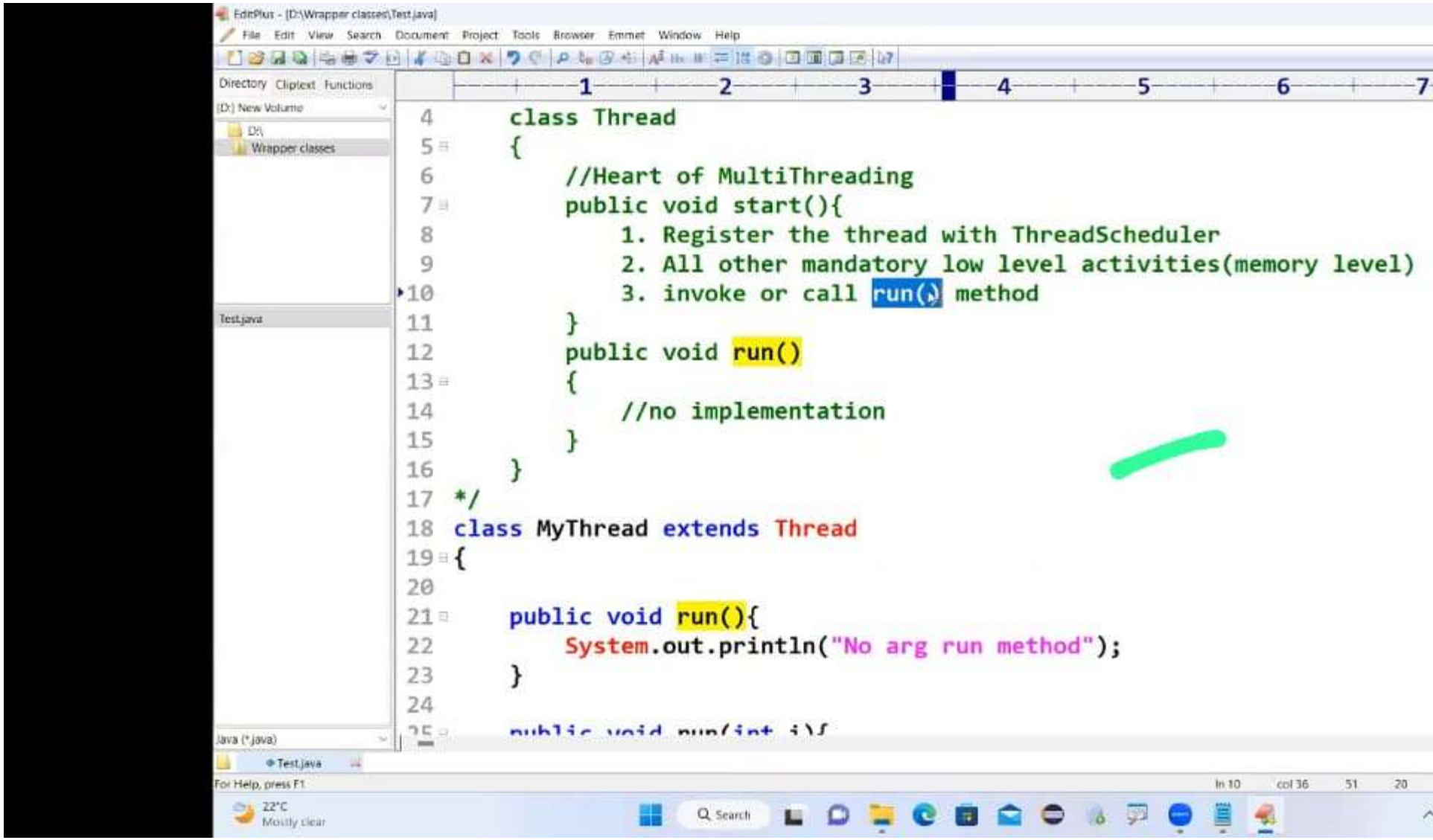
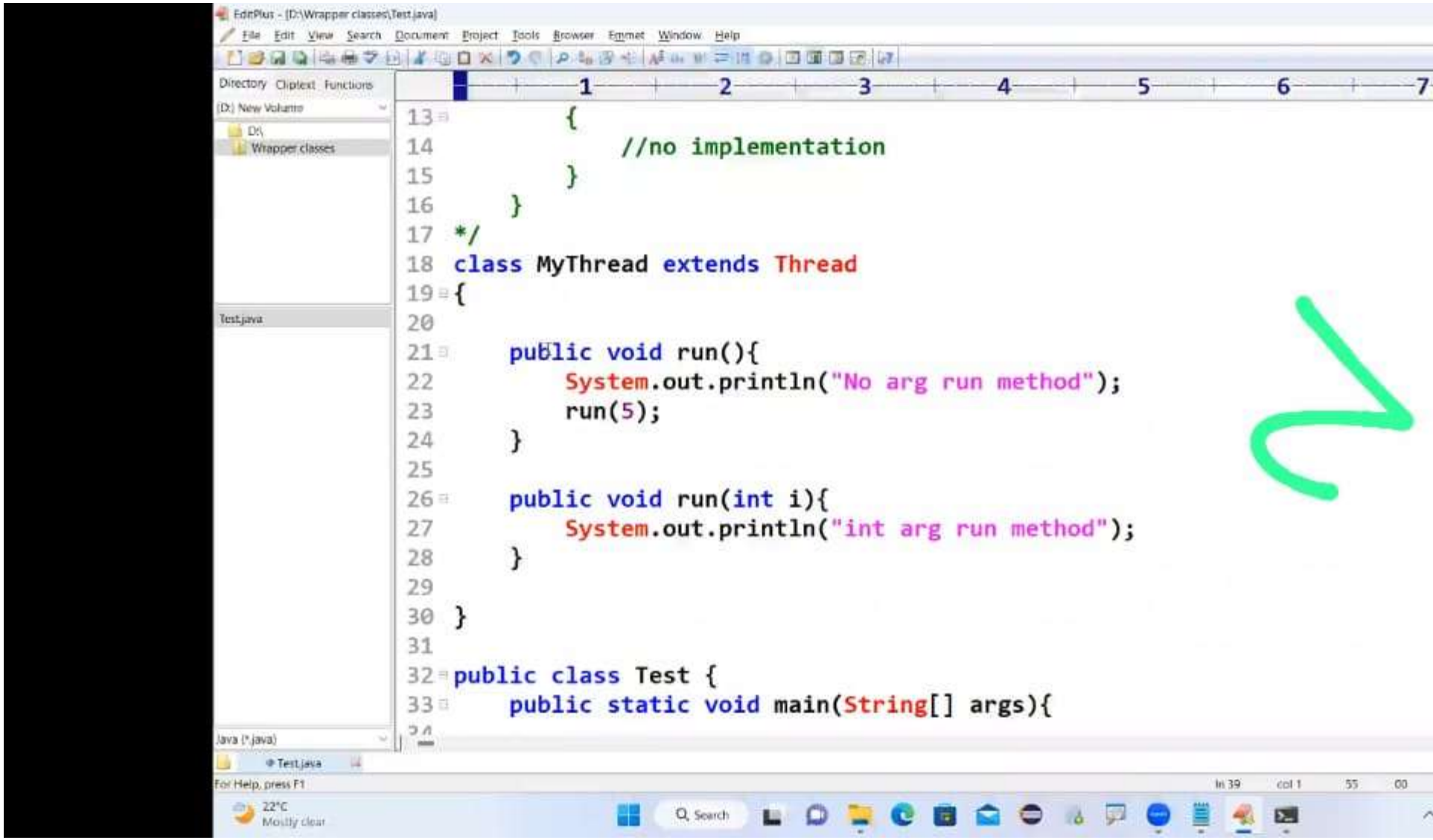
Java Multithreading Part2




EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Format Window Help
Directory Cliptext Functions
(D:) New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
Ln 39 col 1 55 00
22°C
Mostly clear
Search
{
//no implementation
}
}
*/
class MyThread extends Thread
{
public void run(){
System.out.println("No arg run method");
run(5);
}
public void run(int i){
System.out.println("int arg run method");
}
}
public class Test {
public static void main(String[] args){

2_12_2022_Threads_classnotes - Notepad

File Edit View

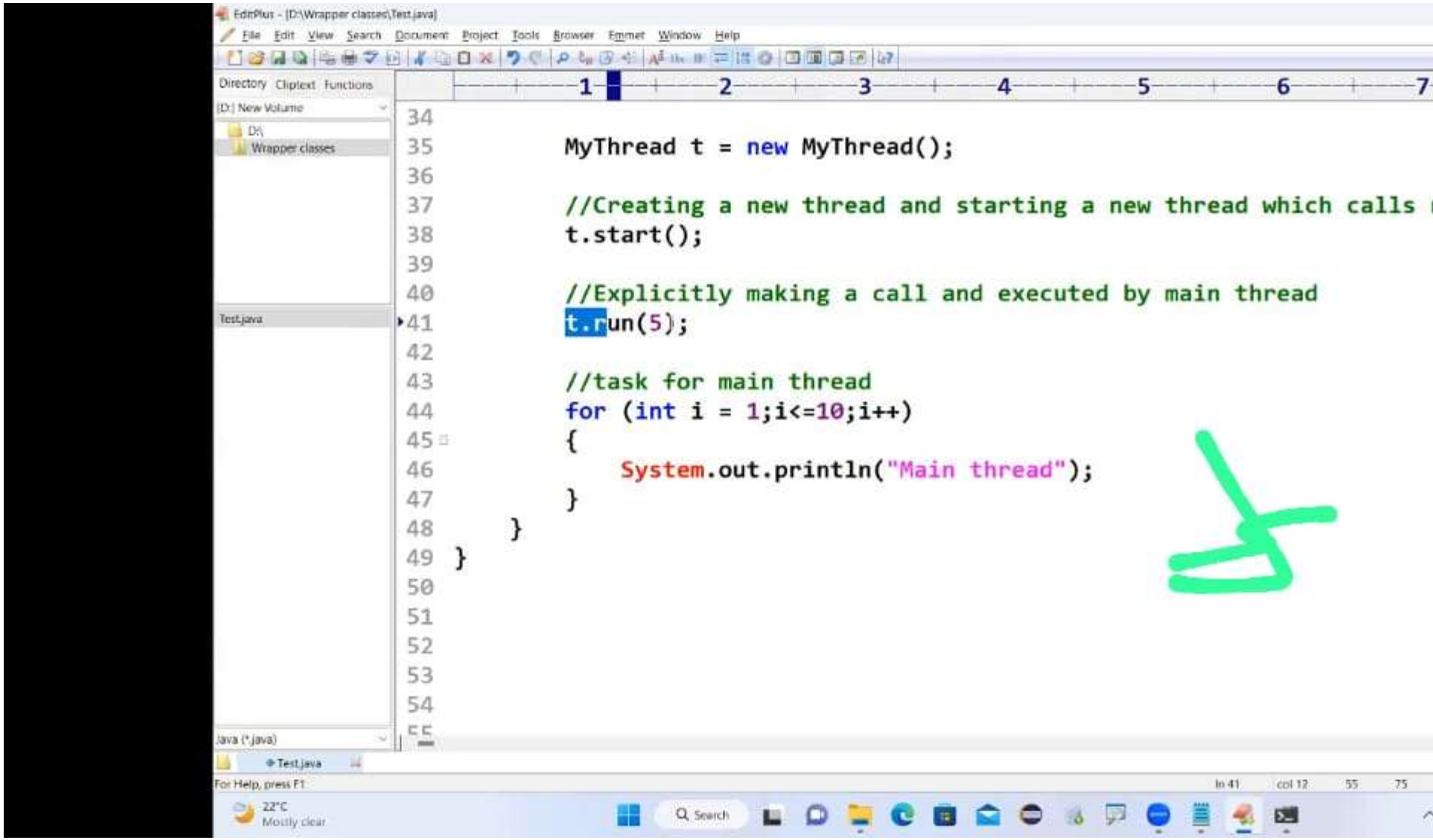
case5:Overloading of run() method
we can overload run() method but Thread class start() will always call run() with zero argument.
if we overload run method with arguments, then we need to explicitly call argument based run method and it will be normal method.
eg::

```
class MyThread extends Thread{
	public void run(){
		System.out.println("no arg method");
	}
	public void run(int i){
		System.out.println("zero arg method");
	}
}
class ThreadDemo{
	public static void main(String... args){
		MyThread t=new MyThread();
		t.start();
	}
}
```

Output:: NO arg method.

Ln 21, Col 1 100%

22°C Mostly clear Search


EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
[D:] New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
Ln 41 col 12 55 75
22°C
Mostly clear
Search
34
35 MyThread t = new MyThread();
36
37 //Creating a new thread and starting a new thread which calls
38 t.start();
39
40 //Explicitly making a call and executed by main thread
41 t.run(5);
42
43 //task for main thread
44 for (int i = 1;i<=10;i++)
45 {
46 System.out.println("Main thread");
47 }
48 }
49 }
50
51
52
53
54
55

2_12_2022_Threads_classnotes - Notepad

File Edit View

Case6::Overriding of start() method

If we override start() then our start() method will be executed just like normal method,but no new Thread will be created and no new Thread will be started.

eg#1.

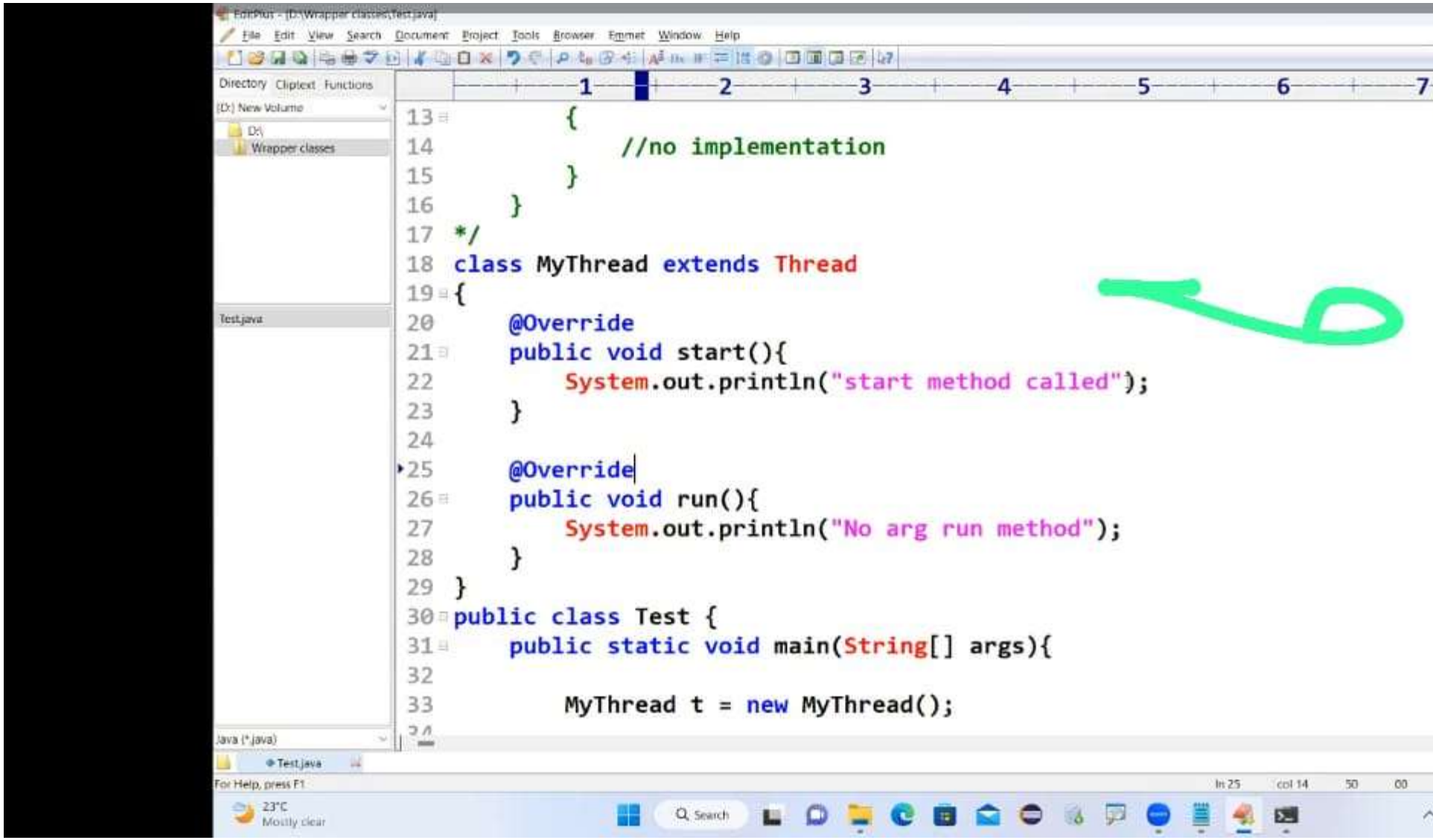
```
class MyThread extends Thread{
	public void run(){
		System.out.println("no arg method");
	}
	public void start(){
		System.out.println("start arg method");
	}
}
class ThreadDemo{
	public static void main(String... args){
		MyThread t=new MyThread();
		t.start();
	}
}
```

