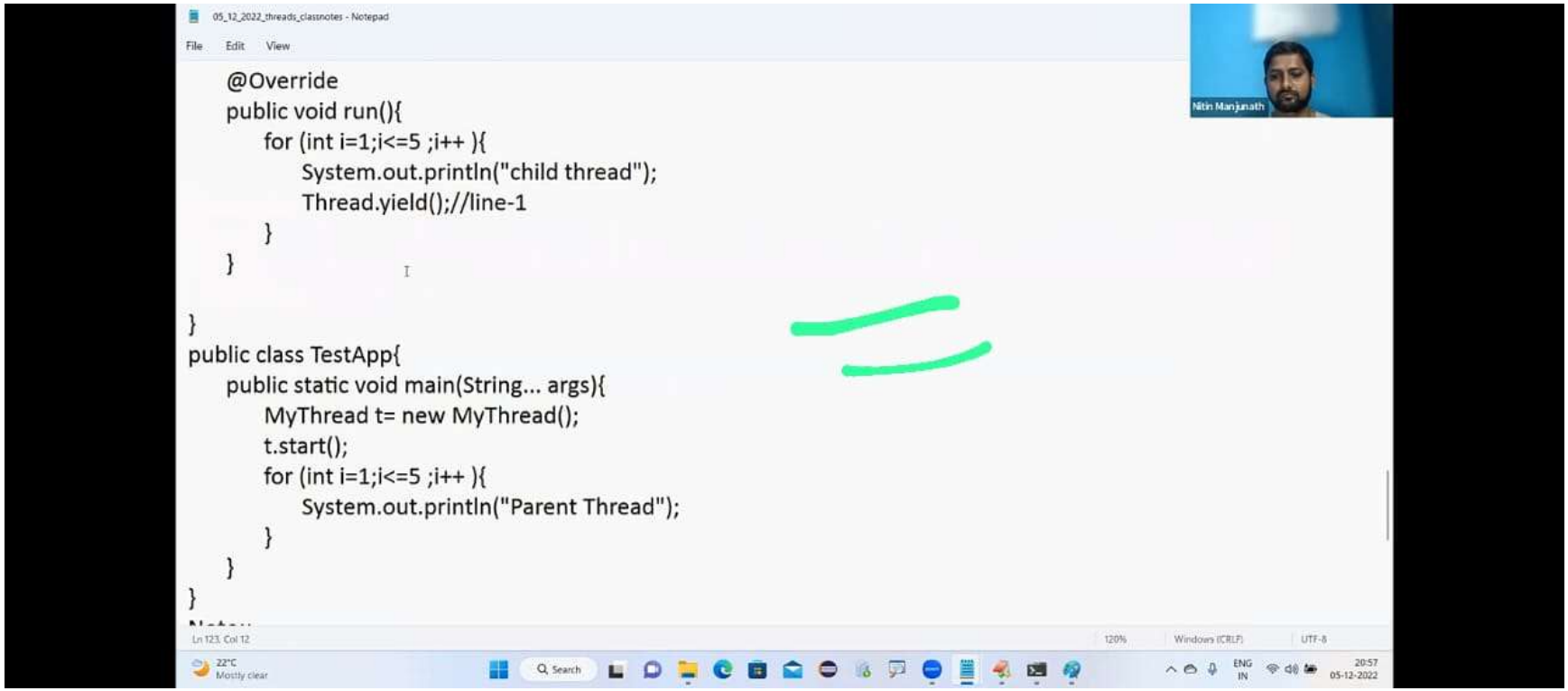
Note::

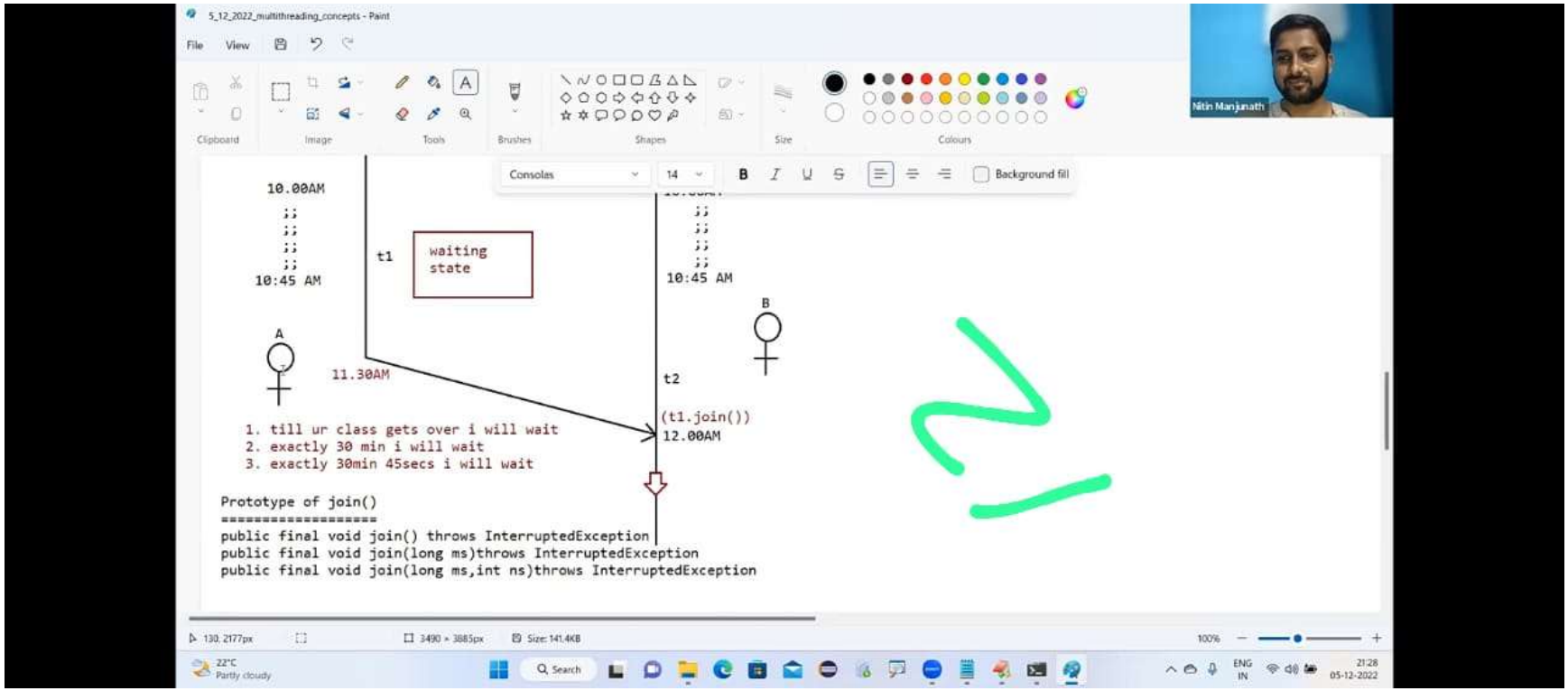
If we comment line-1, then we can't expect the output becoz both the threads have same priority then which thread but if we don't comment line-1, then there is a possibility of main thread getting more no of times,
so main thread execution is faster then child thread will get chance.