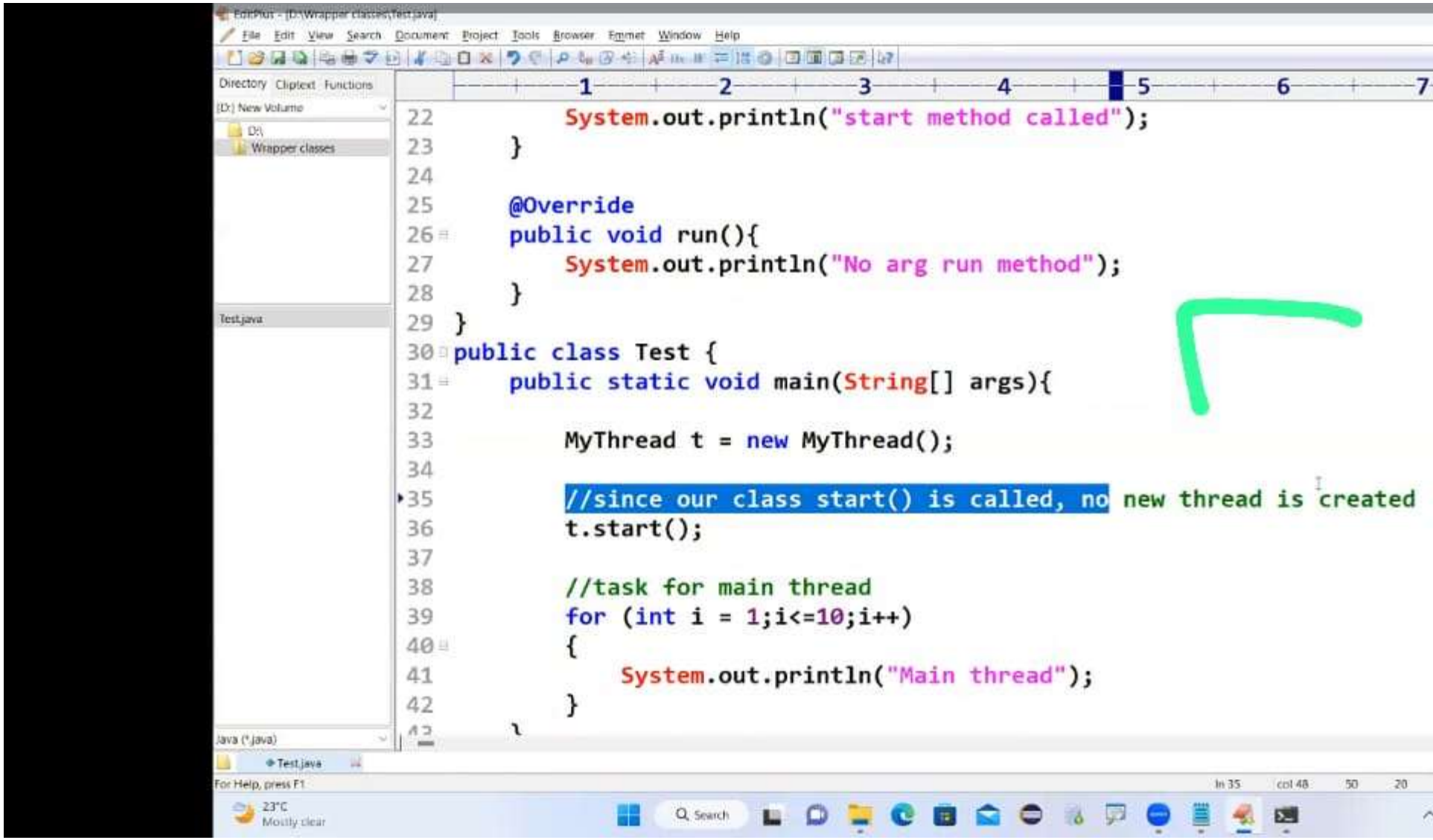
Output:: start arg method

It is never recomended to override start() method.

Ln 24, Col 1 100%

23°C Rain tomorrow Search


File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
(D:) New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
Ln 25 col 14
23°C
Mostly clear
Search
{
//no implementation
}
}
*/
class MyThread extends Thread
{
@Override
public void start(){
System.out.println("start method called");
}
@Override
public void run(){
System.out.println("No arg run method");
}
}
public class Test {
public static void main(String[] args){
MyThread t = new MyThread();




```
22          System.out.println("start method called");
23      }
24
25      @Override
26      public void run(){
27          System.out.println("No arg run method");
28      }
29  }
30  public class Test {
31      public static void main(String[] args){
32
33          MyThread t = new MyThread();
34
35          //since our class start() is called, no new thread is created
36          t.start();
37
38          //task for main thread
39          for (int i = 1;i<=10;i++)
40          {
41              System.out.println("Main thread");
42          }
```

2_12_2022_Threads_classnotes - Notepad

File Edit View

It is never recomended to override start() method.

case7::

```
class MyThread extends Thread{
    public void run(){
        System.out.println("run method");
    }
    public void start(){
        System.out.println("start  method");
    }
}
class ThreadDemo{
    public static void main(String... args){
        MyThread t=new MyThread();
        t.start();
        System.out.println("Main method");
    }
}
```

Ln 57, Col 13 110%

23°C Mostly clear

Search

2_12_2022_Threads_classnotes - Notepad

File Edit View

```
eg#2.
class MyThread extends Thread{
    public void start(){
        super.start();
        System.out.println("start  method");
    }
    public void run(){
        System.out.println("run method");
    }

}
class ThreadDemo{
    public static void main(String... args){
        MyThread t=new MyThread();
        t.start();
        System.out.println("Main method");
    }
}
Output::
```

Ln 64, Col 19 110%

23°C Mostly clear Search

2_12_2022_Threads_classnotes - Notepad

File Edit View

```
}
class ThreadDemo{
      public static void main(String... args){
            MyThread t=new MyThread();
            t.start();
            System.out.println("Main method");
      }
}
```

Output::

MainThread

a. Main method

b. start method

UserDefinedThread

a. run method

Ln 87, Col 18 110%

23°C Mostly clear

Search

2_12_2022_Threads_classnotes - Notepad

File Edit View

MainThread
 a. Main method
 b. start method
UserDefinedThread
 a. run method

case8:: Life cycle of a Thread
MyThread t=new MyThread(); // Thread is in born state
    t.start(); //Thread is in ready/runnable state
if Thread scheduler allocates CPU time then we say thread entered into Running state.
if run() is completed by thread then we say thread entered into dead state.

=> Once we created a Thread object then the Thread is said to be in new state or born state.
=> Once we call start() method then the Thread will be entered into Ready or Runnable state.
=> If Thread Scheduler allocates CPU then the Thread will be entered into running state.
=> Once run() method completes then the Thread will entered into dead state.

Ln 103, Col 1 110%

23°C Mostly clear Search

2_12_2022_threads - Paint

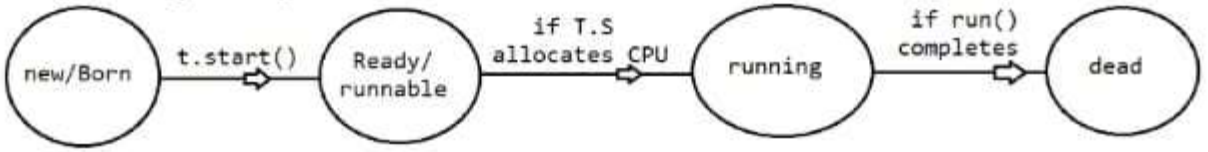
File View

Clipboard Image Tools Brushes Shapes Size Colours

Life cycle of Thread

==========

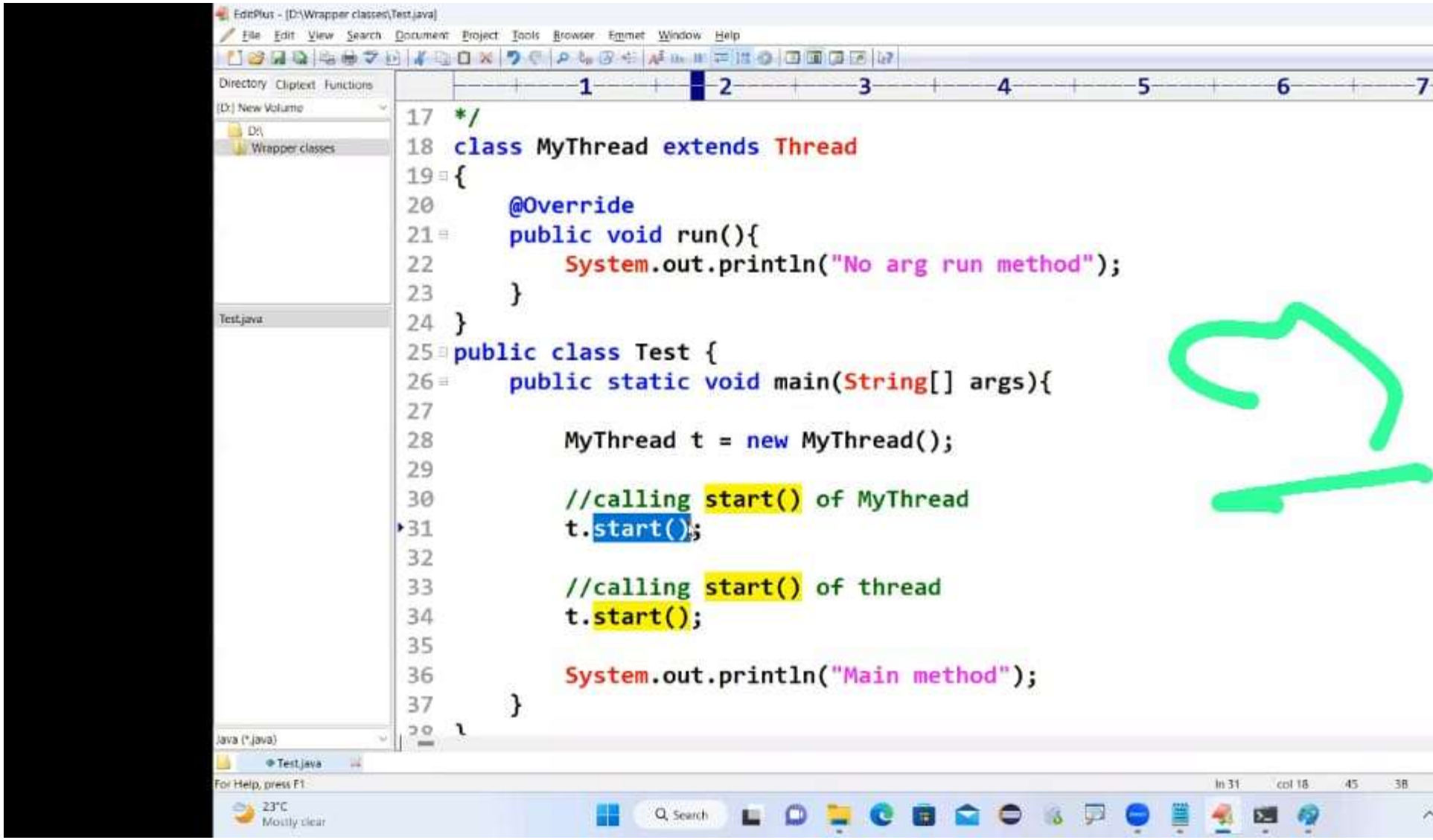
MyThread t =new MyThread()

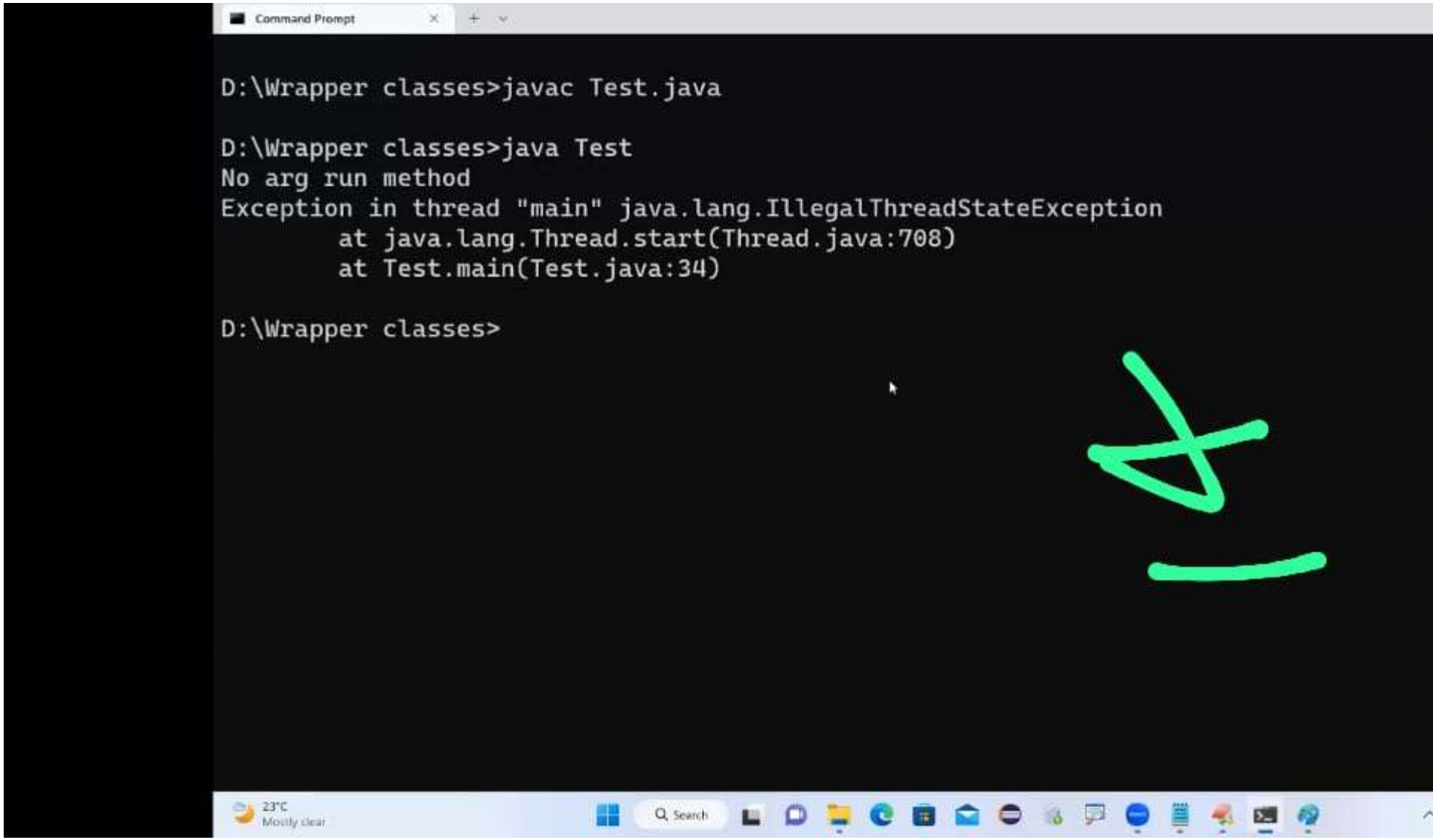


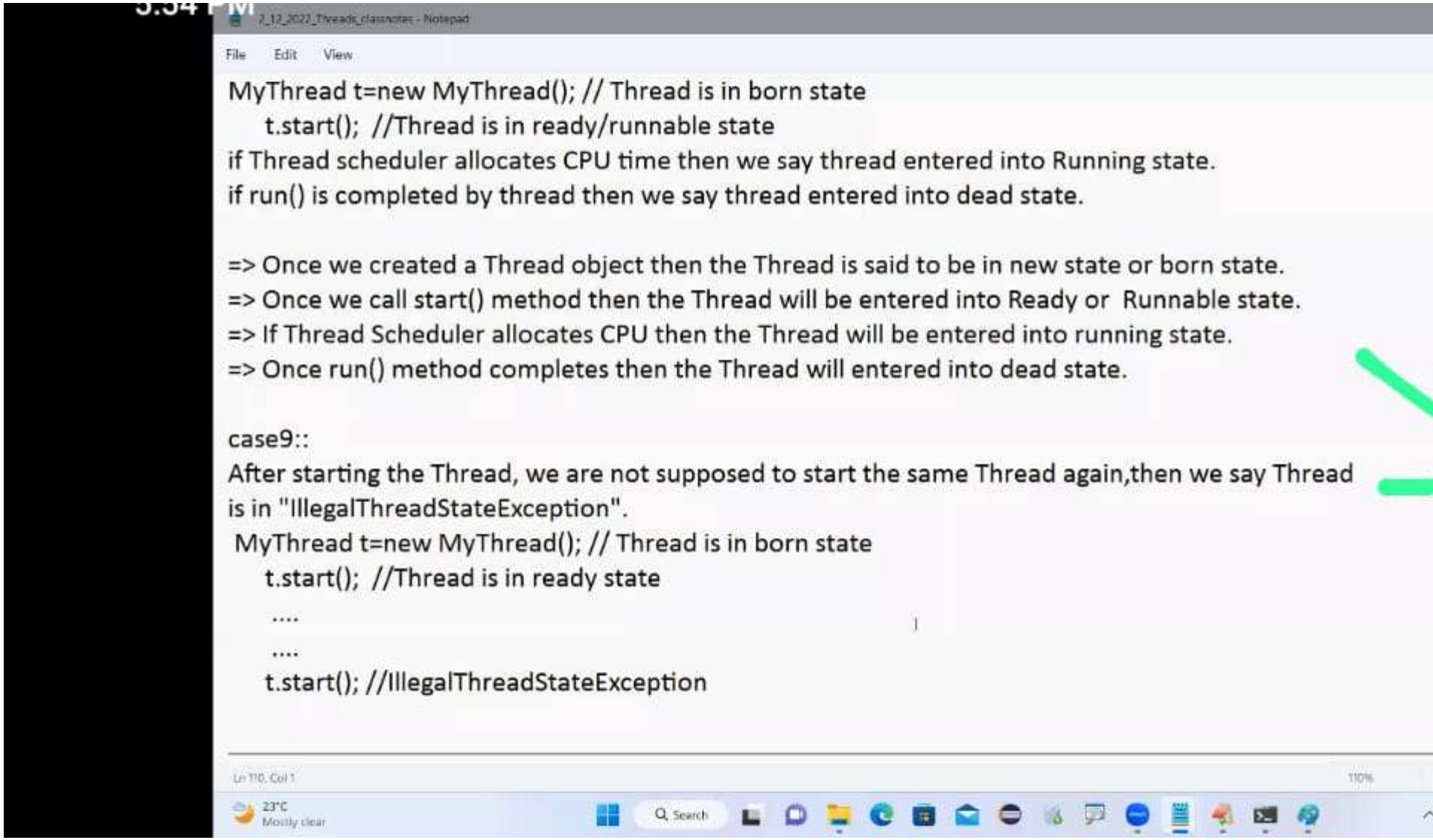
325, 350px 3490 × 3885px Size: 61.9KB

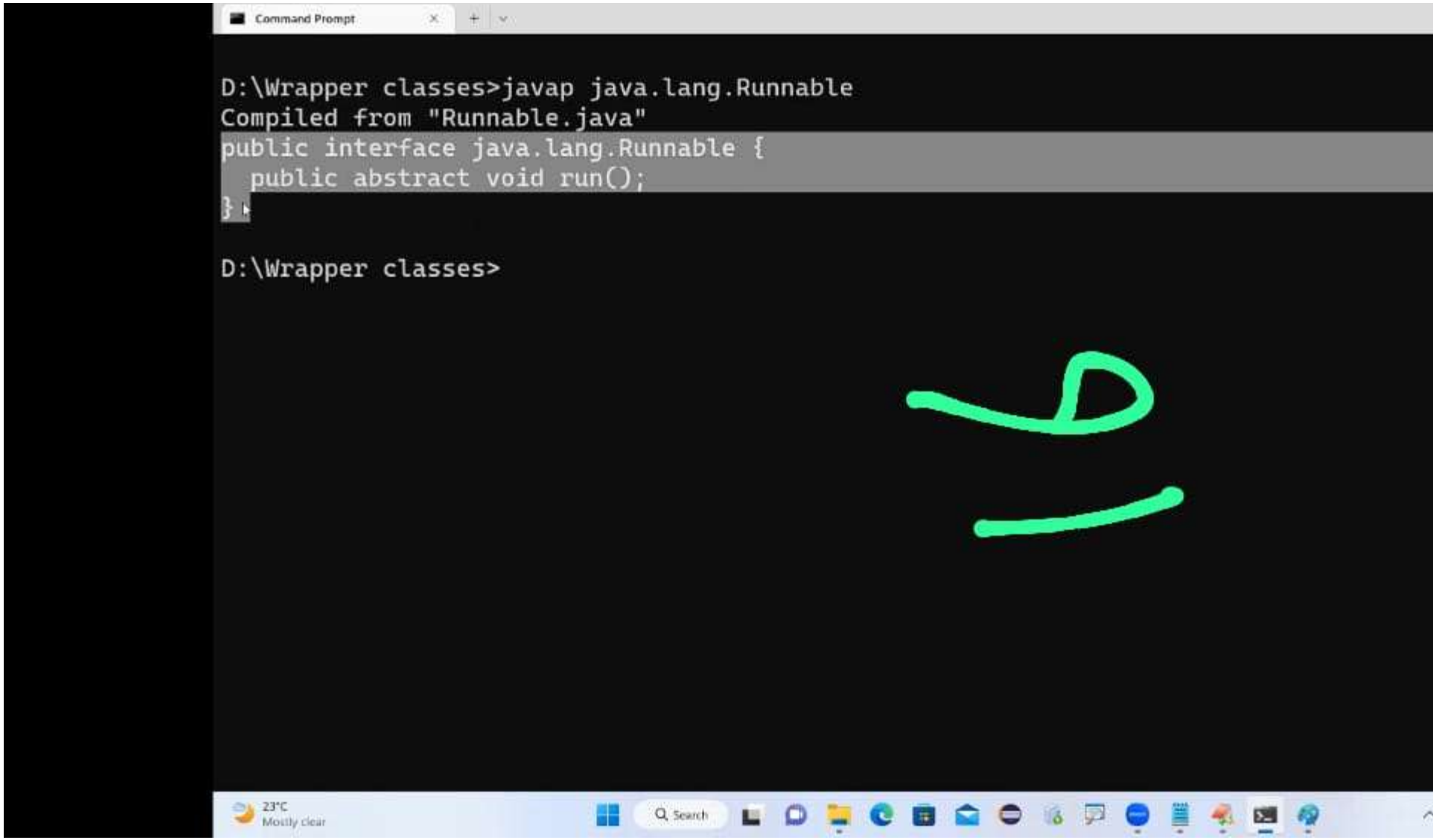
23°C Mostly clear

Search


EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
[D:] New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
*/
class MyThread extends Thread
{
@Override
public void run(){
System.out.println("No arg run method");
}
}
public class Test {
public static void main(String[] args){
MyThread t = new MyThread();
//calling start() of MyThread
t.start();
//calling start() of thread
t.start();
System.out.println("Main method");
}
23°C
Mostly clear
Search


Command Prompt
D:\Wrapper classes>javac Test.java
D:\Wrapper classes>java Test
No arg run method
Exception in thread "main" java.lang.IllegalThreadStateException
at java.lang.Thread.start(Thread.java:708)
at Test.main(Test.java:34)
D:\Wrapper classes>
23°C
Mostly clear
Search


2_12_2022_Threads_classnotes - Notepad
File Edit View
MyThread t=new MyThread(); // Thread is in born state
t.start(); //Thread is in ready/runnable state
if Thread scheduler allocates CPU time then we say thread entered into Running state.
if run() is completed by thread then we say thread entered into dead state.
=> Once we created a Thread object then the Thread is said to be in new state or born state.
=> Once we call start() method then the Thread will be entered into Ready or Runnable state.
=> If Thread Scheduler allocates CPU then the Thread will be entered into running state.
=> Once run() method completes then the Thread will entered into dead state.
case9::
After starting the Thread, we are not supposed to start the same Thread again,then we say Thread is in "IllegalThreadStateException".
MyThread t=new MyThread(); // Thread is in born state
t.start(); //Thread is in ready state
....
....
t.start(); //IllegalThreadStateException
Ln 110, Col 1
110%
23°C
Mostly clear
Search


Command Prompt
D:\Wrapper classes>javap java.lang.Runnable
Compiled from "Runnable.java"
public interface java.lang.Runnable {
public abstract void run();
}
D:\Wrapper classes>
23°C
Mostly clear
Search

*2_12_2022_Threads_classnotes - Notepad

File Edit View

2. Creation of a Thread requirement to SUNMS is an SRS

```
interface Runnable{
        void run();
}
class Thread implements Runnable{// Adapter class
    public void start(){
        1. Register the thread with ThreadScheduler
        2. All other mandatory low level activities(memory level)
        3. invoke or call run() method
    }
    public void run(){
        //job for a thread
    }
}
```

Ln 128, Col 23 110%

23°C Mostly clear Search

*2_12_2022_Threads_classnotes - Notepad

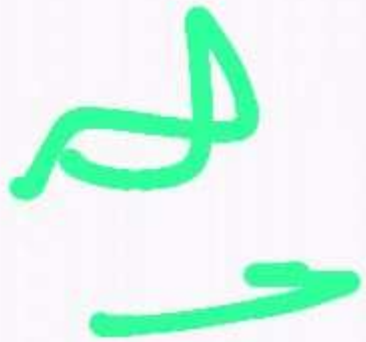
File Edit View

=> Once we call start() method then the Thread will be entered into Ready or Runnable state.
=> If Thread Scheduler allocates CPU then the Thread will be entered into running state.
=> Once run() method completes then the Thread will entered into dead state.

case9::
After starting the Thread, we are not supposed to start the same Thread again,then we say Thread is in "IllegalThreadStateException".

```
MyThread t=new MyThread(); // Thread is in born state
    t.start();  //Thread is in ready state
     ....
     ....
    t.start(); //IllegalThreadStateException
```

Creation of Thread using



Ln 111, Col 26 110%

23°C Mostly clear

Search

2_12_2022_Threads_classnotes - Notepad

File Edit View

Creation of Thread using Runnable interface

==========

1. Creating a Thread using java.lang.Thread class

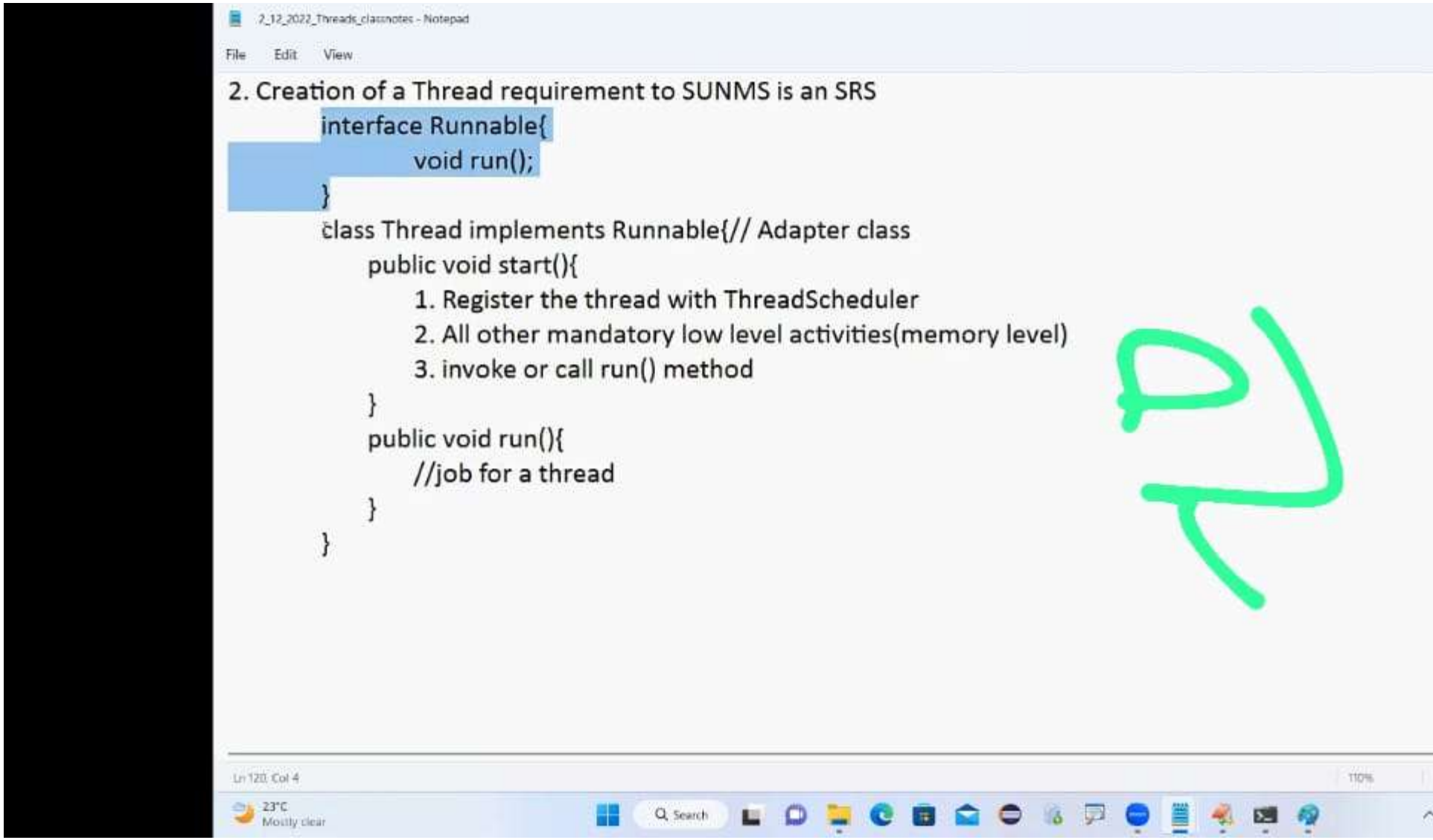
    a. use start() from Thread class

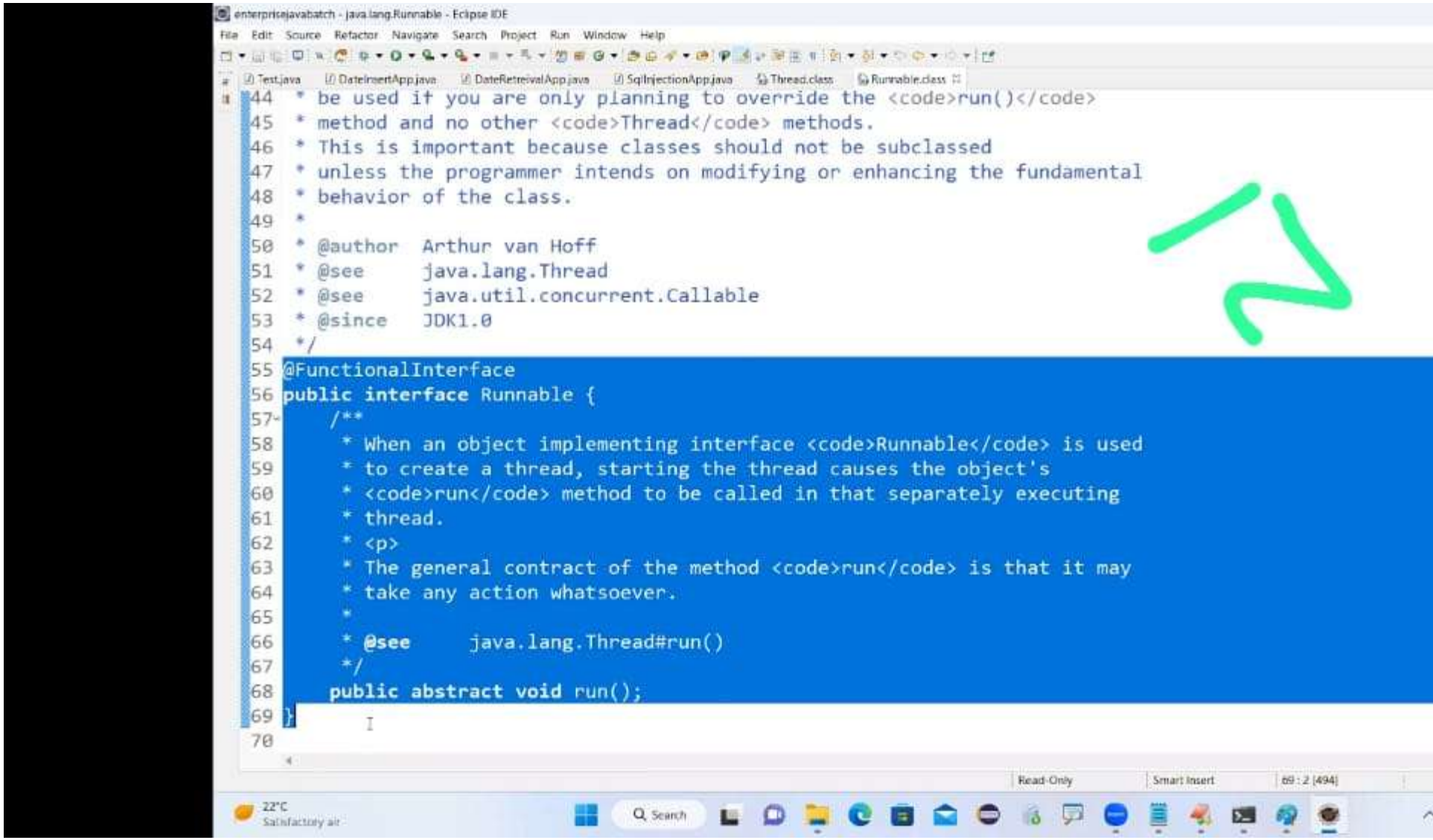
    b. override run() and define the job of the Thread.