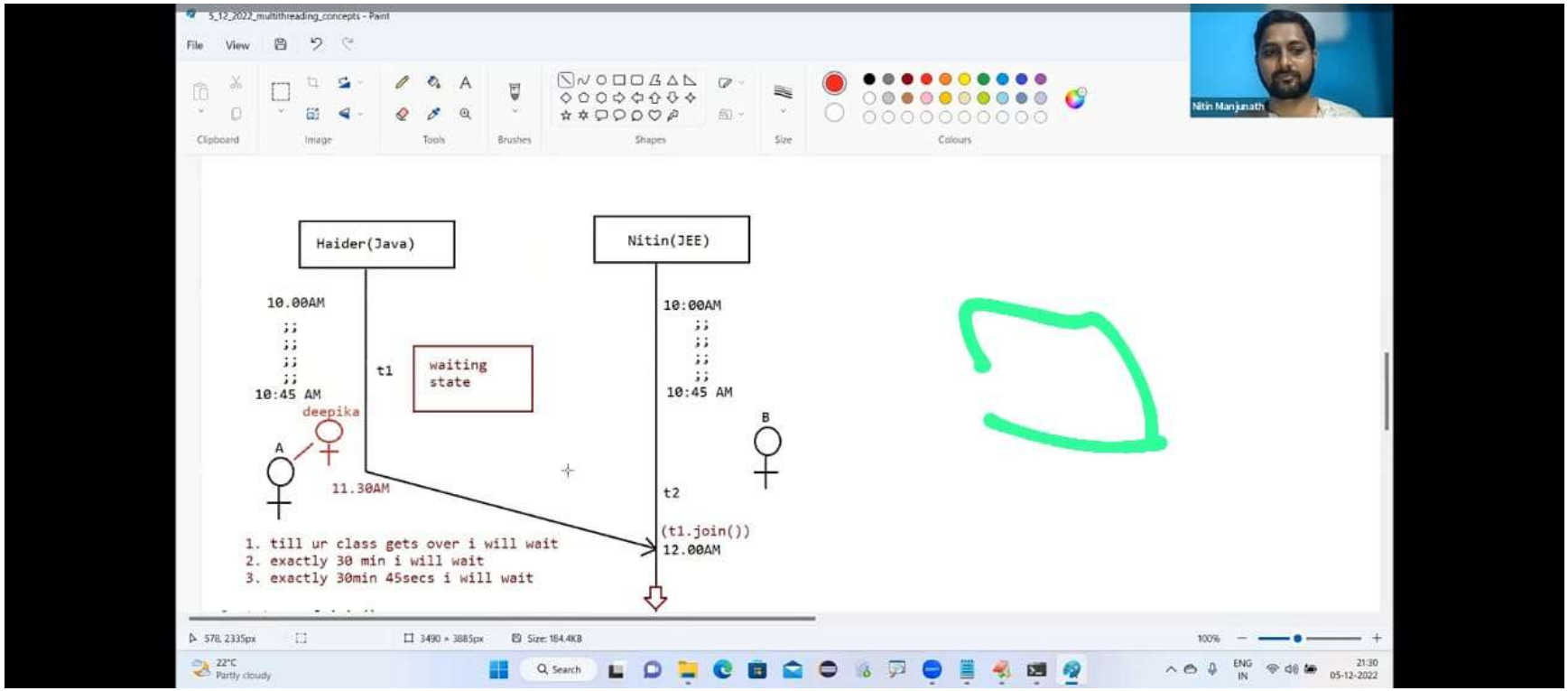
Note: Some platforms wont provide proper support for yield(),becuase it is getting the execution code from other language prefereably from 'C'.

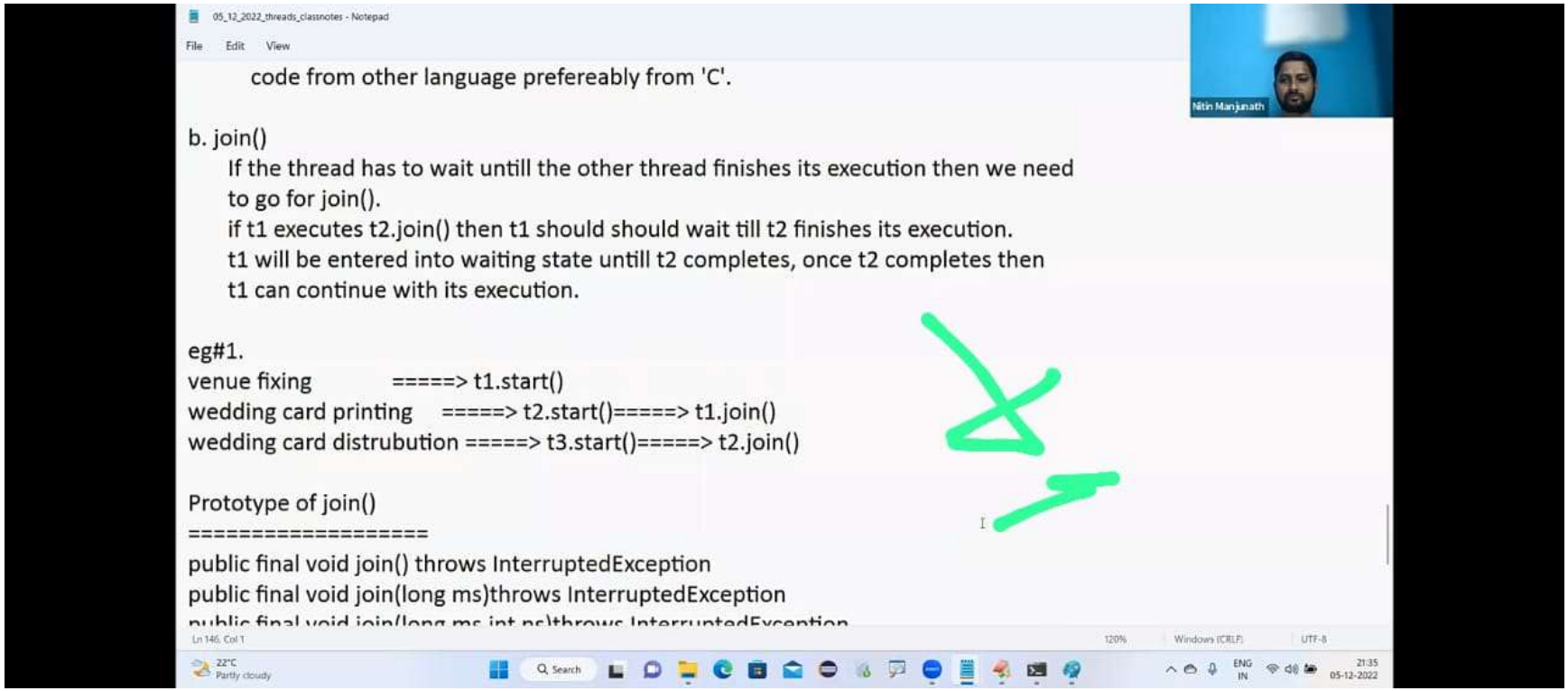
Ln 117, Col 1 120% Windows (CRLF) UTF-8

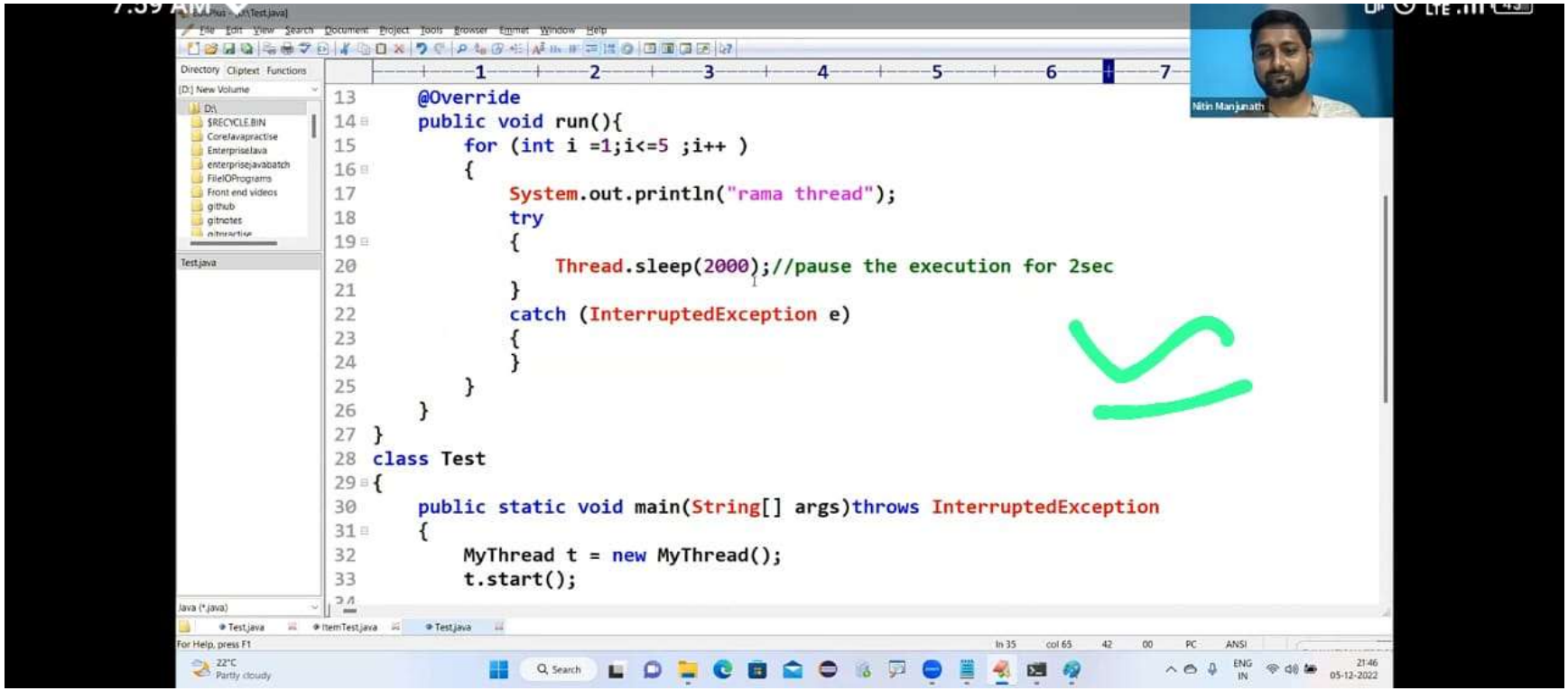
22°C Mostly clear Search ENG IN 20:56 05-12-2022