Ln 117, Col 1 110%

23°C Mostly clear

Search


2_12_2022_Threads_classnotes - Notepad
File Edit View
2. Creation of a Thread requirement to SUNMS is an SRS
interface Runnable{
void run();
}
class Thread implements Runnable{// Adapter class
public void start(){
1. Register the thread with ThreadScheduler
2. All other mandatory low level activities(memory level)
3. invoke or call run() method
}
public void run(){
//job for a thread
}
}
Ln 120, Col 4
110%
23°C
Mostly clear
Search


enterprisejavabatch - java.lang.Runnable - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Test.java DateInsertApp.java DateRetreivalApp.java SqlInjectionApp.java Thread.class Runnable.class
* be used if you are only planning to override the <code>run()</code>
* method and no other <code>Thread</code> methods.
* This is important because classes should not be subclassed
* unless the programmer intends on modifying or enhancing the fundamental
* behavior of the class.
*
* @author Arthur van Hoff
* @see java.lang.Thread
* @see java.util.concurrent.Callable
* @since JDK1.0
*/
@FunctionalInterface
public interface Runnable {
/**
* When an object implementing interface <code>Runnable</code> is used
* to create a thread, starting the thread causes the object's
* <code>run</code> method to be called in that separately executing
* thread.
* <p>
* The general contract of the method <code>run</code> is that it may
* take any action whatsoever.
*
* @see java.lang.Thread#run()
*/
public abstract void run();
}
Read-Only
Smart Insert
69 : 2 [494]
22°C
Satisfactory air
Search

*2_12_2022_Threads_classnotes - Notepad

File Edit View

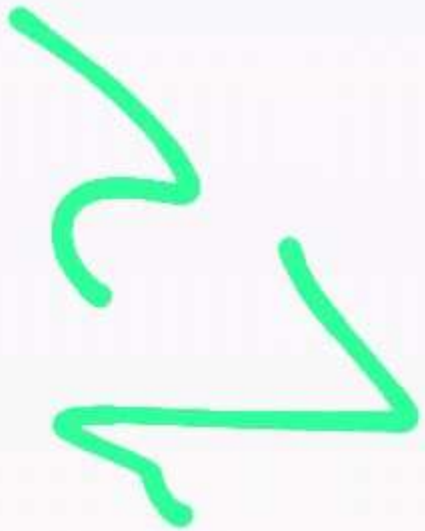
2. Creation of a Thread requirement to SUNMS is an SRS

```
interface Runnable{
        void run();
}
class Thread implements Runnable{ // Adapter class
    public void start(){
        1. Register the thread with ThreadScheduler
        2. All other mandatory low level activities(memory level)
        3. invoke or call run() method
    }
    public void run(){
        //job for a thread
    }
}
```

shortcuts of eclp

ctrl+shift+T => To open a defnition of any class

ctrl + o => To list all the methods of the class



Ln 131, Col 18 110%

22°C Mostly clear

Search

2_12_2022_Threads_classnotes - Notepad

File Edit View

2. Creation of a Thread requirement to SUNMS is an SRS

```
interface Runnable{
        void run();
}
class Thread implements Runnable{ // Adapter class
    public void start(){
        1. Register the thread with ThreadScheduler
        2. All other mandatory low level activities(memory level)
        3. invoke or call run() method
    }
    public void run(){
        //job for a thread
    }
}
```

shortcuts of eclipse

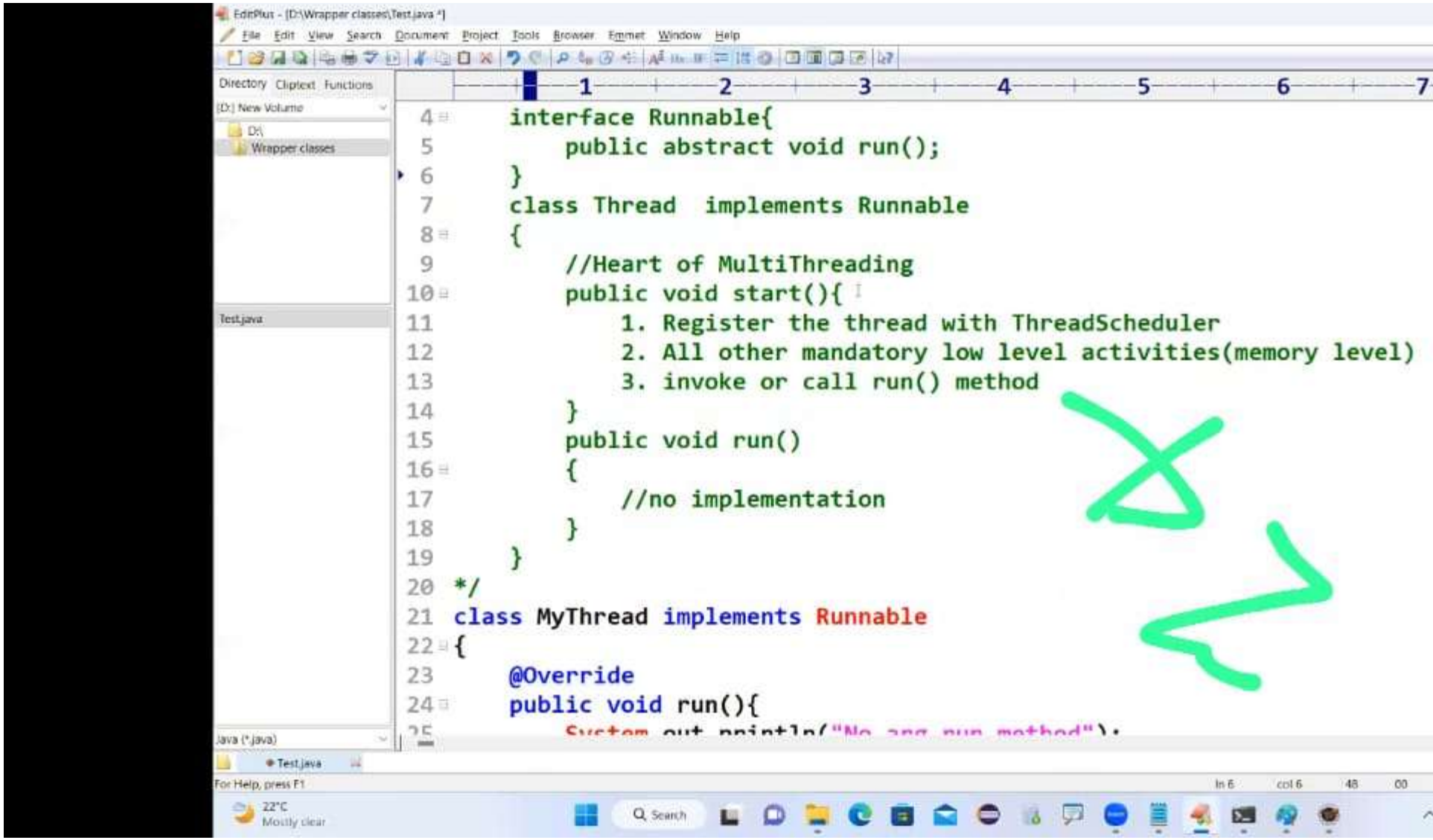
ctrl+shift+T => To open a defnition of any class

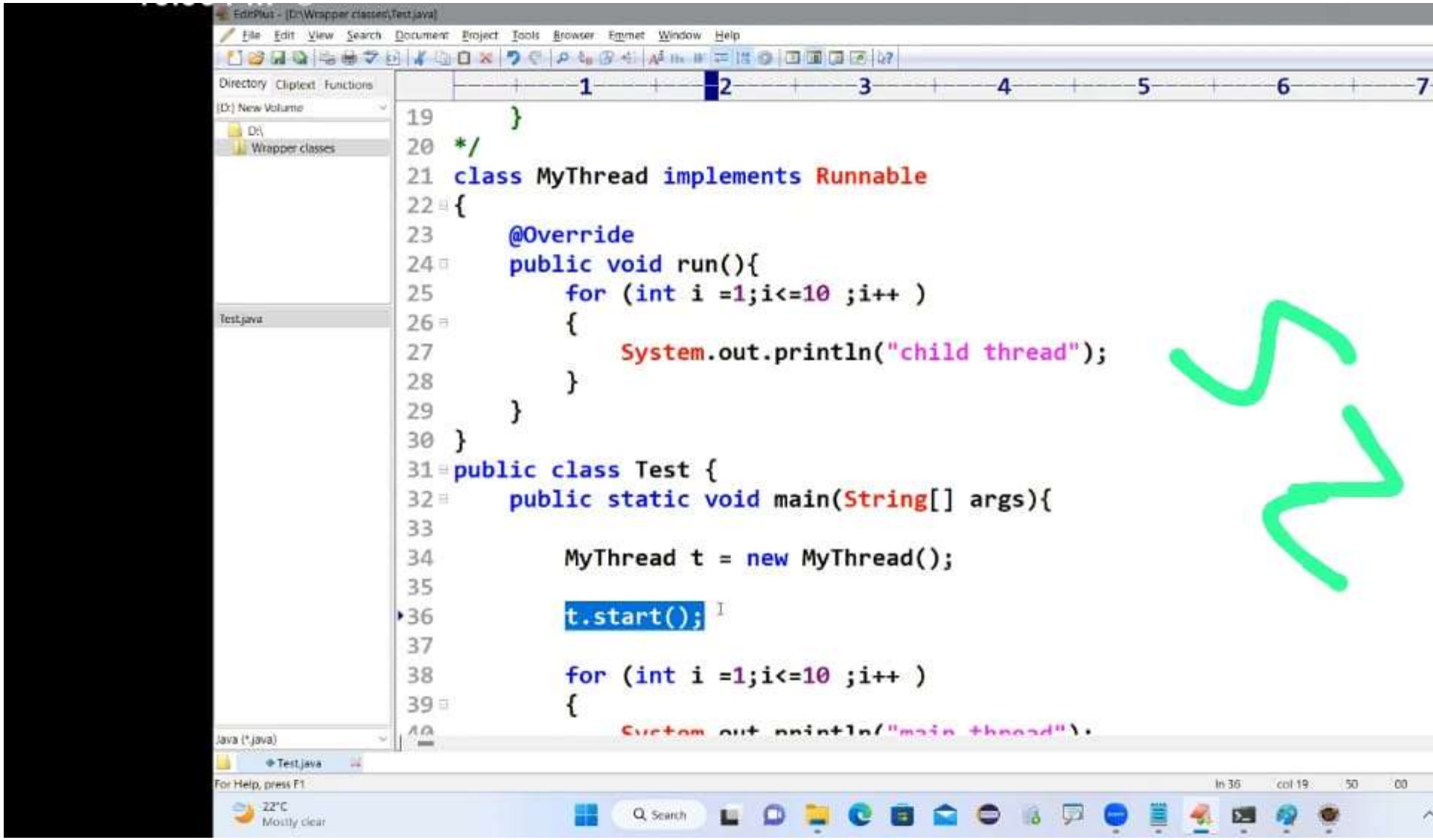
ctrl + o => To list all the methods of the class