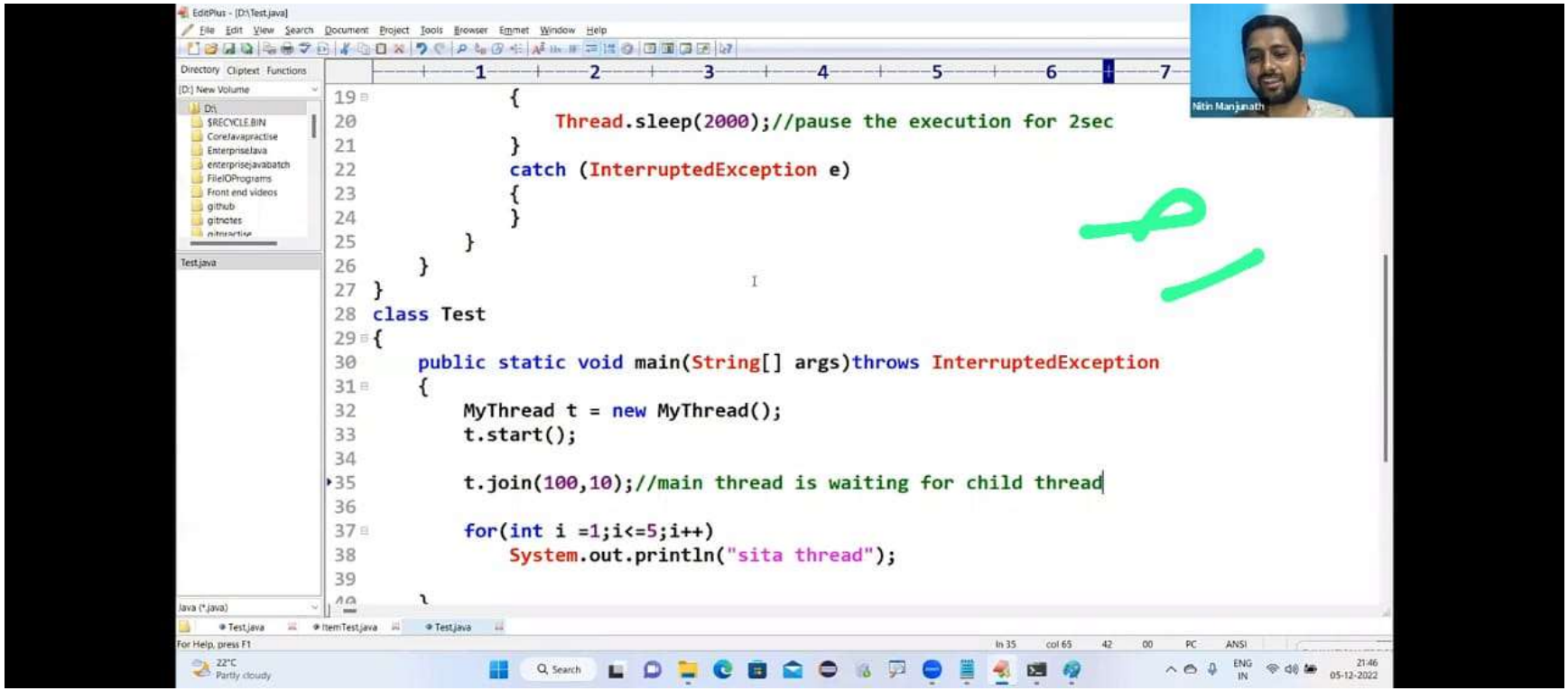

05_12_2022_threads_classnotes - Notepad
File Edit View
@Override
public void run(){
for (int i=1;i<=5 ;i++){
System.out.println("child thread");
Thread.yield();//line-1
}
}
}
public class TestApp{
public static void main(String... args){
MyThread t= new MyThread();
t.start();
for (int i=1;i<=5 ;i++){
System.out.println("Parent Thread");
}
}
}
Nitin Manjunath
Ln 123, Col 12
120%
Windows (CRLF)
UTF-8
22°C
Mostly clear
Search
ENG
IN
20:57
05-12-2022

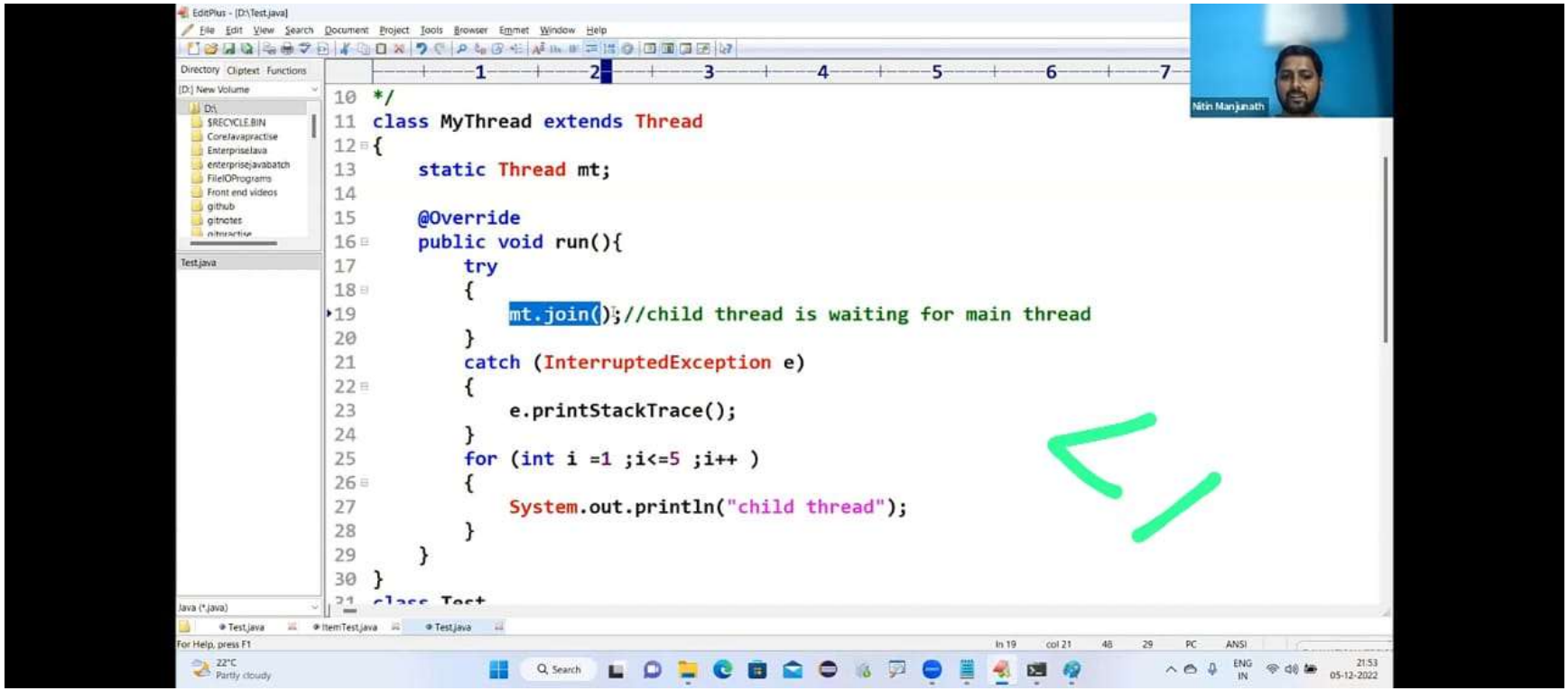

5_12_2022_multithreading_concepts - Paint
File
View
Nitin Manjunath
Consolas
14
Background fill
10.00AM
10:45 AM
t1
waiting state
A
11.30AM
B
10:45 AM
t2
(t1.join())
12.00AM
1. till ur class gets over i will wait
2. exactly 30 min i will wait
3. exactly 30min 45secs i will wait
Prototype of join()
public final void join() throws InterruptedException
public final void join(long ms)throws InterruptedException
public final void join(long ms,int ns)throws InterruptedException