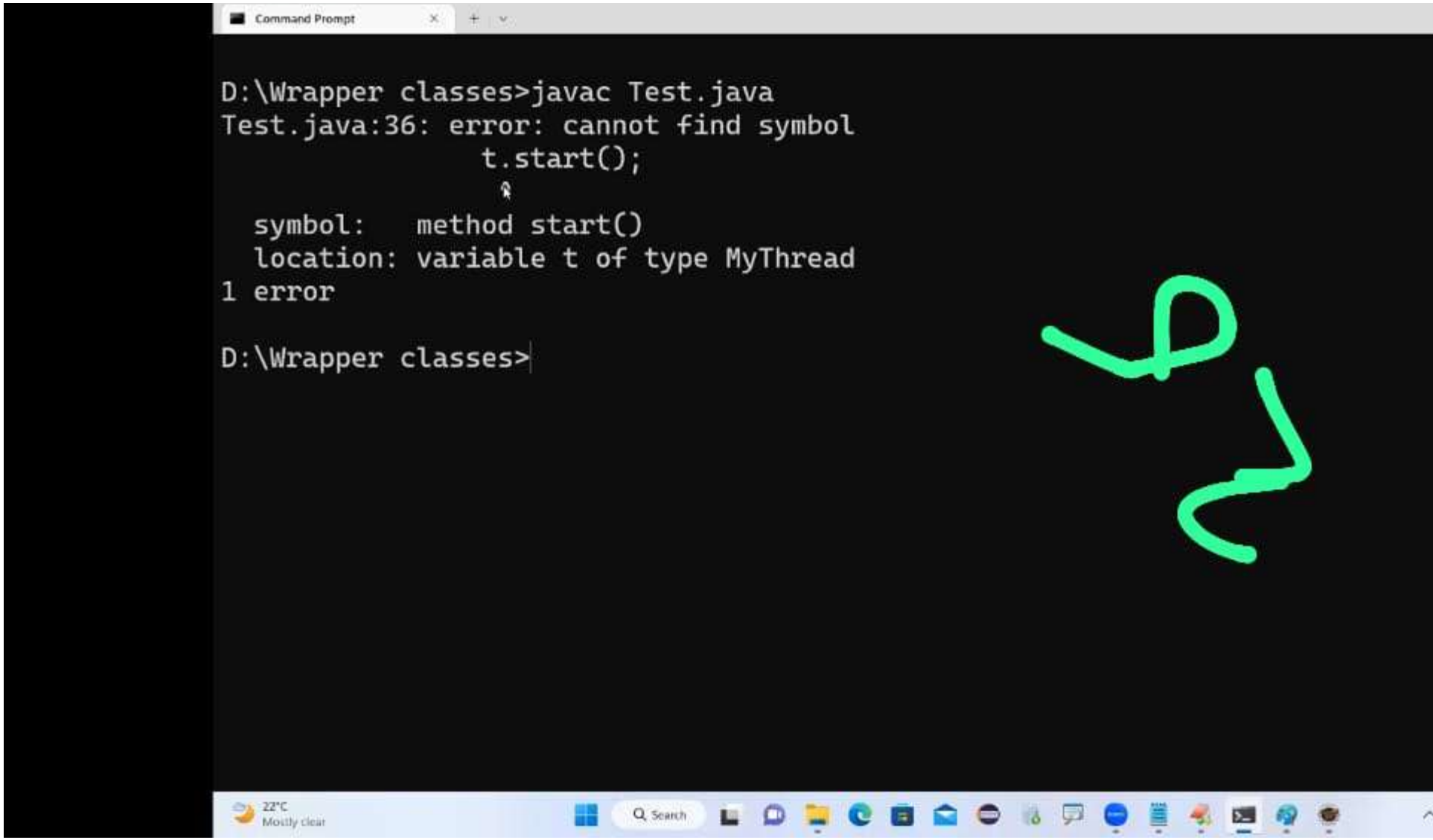
Ln 131, Col 1 110%

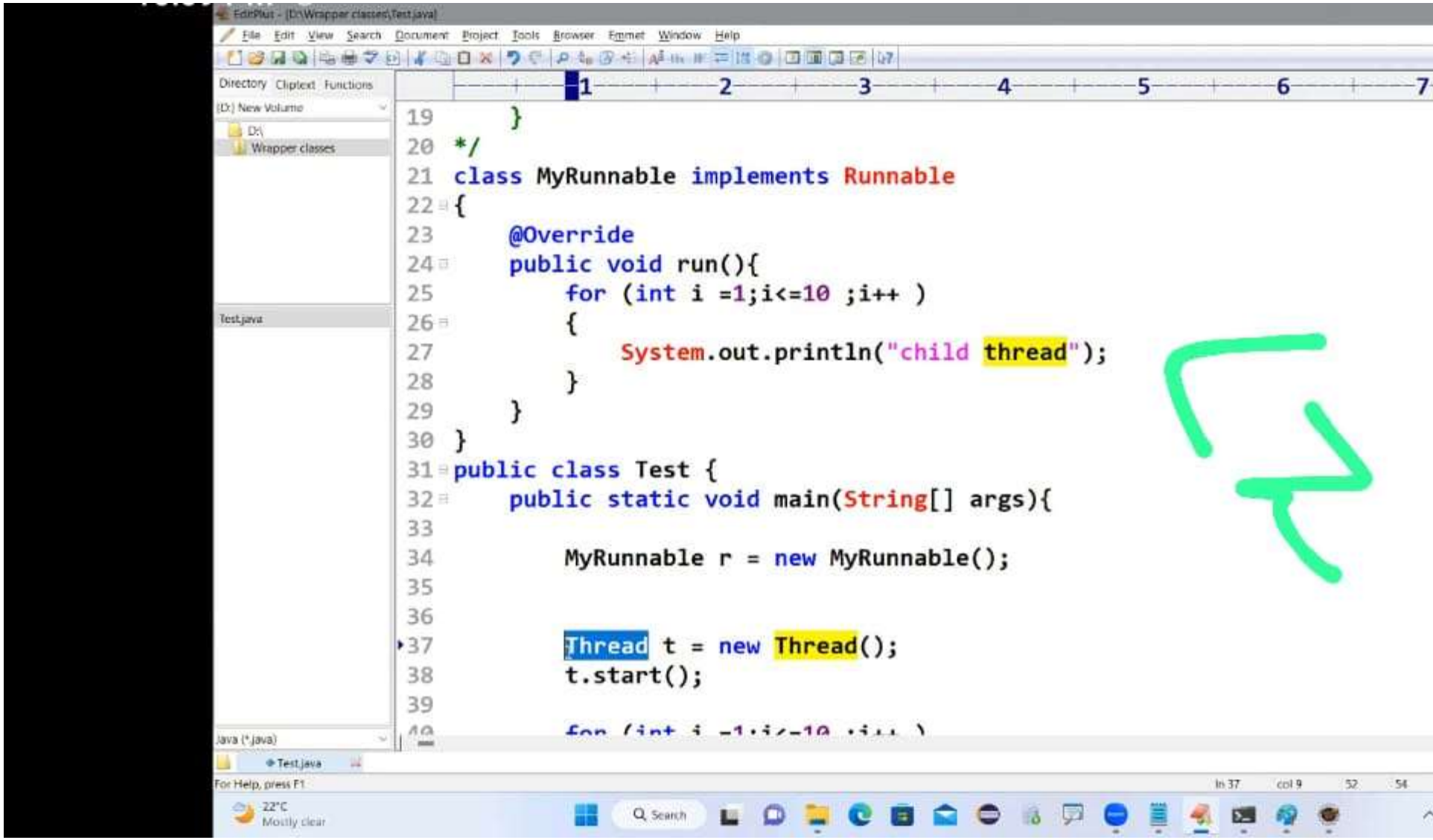
22°C Mostly clear

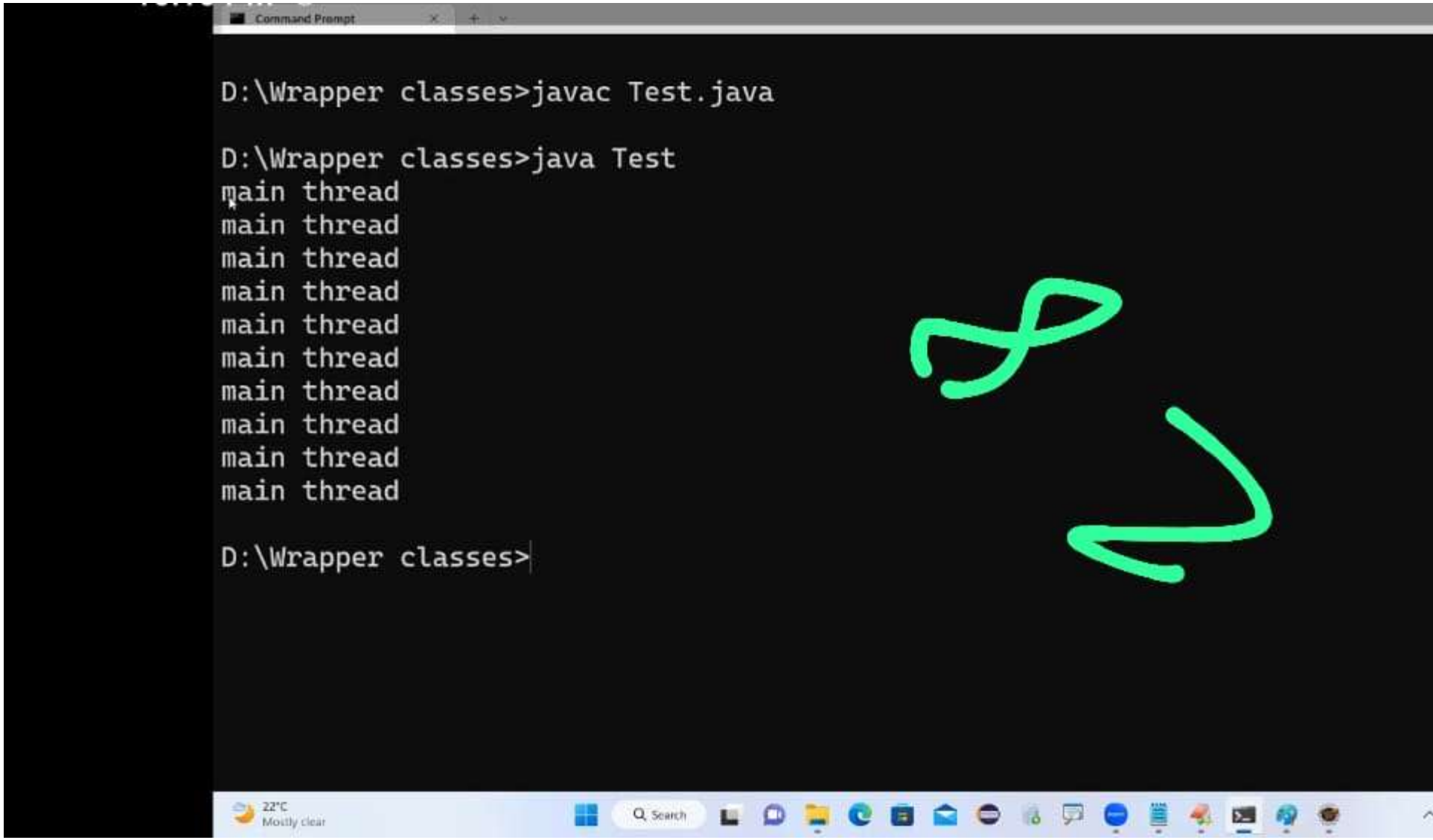
Search


EditPlus - [D:\Wrapper classes\Test.java *]
File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
[D:] New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
22°C
Mostly clear
Search
interface Runnable{
public abstract void run();
}
class Thread implements Runnable
{
//Heart of MultiThreading
public void start(){
1. Register the thread with ThreadScheduler
2. All other mandatory low level activities(memory level)
3. invoke or call run() method
}
public void run()
{
//no implementation
}
}
*/
class MyThread implements Runnable
{
@Override
public void run(){


EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
(D:) New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
Ln 36 col 19
22°C
Mostly clear
Search
19 }
20 */
21 class MyThread implements Runnable
22 {
23 @Override
24 public void run(){
25 for (int i =1;i<=10 ;i++)
26 {
27 System.out.println("child thread");
28 }
29 }
30 }
31 public class Test {
32 public static void main(String[] args){
33
34 MyThread t = new MyThread();
35
36 t.start();
37
38 for (int i =1;i<=10 ;i++)
39 {


Command Prompt
D:\Wrapper classes>javac Test.java
Test.java:36: error: cannot find symbol
t.start();
symbol: method start()
location: variable t of type MyThread
1 error
D:\Wrapper classes>
22°C
Mostly clear
Search


EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
(D:) New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
Ln 37 col 9 52 54
22°C
Mostly clear
Search
19 }
20 */
21 class MyRunnable implements Runnable
22 {
23 @Override
24 public void run(){
25 for (int i =1;i<=10 ;i++)
26 {
27 System.out.println("child thread");
28 }
29 }
30 }
31 public class Test {
32 public static void main(String[] args){
33
34 MyRunnable r = new MyRunnable();
35
36
37 Thread t = new Thread();
38 t.start();
39


Command Prompt
D:\Wrapper classes>javac Test.java
D:\Wrapper classes>java Test
main thread
main thread
main thread
main thread
main thread
main thread
main thread
main thread
main thread
main thread
D:\Wrapper classes>
22°C
Mostly clear
Search

2_12_2022_Threads_classnotes - Notepad

File Edit View

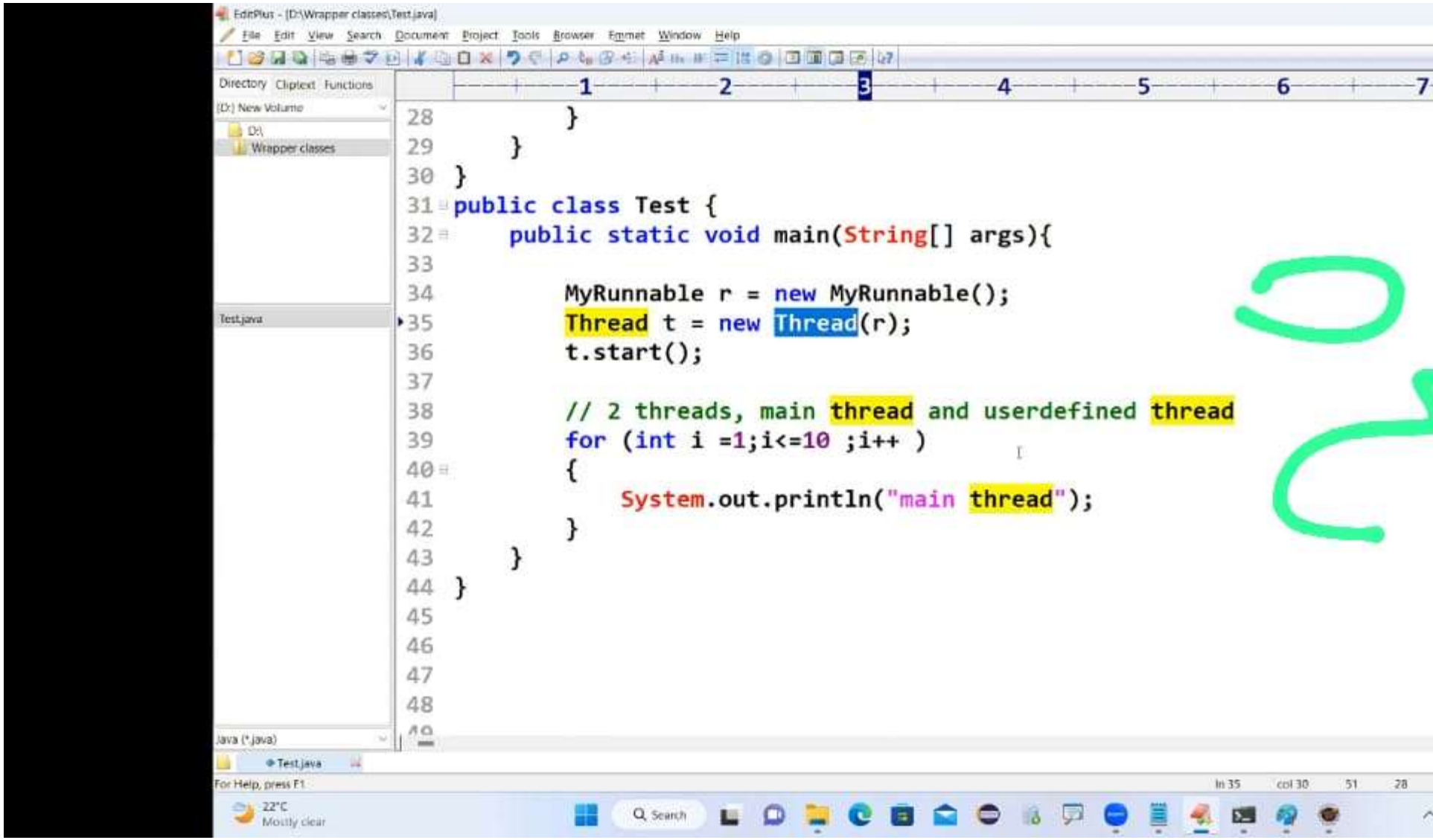
```
                1. Register the thread with ThreadScheduler
                2. All other mandatory low level activities(memory level)
                3. invoke or call run() method
            }
            public void run(){
                //job for a thread
            }
        }
```

shortcuts of eclipse
ctrl+shift+T => To open a defnition of any class
ctrl + o => To list all the methods of the class

```
public java.lang.Thread();
                    |=> thread class start(), followed by thread class run()
public java.lang.Thread(java.lang.Runnable);
                    |=> thread class start(), followed by implementation class of Runnable run()
```

Ln 139, Col 82 110%

22°C Mostly clear Search


EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Format Window Help
Directory Cliptext Functions
(D:) New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
Ln 35 col 30 51 28
22°C
Mostly clear
Search
}
}
}
public class Test {
public static void main(String[] args){
MyRunnable r = new MyRunnable();
Thread t = new Thread(r);
t.start();
// 2 threads, main thread and userdefined thread
for (int i =1;i<=10 ;i++)
{
System.out.println("main thread");
}
}
}

*2_12_2022_Threads_classnotes - Notepad

File Edit View

Case study

=========

```
MyRunnable r=new MyRunnable();
Thread t1=new Thread();
Thread t2=new Thread(r);
```

case1: t1.start()

A new thread will be created,which is responsible for executing Thread class run()

case2: t2.start()

A new thread will be created,which is responsible for executing MyRunnable run()

mainthread

main thread

main thread

main thread

main thread

userdefinedthread

Ln 204, Col 2 110%

22°C Mostly clear

Search

*2_12_2022_Threads_classnotes - Notepad

File Edit View

A new thread will be created,which is responsible for executing Thread class run()

case2: t2.start()

A new thread will be created,which is responsible for executing MyRunnable run()

output

mainthread

main thread
main thread
main thread
main thread
main thread

userdefinedthread

child thread
child thread
child thread
child thread
child thread

Ln 205, Col 14 110%

22°C Mostly clear

Search

2_12_2022_Threads_classnotes - Notepad

File Edit View

child thread
child thread
child thread

case3: t1.run()

No new thread will be created,but Thread class run() will be executed just like normal metho

output

mainthread

main thread
main thread
main thread
main thread
main thread

case4: t2.run()

Ln 223, Col 103 110%

22°C Mostly clear Search

*2_12_2022_Threads_classnotes - Notepad

File Edit View

case4: t2.run()

No new thread will be created,but MyRunnable class run() will be executed just like normal meth

output

mainthread

child thread

child thread

child thread

child thread

child thread

main thread

main thread

main thread

main thread

main thread

case5: r.s

Ln 248, Col 11 110%

22°C Mostly clear

Search

2_12_2022_Threads_classnotes - Notepad

File Edit View

case5: r.start()

It results in CompileTime Error

case6. r.run()

No new thread will be created,but MyRunnable class run() will be executed just like normal meth

output

mainthread

child thread

child thread

child thread

child thread

child thread

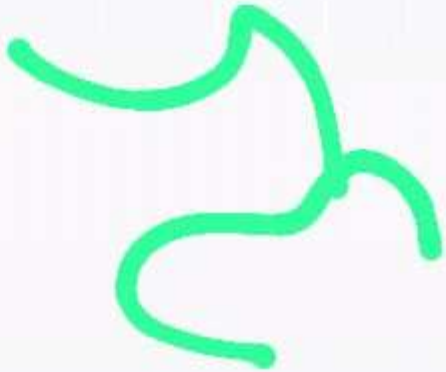
main thread

main thread

main thread

main thread

main thread



Ln 265, Col 13 110%

22°C Mostly clear

Search

*2_12_2022_Threads_classnotes - Notepad

File Edit View

case6: r.run()

In which of the above cases a new Thread will be created which is responsible for the execution of MyRun

```
t2.start();
```

In which of the above cases a new Thread will be created ?

```
t1.start();
t2.start();
```

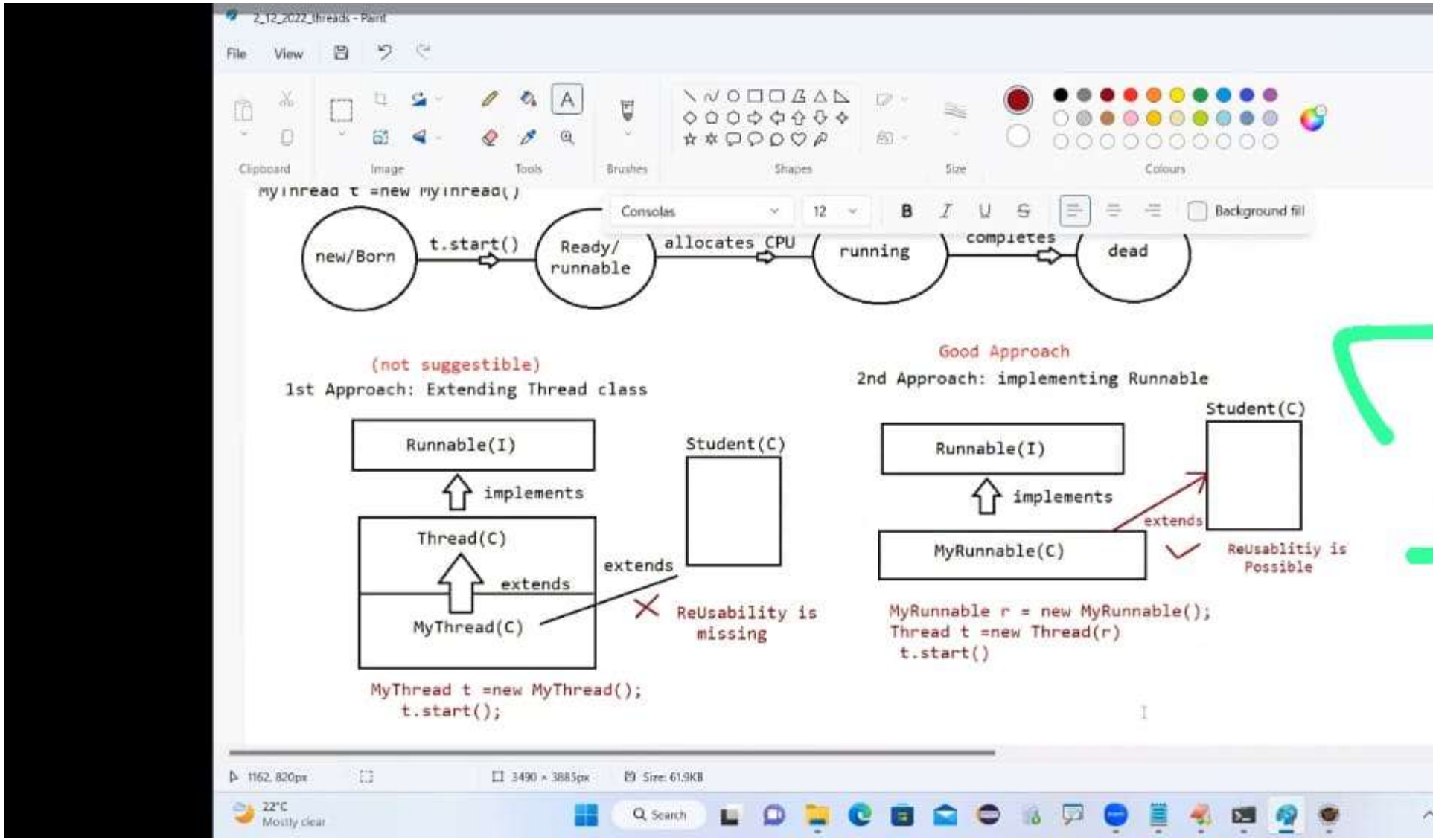
In which of the above cases MyRunnable class run() will be executed?

```
t2.start();
t2.run();
r.run();
```

Ln 29, Col 1 110%

22°C Mostly clear

Search


2_12_2022_threads - Paint
File
View
Clipboard
Image
Tools
Brushes
Shapes
Size
Colours
MyThread t =new MyThread()
Consolas
12
Background fill
new/Born
t.start()
Ready/
runnable
allocates CPU
running
completes
dead
(not suggestible)
1st Approach: Extending Thread class
Runnable(I)
implements
Thread(C)
extends
MyThread(C)
Student(C)
extends
ReUsability is
missing
MyThread t =new MyThread();
t.start();
Good Approach
2nd Approach: implementing Runnable
Runnable(I)
implements
MyRunnable(C)
Student(C)
extends
ReUsablitiy is
Possible
MyRunnable r = new MyRunnable();
Thread t =new Thread(r)
t.start()
1162, 820px
3490 × 3885px
Size: 61.9KB
22°C
Mostly clear
Search

2_12_2022_Threads_classnotes - Notepad

File Edit View

```
t2.run();
r.run();
```

Different approach for creating a Thread?

A. extending Thread class

B. implementing Runnable interface

Which approach is the best approach?

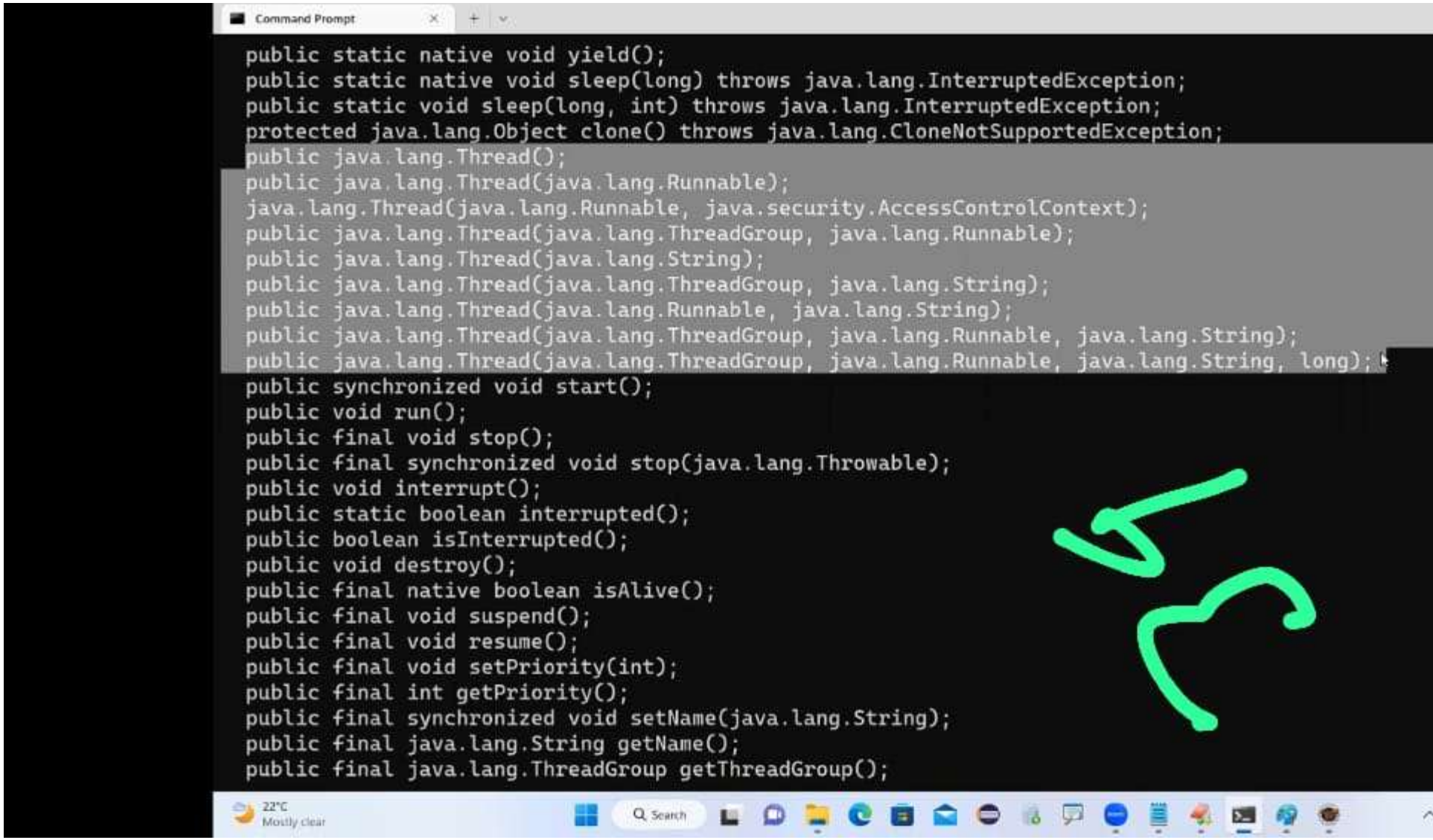
a. implements Runnable interface is recomended becoz our class can extend other class through which inheritance benift can brought in to our class.
Internally performance and memory level is also good when we work with interface.

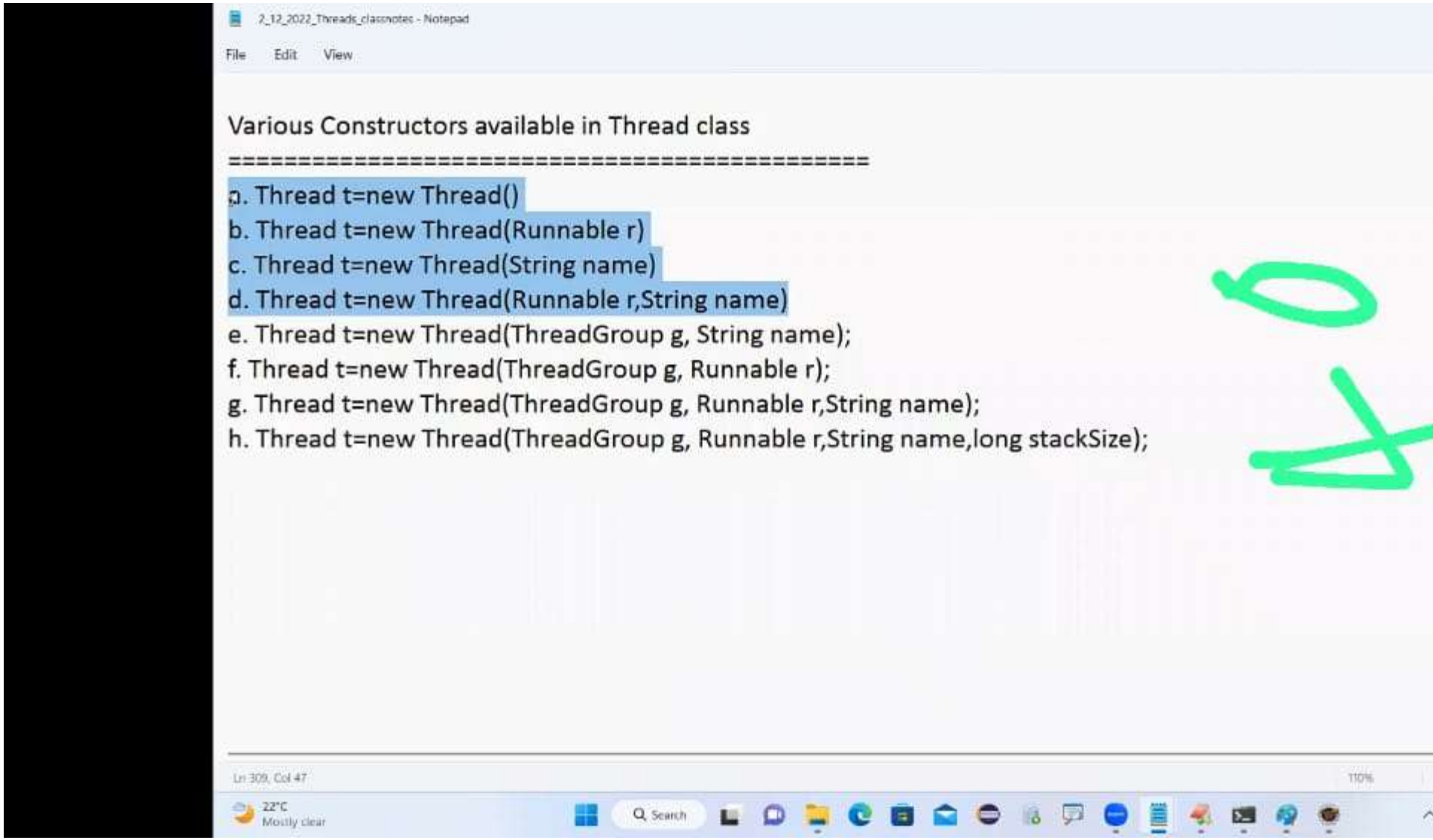
b. if we work with extends feature then we will miss out inheritance benefit becoz already our class has inherited the feature from "Thread class", so we normally we don't prefere extends approach rather implements approach is used in real time for working with "MultiThreading".