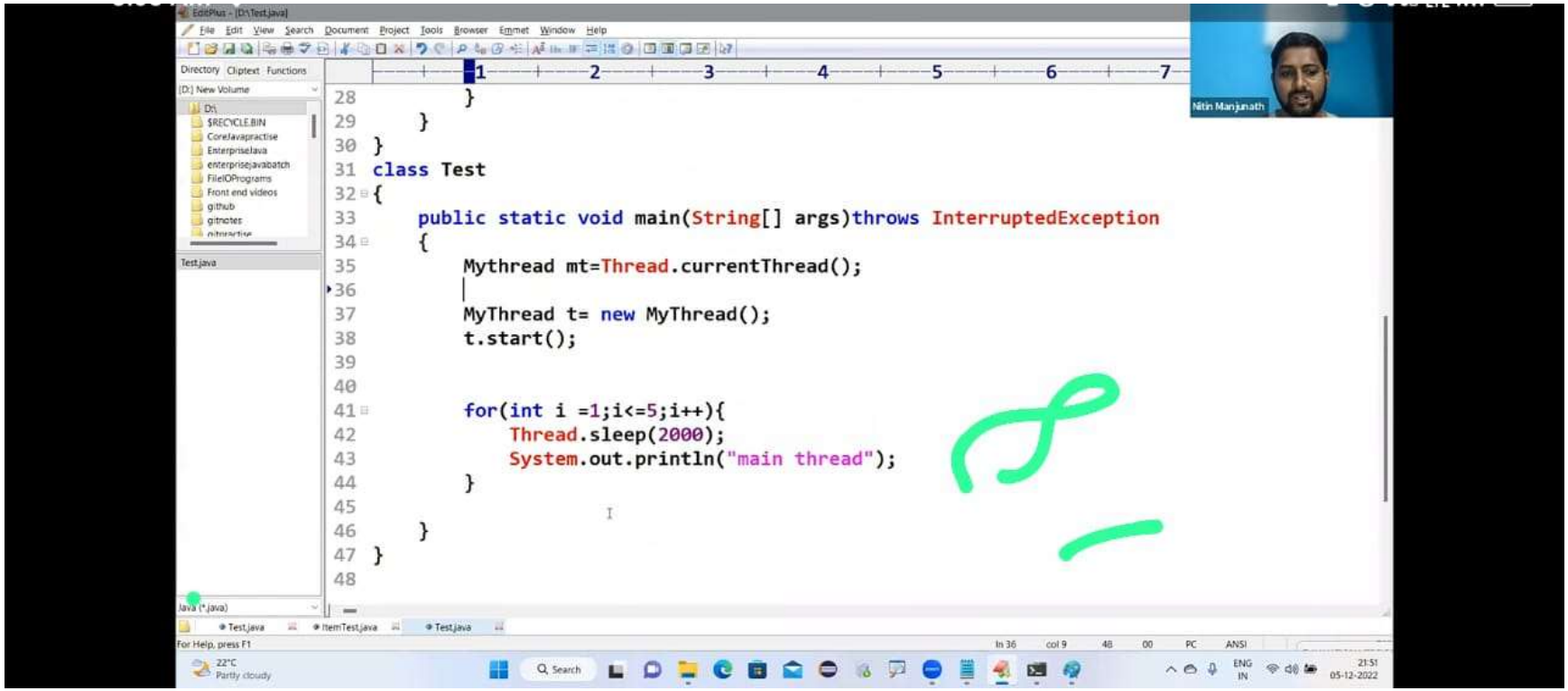

5_12_2022_multithreading_concepts - Paint
File
View
Nitin Manjunath
Haider(Java)
Nitin(JEE)
10.00AM
10:00AM
10:45 AM
10:45 AM
t1
waiting state
deepika
A
B
11.30AM
t2
(t1.join())
12.00AM
1. till ur class gets over i will wait
2. exactly 30 min i will wait
3. exactly 30min 45secs i will wait

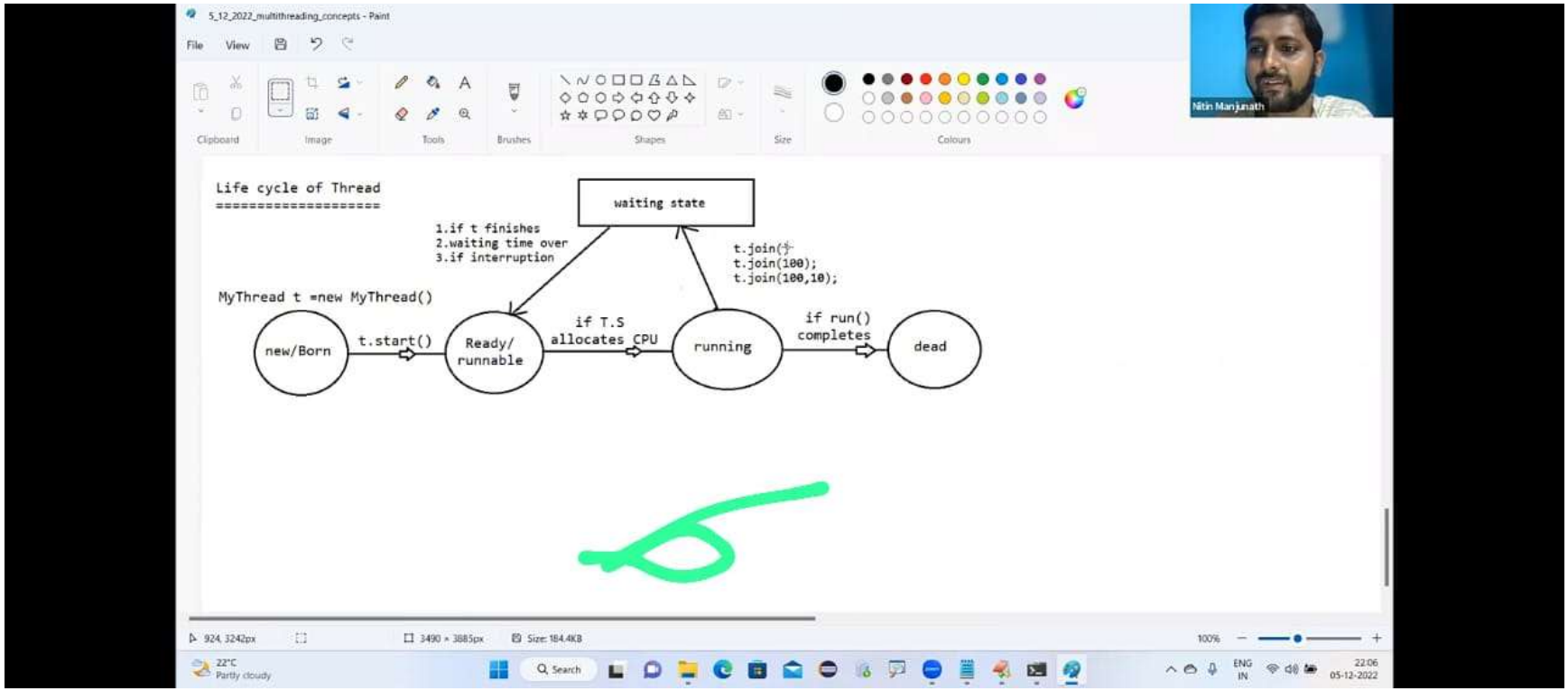

05_12_2022_threads_classnotes - Notepad
File Edit View
code from other language prefereably from 'C'.
b. join()
If the thread has to wait untill the other thread finishes its execution then we need to go for join().
if t1 executes t2.join() then t1 should should wait till t2 finishes its execution.
t1 will be entered into waiting state untill t2 completes, once t2 completes then t1 can continue with its execution.
eg#1.
venue fixing =====> t1.start()
wedding card printing =====> t2.start()=====> t1.join()
wedding card distrubution =====> t3.start()=====> t2.join()
Prototype of join()
==========
public final void join() throws InterruptedException
public final void join(long ms)throws InterruptedException
Nitin Manjunath
Ln 146, Col 1
120%
Windows (CRLF)
UTF-8
22°C
Partly cloudy
Search
ENG
IN
21:35
05-12-2022