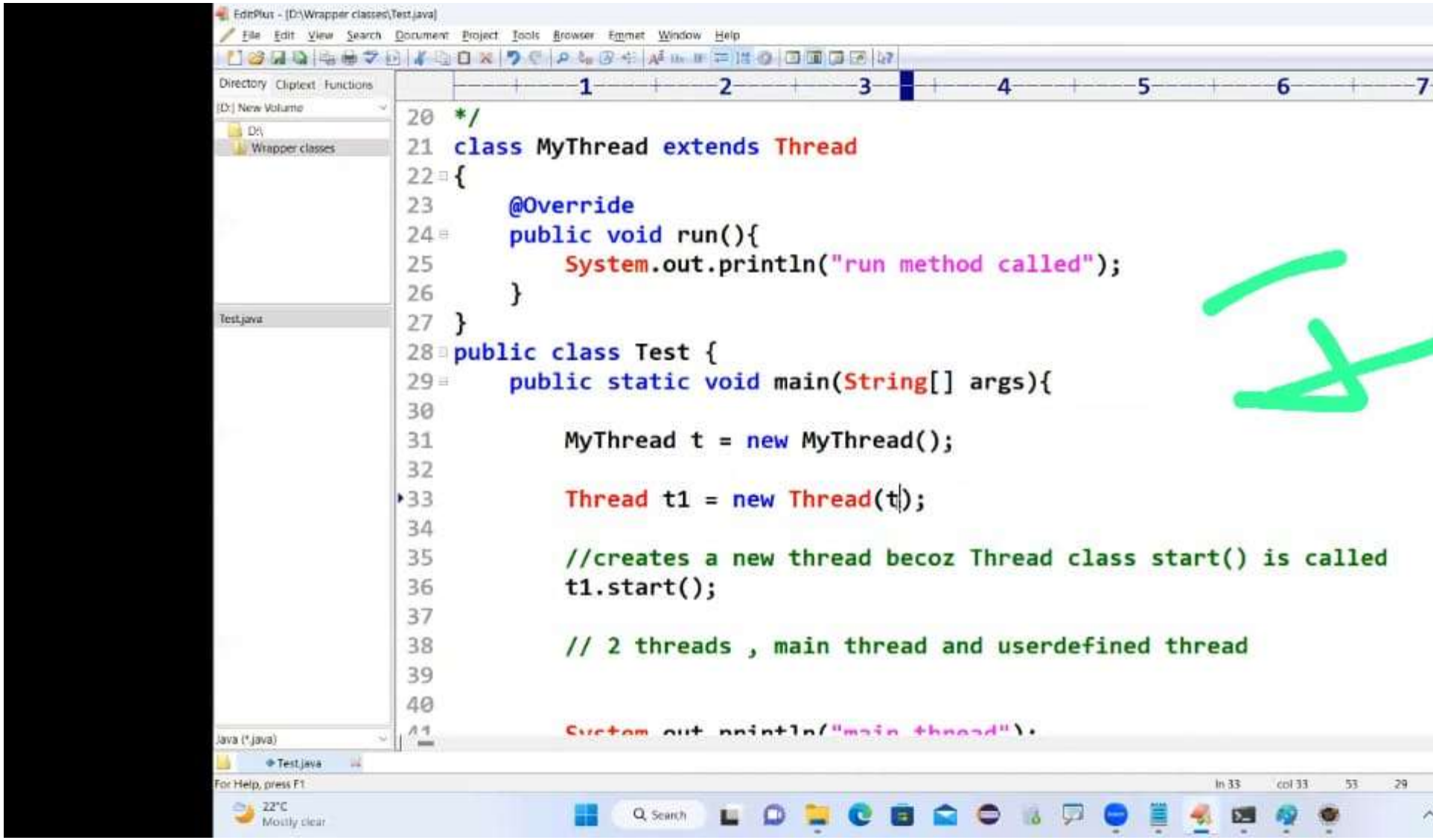
Ln 306, Col 1 110%

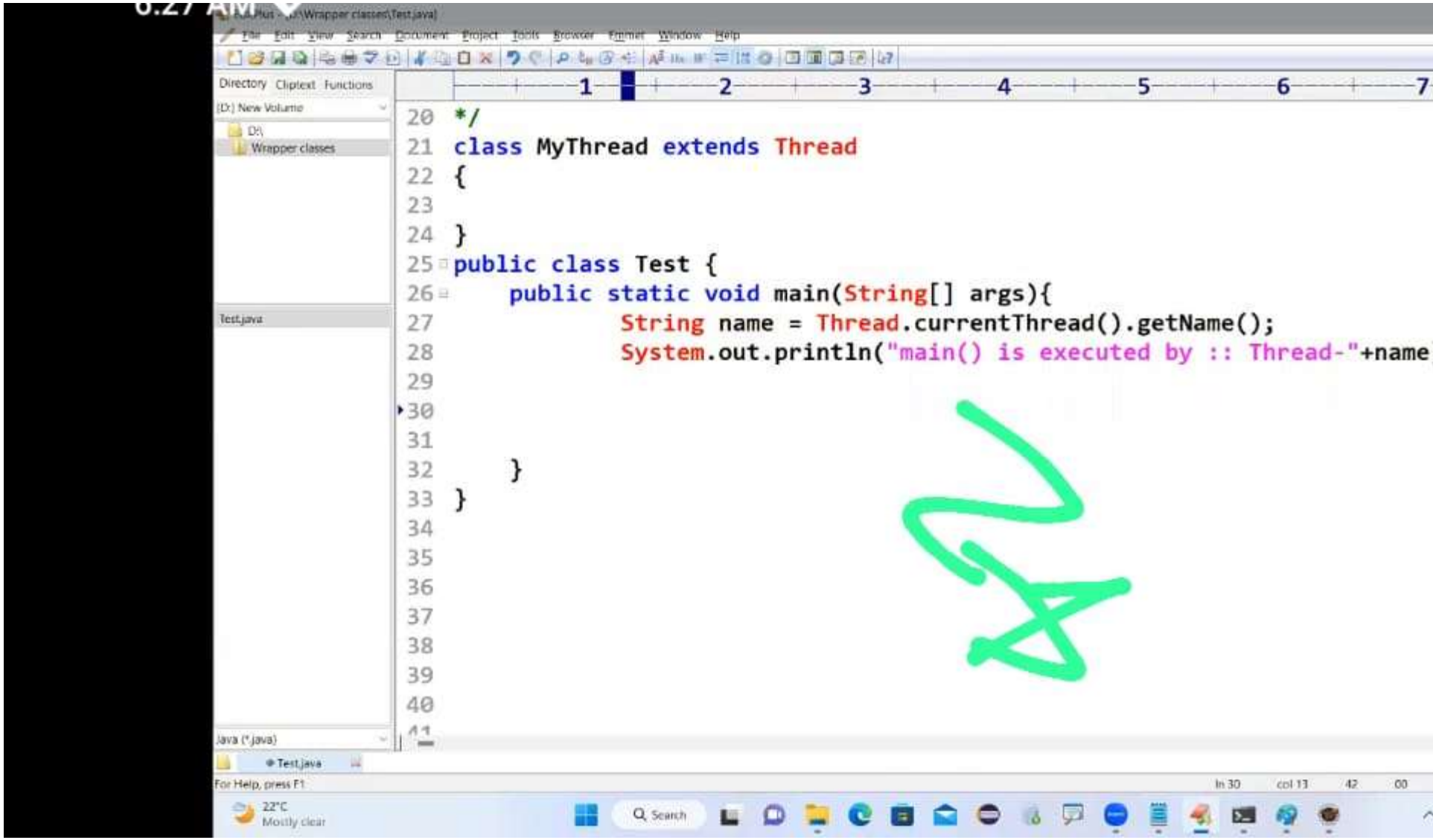
22°C Mostly clear

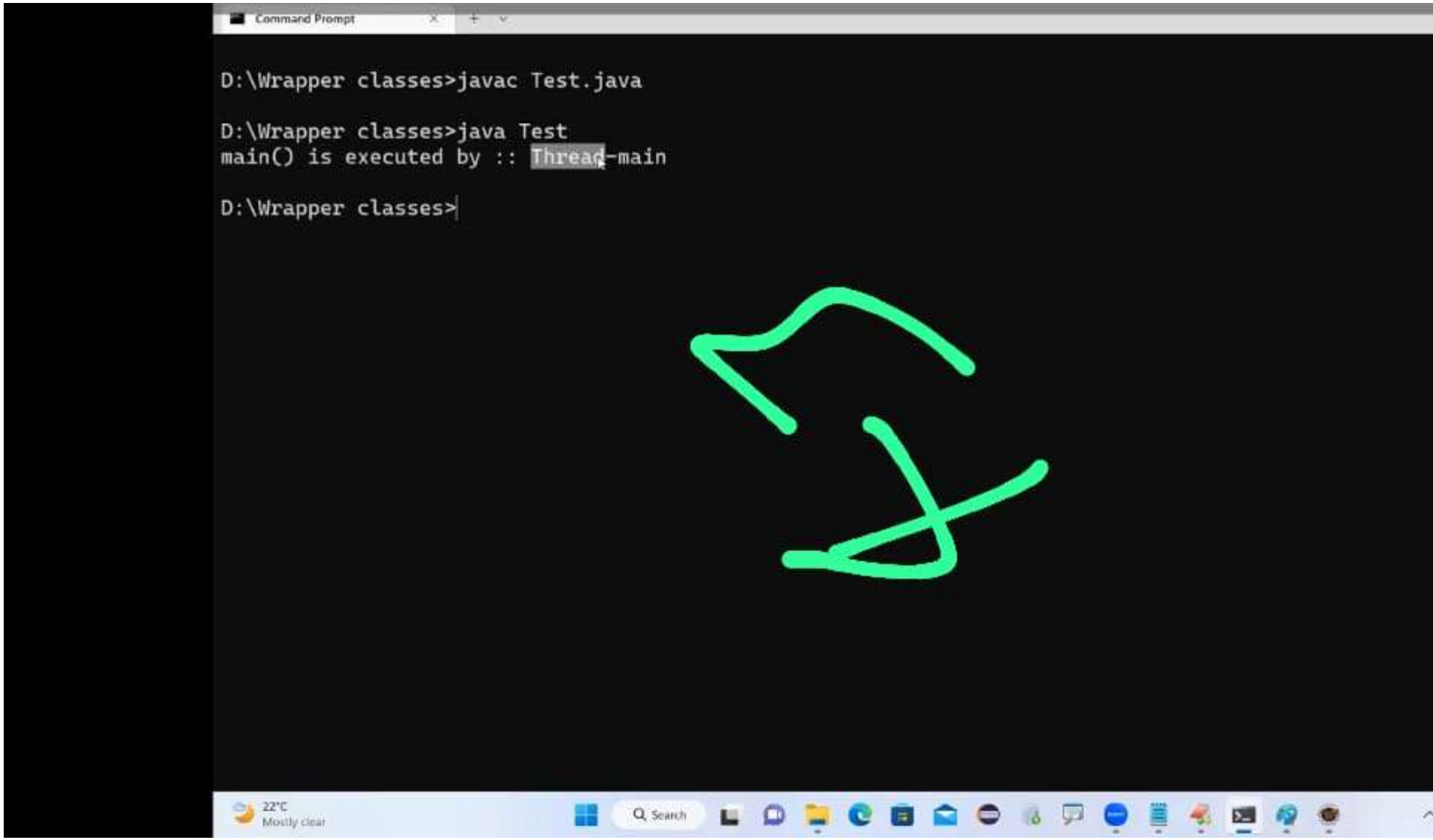
Search

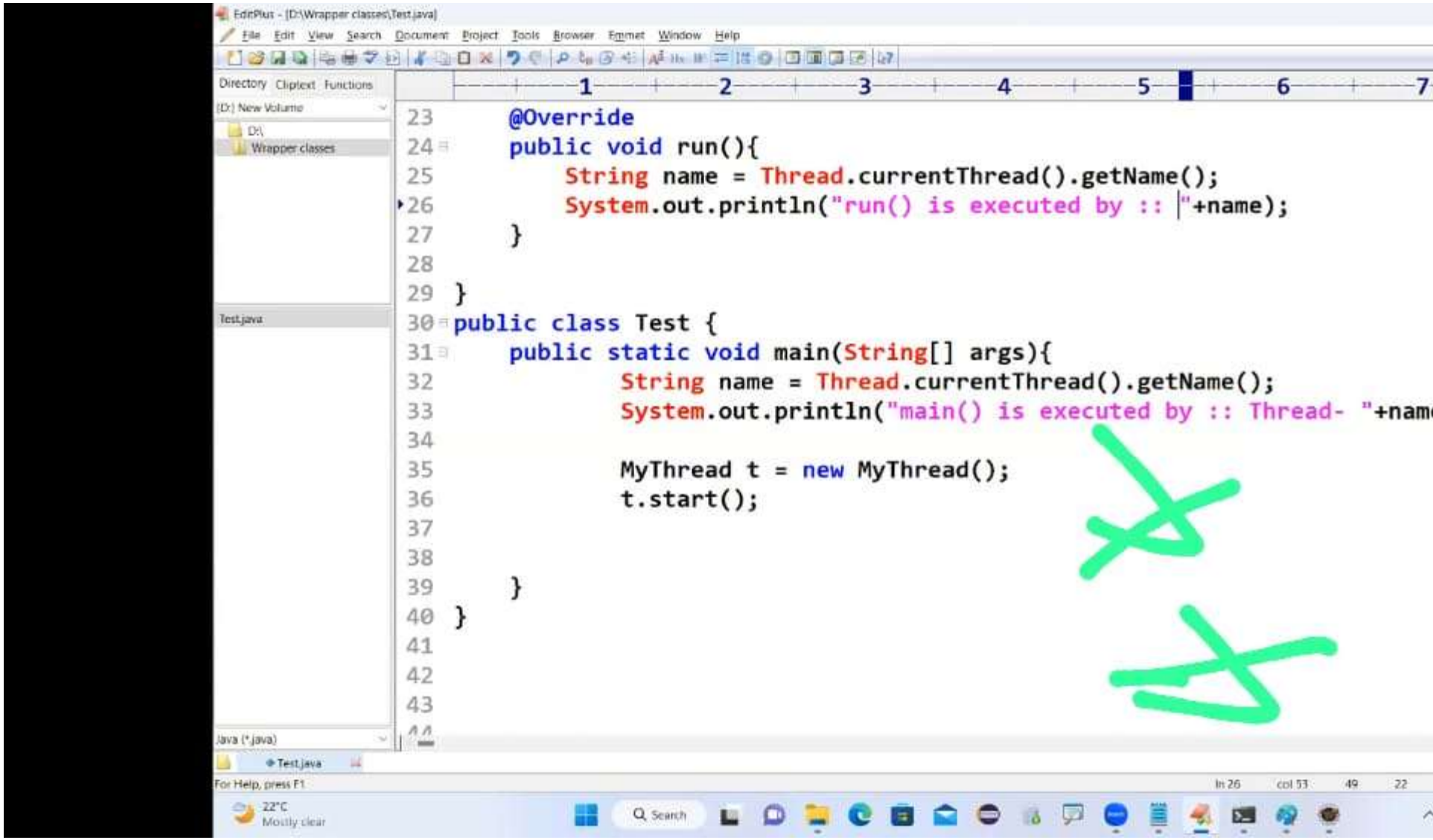

Command Prompt
public static native void yield();
public static native void sleep(long) throws java.lang.InterruptedException;
public static void sleep(long, int) throws java.lang.InterruptedException;
protected java.lang.Object clone() throws java.lang.CloneNotSupportedException;
public java.lang.Thread();
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);
public synchronized void start();
public void run();
public final void stop();
public final synchronized void stop(java.lang.Throwable);
public void interrupt();
public static boolean interrupted();
public boolean isInterrupted();
public void destroy();
public final native boolean isAlive();
public final void suspend();
public final void resume();
public final void setPriority(int);
public final int getPriority();
public final synchronized void setName(java.lang.String);
public final java.lang.String getName();
public final java.lang.ThreadGroup getThreadGroup();
22°C
Mostly clear
Search

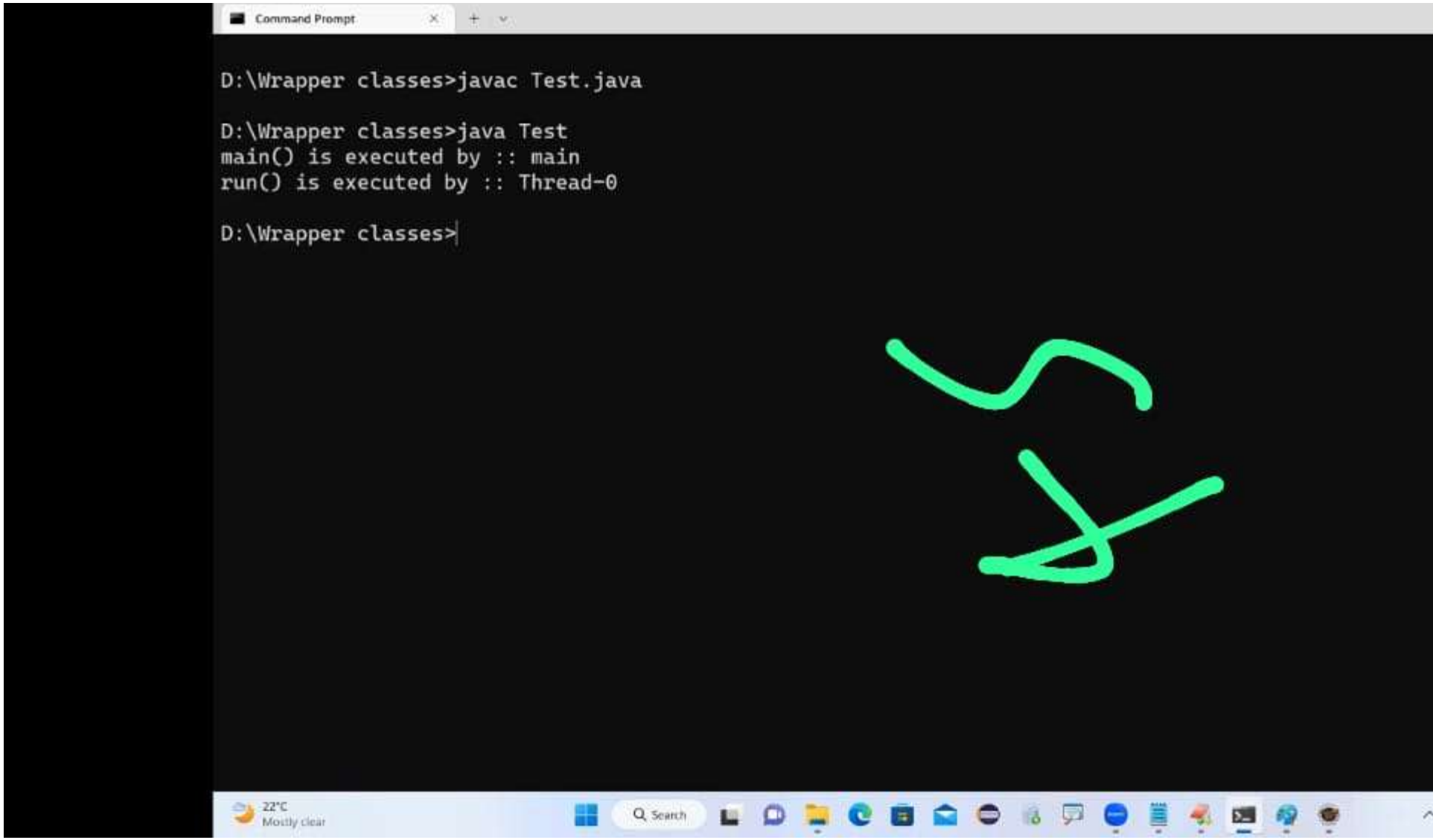

2_12_2022_Threads_classnotes - Notepad
File Edit View
Various Constructors available in Thread class
==========
a. Thread t=new Thread()
b. Thread t=new Thread(Runnable r)
c. Thread t=new Thread(String name)
d. Thread t=new Thread(Runnable r,String name)
e. Thread t=new Thread(ThreadGroup g, String name);
f. Thread t=new Thread(ThreadGroup g, Runnable r);
g. Thread t=new Thread(ThreadGroup g, Runnable r,String name);
h. Thread t=new Thread(ThreadGroup g, Runnable r,String name,long stackSize);
Ln 309, Col 47
110%
22°C
Mostly clear
Search

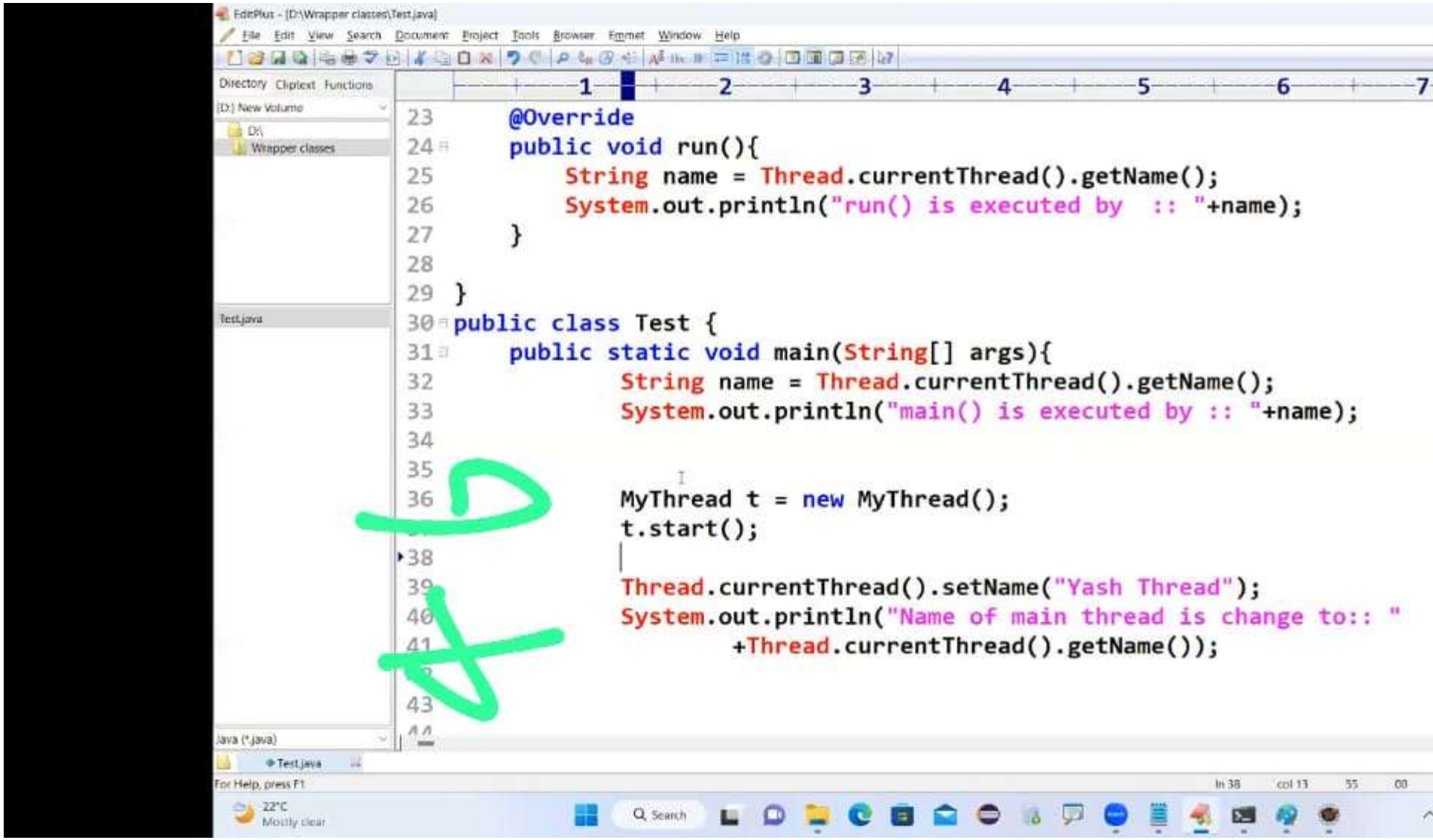

EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Format Window Help
Directory Cliptext Functions
[D:] New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
*/
class MyThread extends Thread
{
@Override
public void run(){
System.out.println("run method called");
}
}
public class Test {
public static void main(String[] args){
MyThread t = new MyThread();
Thread t1 = new Thread(t);
//creates a new thread becoz Thread class start() is called
t1.start();
// 2 threads , main thread and userdefined thread
22°C
Mostly clear
Search