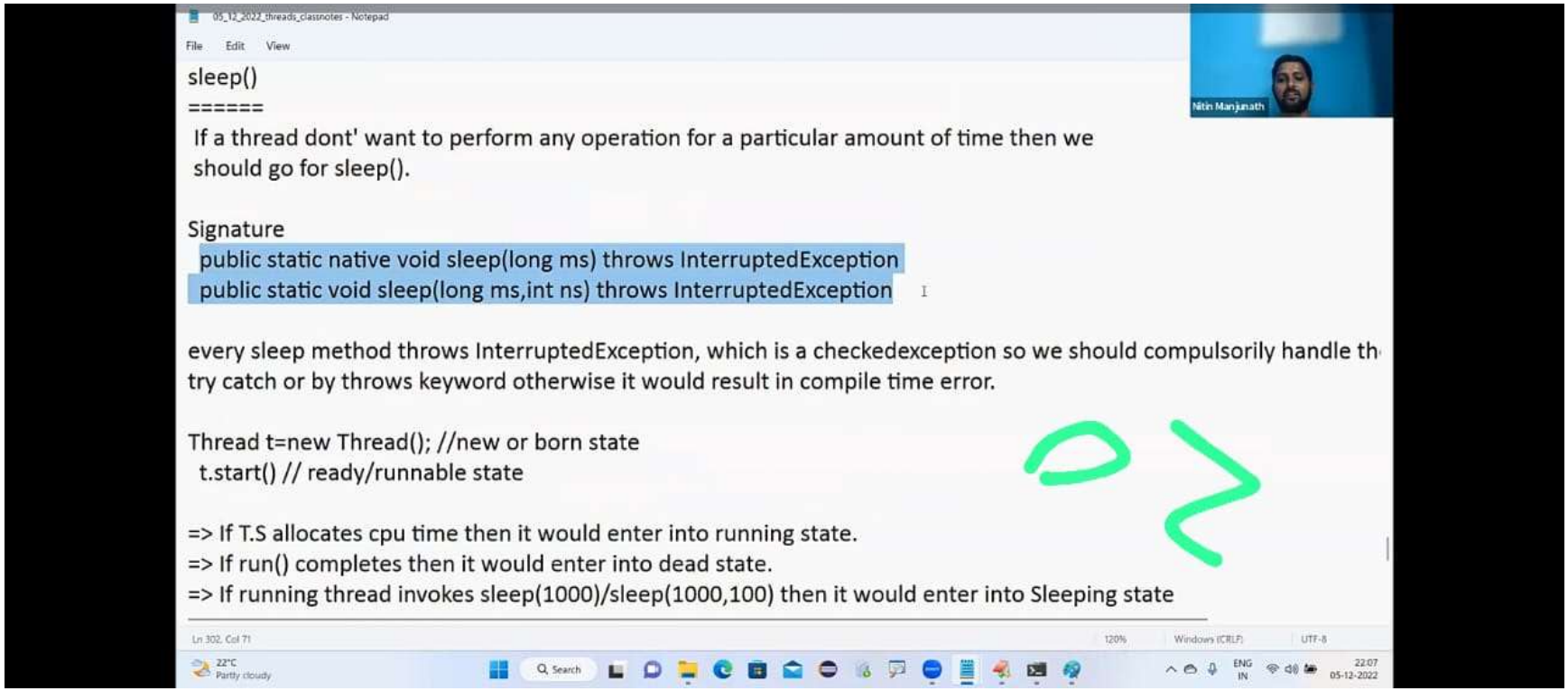

Nitin Manjunath
@Override
public void run(){
for (int i =1;i<=5 ;i++)
{
System.out.println("rama thread");
try
{
Thread.sleep(2000);//pause the execution for 2sec
}
catch (InterruptedException e)
{
}
}
}
class Test
{
public static void main(String[] args)throws InterruptedException
{
MyThread t = new MyThread();
t.start();

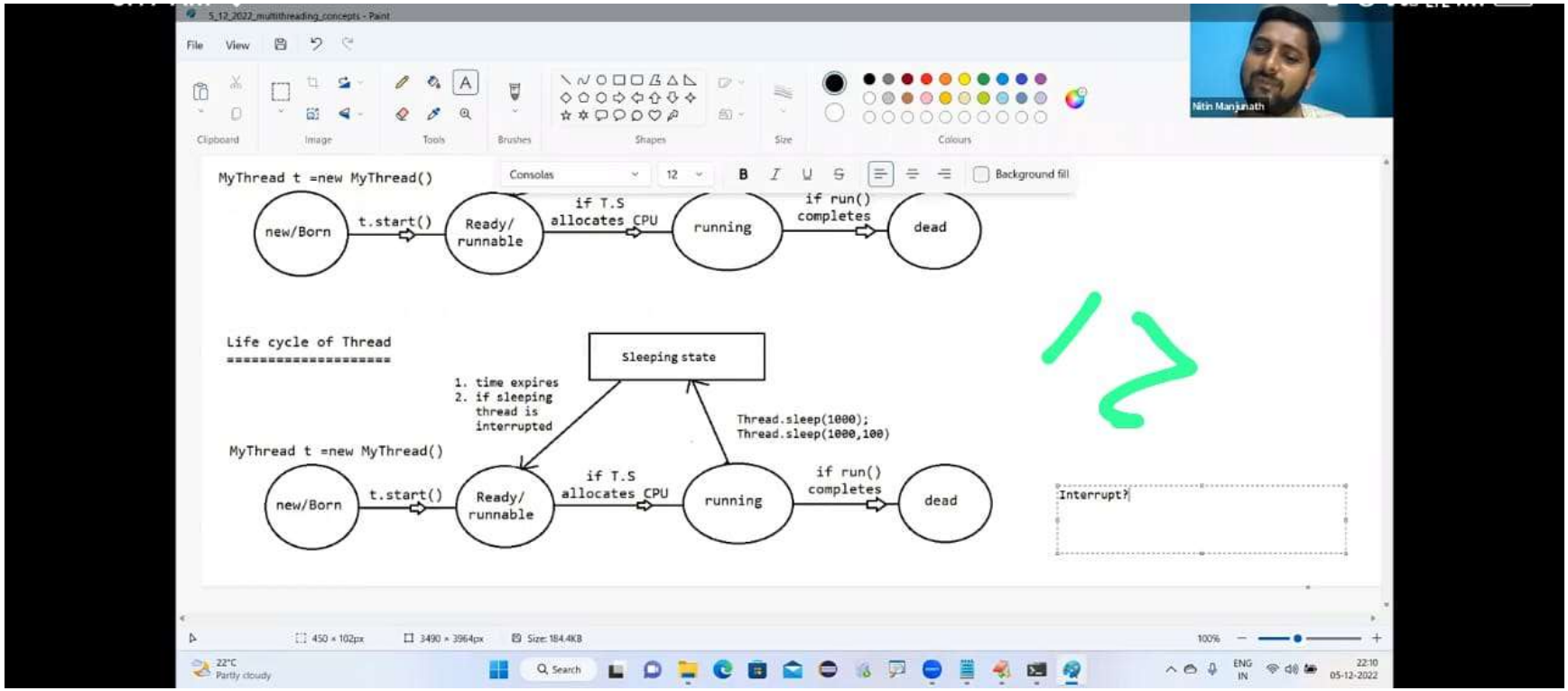

EditPlus - [D:\Test.java]
File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
[D:] New Volume
D:\
$RECYCLE.BIN
CoreJavapractise
EnterpriseJava
enterprisejavabatch
FileIOPrograms
Front end videos
github
gitnotes
Test.java
Nitin Manjunath
19 {
20 Thread.sleep(2000);//pause the execution for 2sec
21 }
22 catch (InterruptedException e)
23 {
24 }
25 }
26 }
27 }
28 class Test
29 {
30 public static void main(String[] args)throws InterruptedException
31 {
32 MyThread t = new MyThread();
33 t.start();
34
35 t.join(100,10);//main thread is waiting for child thread
36
37 for(int i =1;i<=5;i++)
38 System.out.println("sita thread");
39
Test.java
ItemTest.java
Test.java
For Help, press F1
ln 35 col 65 42 00 PC ANSI
22°C
Partly cloudy
Search
ENG
IN
21:46
05-12-2022

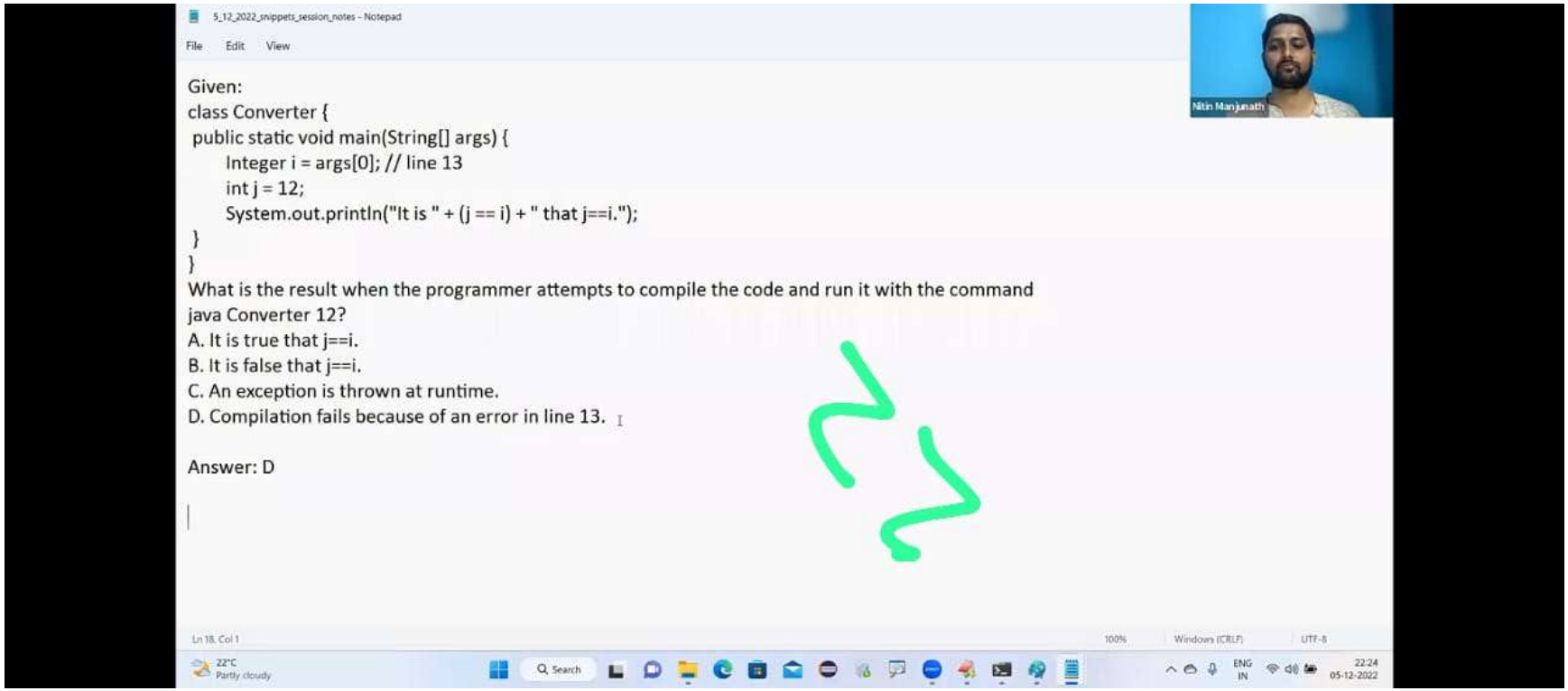

EditPlus - [D:\Test.java]
File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
[D:] New Volume
D:\
$RECYCLE.BIN
CoreJavapractise
Enterprisejava
enterprisejavabatch
FileIOPrograms
Front end videos
github
gitnotes
Test.java
Nitin Manjunath
*/
class MyThread extends Thread
{
static Thread mt;
@Override
public void run(){
try
{
mt.join();//child thread is waiting for main thread
}
catch (InterruptedException e)
{
e.printStackTrace();
}
for (int i =1 ;i<=5 ;i++)
{
System.out.println("child thread");
}
}
}
Test.java
ItemTest.java
Test.java
For Help, press F1
ln 19 col 21 48 29 PC ANSI
22°C
Partly cloudy
Search
ENG
IN
21:53
05-12-2022

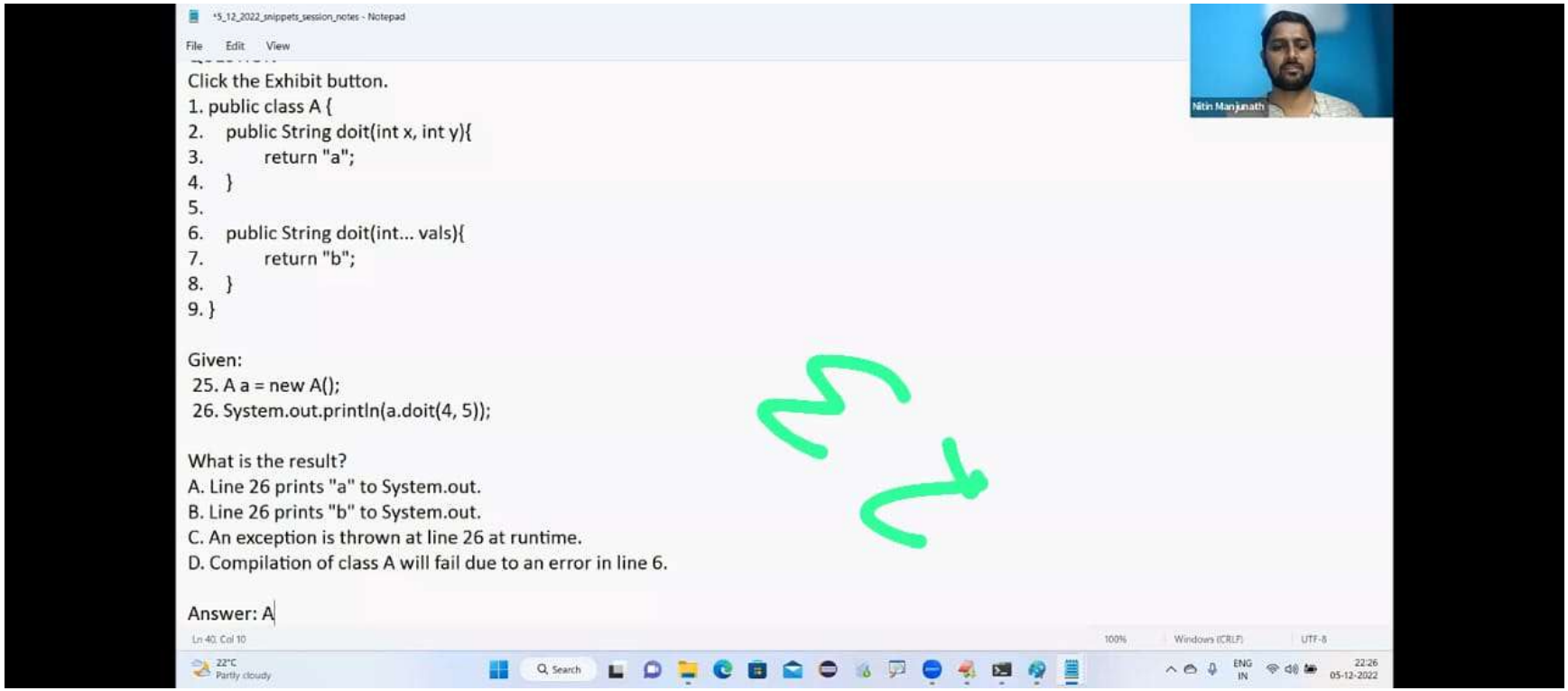

Nitin Manjunath
}
}
class Test
{
public static void main(String[] args)throws InterruptedException
{
Mythread mt=Thread.currentThread();
MyThread t= new MyThread();
t.start();
for(int i =1;i<=5;i++){
Thread.sleep(2000);
System.out.println("main thread");
}
}
}


5_12_2022_multithreading_concepts - Paint
File
View
Clipboard
Image
Tools
Brushes
Shapes
Size
Colours
Nitin Manjunath
Life cycle of Thread
==========
waiting state
1.if t finishes
2.waiting time over
3.if interruption
t.join();
t.join(100);
t.join(100,10);
MyThread t =new MyThread()
new/Born
t.start()
Ready/
runnable
if T.S
allocates CPU
running
if run()
completes
dead
924, 3242px
3490 × 3885px
Size: 184.4KB
100%
22°C
Partly cloudy
Search
ENG
IN
22:06
05-12-2022


05_12_2022_threads_classnotes - Notepad
File Edit View
sleep()
======
If a thread dont' want to perform any operation for a particular amount of time then we should go for sleep().
Signature
public static native void sleep(long ms) throws InterruptedException
public static void sleep(long ms,int ns) throws InterruptedException
every sleep method throws InterruptedException, which is a checkedexception so we should compulsorily handle th
try catch or by throws keyword otherwise it would result in compile time error.
Thread t=new Thread(); //new or born state
t.start() // ready/runnable state
=> If T.S allocates cpu time then it would enter into running state.
=> If run() completes then it would enter into dead state.
=> If running thread invokes sleep(1000)/sleep(1000,100) then it would enter into Sleeping state
Nitin Manjunath
Ln 302, Col 71
120%
Windows (CRLF)
UTF-8
22°C
Partly cloudy
Search
ENG
IN
22:07
05-12-2022


5_12_2022_multithreading_concepts - Paint
File
View
Clipboard
Image
Tools
Brushes
Shapes
Size
Colours
Nitin Manjunath
MyThread t =new MyThread()
Consolas
12
Background fill
new/Born
t.start()
Ready/
runnable
if T.S
allocates CPU
running
if run()
completes
dead
Life cycle of Thread
Sleeping state
1. time expires
2. if sleeping
thread is
interrupted
Thread.sleep(1000);
Thread.sleep(1000,100)
MyThread t =new MyThread()
new/Born
t.start()
Ready/
runnable
if T.S
allocates CPU
running
if run()
completes
dead
Interrupt?
17
450 × 102px
3490 × 3964px
Size: 184.4KB
100%
22°C
Partly cloudy
Search
ENG
IN
22:10
05-12-2022


5_12_2022_snippets_session_notes - Notepad
File Edit View
Given:
class Converter {
public static void main(String[] args) {
Integer i = args[0]; // line 13
int j = 12;
System.out.println("It is " + (j == i) + " that j==i.");
}
}
What is the result when the programmer attempts to compile the code and run it with the command
java Converter 12?
A. It is true that j==i.
B. It is false that j==i.
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 13.
Answer: D
Nitin Manjunath
Ln 18, Col 1
100%
Windows (CRLF)
UTF-8
22°C
Partly cloudy
Search
ENG
IN
22:24
05-12-2022


*5_12_2022_snippets_session_notes - Notepad
File Edit View
Click the Exhibit button.
1. public class A {
2. public String doit(int x, int y){
3. return "a";
4. }
5.
6. public String doit(int... vals){
7. return "b";
8. }
9. }
Given:
25. A a = new A();
26. System.out.println(a.doit(4, 5));
What is the result?
A. Line 26 prints "a" to System.out.
B. Line 26 prints "b" to System.out.
C. An exception is thrown at line 26 at runtime.
D. Compilation of class A will fail due to an error in line 6.
Answer: A
Nitin Manjunath
Ln 40, Col 10
100%
Windows (CRLF)
UTF-8
22°C
Partly cloudy
Search
ENG
IN
22:26
05-12-2022