*/
class MyThread extends Thread
{
}
public class Test {
public static void main(String[] args){
String name = Thread.currentThread().getName();
System.out.println("main() is executed by :: Thread-"+name
}
}


Command Prompt
D:\Wrapper classes>javac Test.java
D:\Wrapper classes>java Test
main() is executed by :: Thread-main
D:\Wrapper classes>
22°C
Mostly clear
Search


EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Format Window Help
Directory Cliptext Functions
(D:) New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
Ln 26 col 53 49 22
22°C
Mostly clear
Search
@Override
public void run(){
String name = Thread.currentThread().getName();
System.out.println("run() is executed by :: "+name);
}
}
public class Test {
public static void main(String[] args){
String name = Thread.currentThread().getName();
System.out.println("main() is executed by :: Thread- "+nam
MyThread t = new MyThread();
t.start();
}
}


Command Prompt
D:\Wrapper classes>javac Test.java
D:\Wrapper classes>java Test
main() is executed by :: main
run() is executed by :: Thread-0
D:\Wrapper classes>
22°C
Mostly clear
Search


EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
[D:] New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
@Override
public void run(){
String name = Thread.currentThread().getName();
System.out.println("run() is executed by :: "+name);
}
}
public class Test {
public static void main(String[] args){
String name = Thread.currentThread().getName();
System.out.println("main() is executed by :: "+name);
MyThread t = new MyThread();
t.start();
Thread.currentThread().setName("Yash Thread");
System.out.println("Name of main thread is change to:: "
+Thread.currentThread().getName());
22°C
Mostly clear
Search

2_12_2022_Threads_classnotes - Notepad

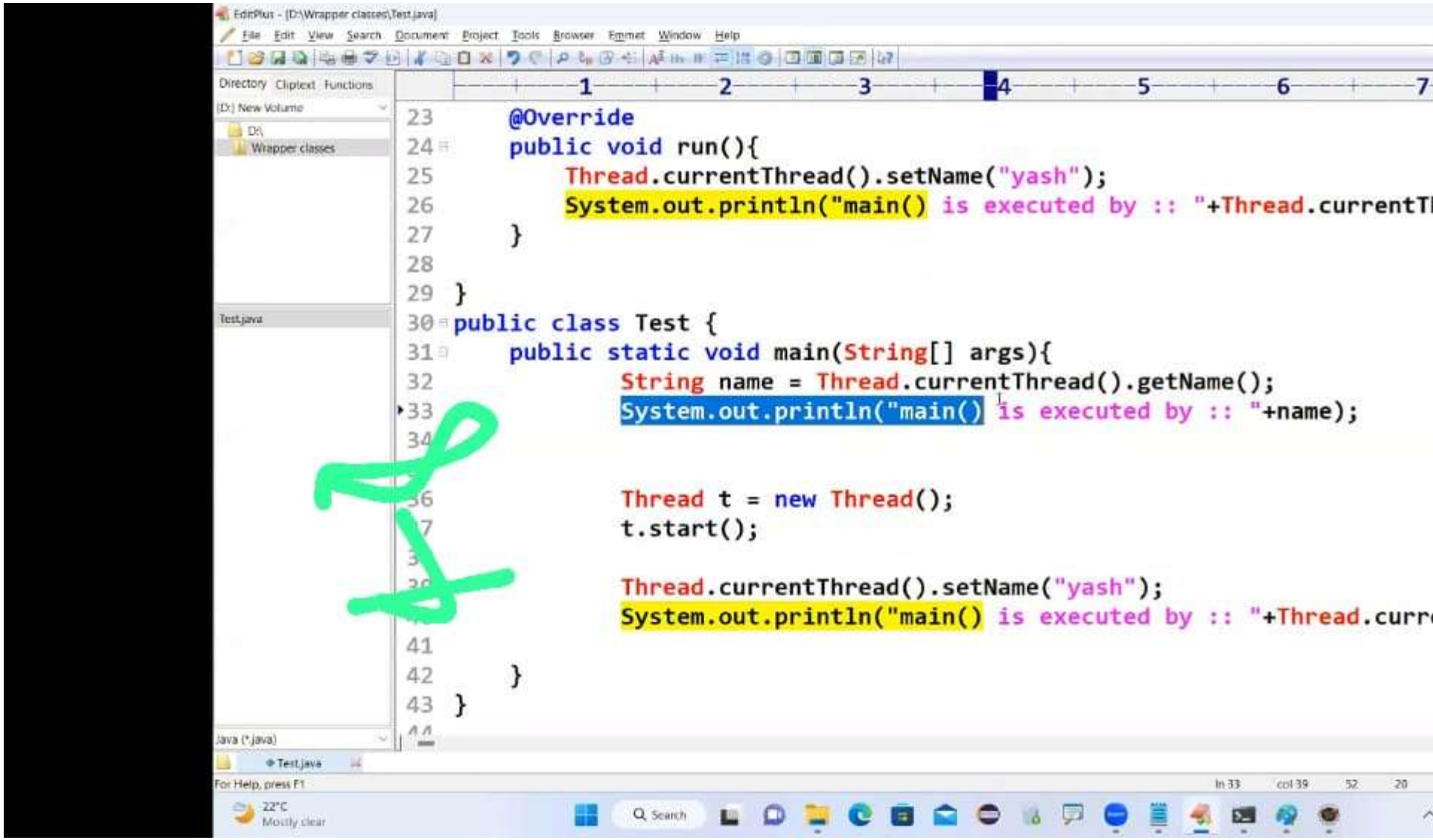
File Edit View

```
public class TestApp{
    public static void main(String... args){
        System.out.println(Thread.currentThread().getName());//main

        MyThread t=new MyThread();
        t.start();
        System.out.println(t.getName());//Thread-0

        Thread.currentThread().setName("Yash");//Yash
        System.out.println(Thread.currentThread().getName());//Yash

        System.out.println(10/0);
          //Exception in thread "yash" java.lang.ArithmeticException:/by zero
                TestApp.main()

    }
}
```

It is also possible to change the name of the Thread using setName().
It is possible to get the name of the Thread using getName().

Ln 392, Col 9 110%

22°C Mostly clear Search


EditPlus - [D:\Wrapper classes\Test.java]
File Edit View Search Document Project Tools Browser Emmet Window Help
Directory Cliptext Functions
(D:) New Volume
D:\
Wrapper classes
Test.java
Java (*.java)
Test.java
For Help, press F1
@Override
public void run(){
Thread.currentThread().setName("yash");
System.out.println("main() is executed by :: "+Thread.currentT
}
}
public class Test {
public static void main(String[] args){
String name = Thread.currentThread().getName();
System.out.println("main() is executed by :: "+name);
Thread t = new Thread();
t.start();
Thread.currentThread().setName("yash");
System.out.println("main() is executed by :: "+Thread.curr
}
}
22°C
Mostly clear
Search

2_12_2022_snippets_session_notes - Notepad

File Edit View

```
public class Breaker {
 static String o = "";
 public static void main(String[] args) {
       z: o = o + 2;
      for (int x = 3; x < 8; x++) {
            if (x == 4)
                  break;
            if (x == 6)
                  break z;
            o = o + x;
      }
      System.out.println(o);
 }
}
```

What is the result?

A. 23
B. 234
C. 235
D. 2345
E. 2357
F. 23457
G. Compilation fails.

Ln 23, Col 22 100%

22°C Mostly clear

Search

*2_12_2022_snippets_session_notes - Notepad

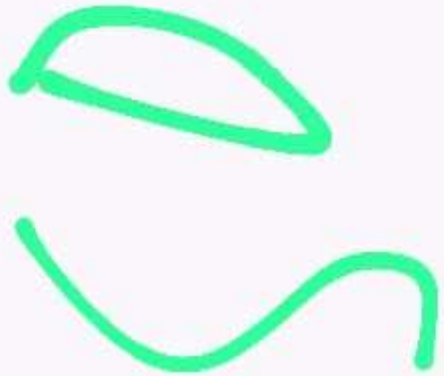
File Edit View

```
            if (x == 6)
                break z;
            o = o + x;
        }
        System.out.println(o);
 }
}
```

What is the result?

A. 23
B. 234
C. 235
D. 2345
E. 2357
F. 23457
G. Compilation fails.

Answer: G



Ln 30, Col 1 100%

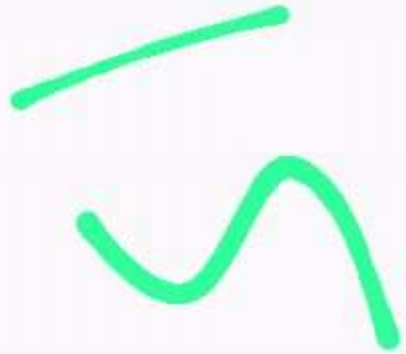
22°C Mostly clear

Search

2_12_2022_snippets_session_notes - Notepad

File Edit View

```
public class Tahiti {
    Tahiti t;
    public static void main(String[] args) {
        Tahiti t = new Tahiti();
        Tahiti t2 = t.go(t);
        t2 = null;
        // more code here 11
    }
    Tahiti go(Tahiti t) {
        Tahiti t1 = new Tahiti();
        Tahiti t2 = new Tahiti();

        t1.t = t2;
        t2.t = t1;
        t.t = t2;
        return t1;
    }
}
```



When line 11 is reached, how many objects are eligible for garbage collection?

A. 0

B. 1

C. 2

Ln 47, Col 5 100%

22°C Mostly clear

Search

2_12_2022_snippets_session_notes - Notepad

File Edit View

```
        t2 = null;
        // more code here 11
    }
    Tahiti go(Tahiti t) {
        Tahiti t1 = new Tahiti();
        Tahiti t2 = new Tahiti();

        t1.t = t2;
        t2.t = t1;
        t.t = t2;
        return t1;
    }
}
```

When line 11 is reached, how many objects are eligible for garbage collection?

A. 0

B. 1

C. 2

D. 3

Answer: 0(island of isolation)

Ln 54, Col 1 100%

21°C Mostly clear

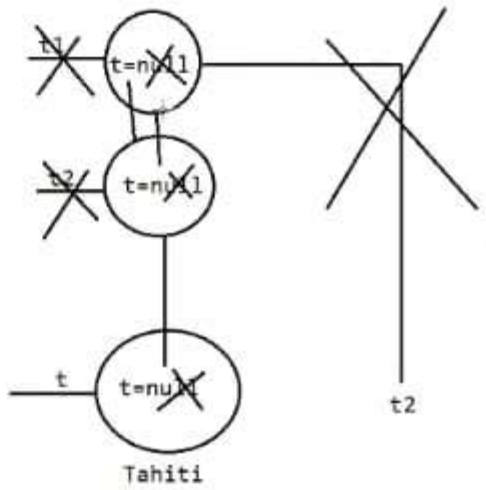
Search

Untitled - Paint

File View

Clipboard Image Tools Brushes Shapes Size Colours

```
public class Tahiti {

      Tahiti t;

     Tahiti go(Tahiti t) {

          Tahiti t1 = new Tahiti();
          Tahiti t2 = new Tahiti();

          t1.t = t2;
          t2.t = t1;

          t.t = t2;

          return t1;

     }
     public static void main(String[] args) {
          Tahiti t = new Tahiti();
          Tahiti t2 = t.go(t);
          t2 = null;
          // more code here 11
      }

}
```



1009, 154px 3490 × 3885px

21°C Mostly clear

Search

2_12_2022_snippets_session_notes - Notepad

File Edit View

Q>

```
public class ItemTest {
    private final int id;
    public ItemTest(int id) {
        this.id = id;
    }
    public void updateId(int newId) {
        id = newId;
    }
    public static void main(String[] args) {
        ItemTest fa = new ItemTest(42);
        fa.updateId(69);
        System.out.println(fa.id);
    }
}
```

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The attribute id in the ItemTest object remains unchanged.

D. The attribute id in the ItemTest object is modified to the new value.

E. A new ItemTest object is created with the preferred value in the id attribute.

Ln 74, Col 62 100%

21°C Mostly clear

Search

2_12_2022_snippets_session_notes - Notepad

File Edit View

```
    }
    public void updateId(int newId) {
        id = newId;
    }
    public static void main(String[] args) {
            ItemTest fa = new ItemTest(42);
            fa.updateId(69);
            System.out.println(fa.id);
    }
}
```

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The attribute id in the ItemTest object remains unchanged.

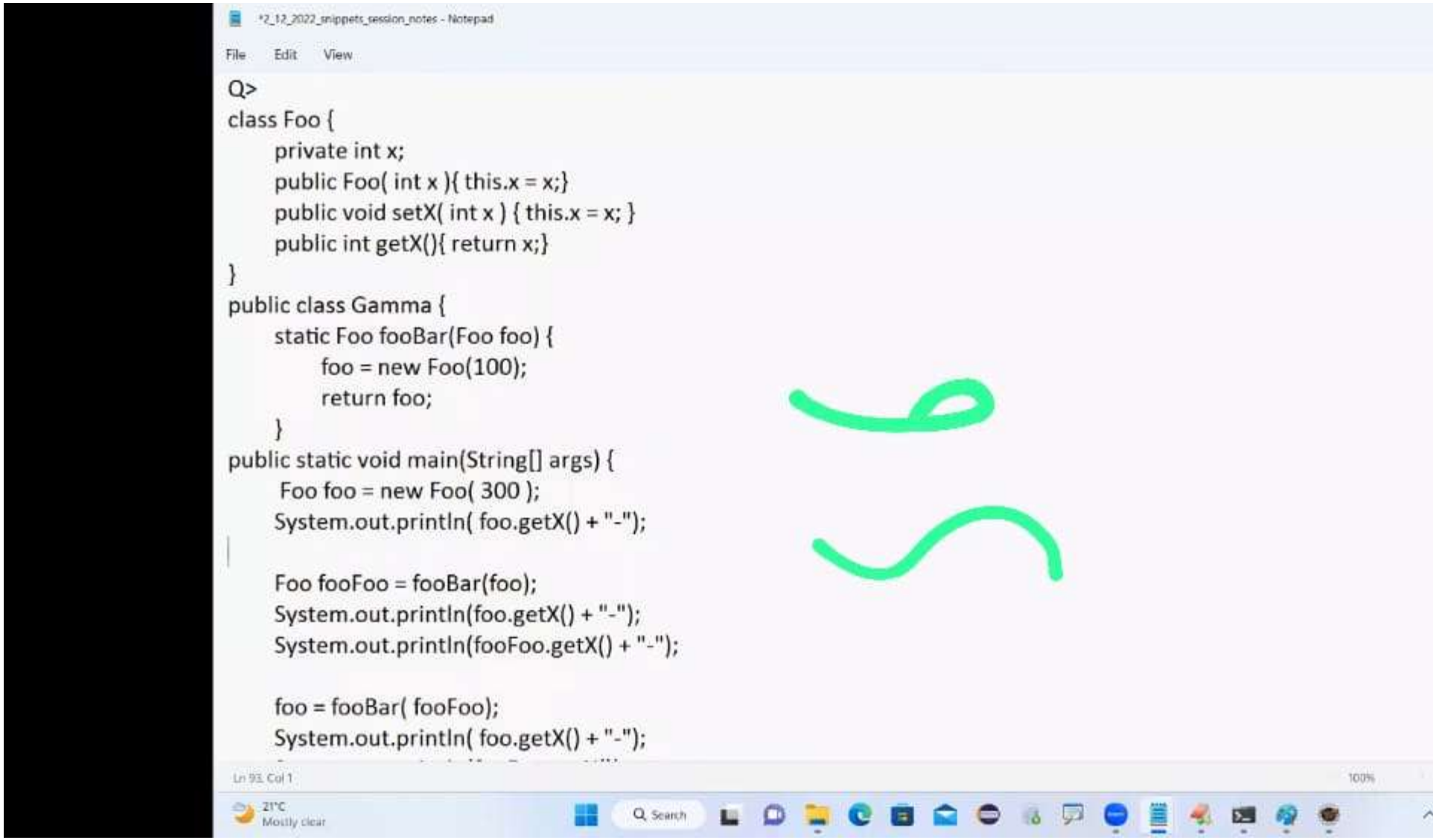
D. The attribute id in the ItemTest object is modified to the new value.

E. A new ItemTest object is created with the preferred value in the id attribute.

Answer: A

Ln 78, Col 1 100%

21°C Mostly clear Search


*2_12_2022_snippets_session_notes - Notepad
File Edit View
Q>
class Foo {
private int x;
public Foo(int x){ this.x = x;}
public void setX(int x) { this.x = x; }
public int getX(){ return x;}
}
public class Gamma {
static Foo fooBar(Foo foo) {
foo = new Foo(100);
return foo;
}
public static void main(String[] args) {
Foo foo = new Foo(300);
System.out.println(foo.getX() + "-");
Foo fooFoo = fooBar(foo);
System.out.println(foo.getX() + "-");
System.out.println(fooFoo.getX() + "-");
foo = fooBar(fooFoo);
System.out.println(foo.getX() + "-");
Ln 93, Col 1
100%
21°C
Mostly clear
Search

2_12_2022_snippets_session_notes - Notepad

File Edit View

```
        System.out.println( foo.getX() + "-");
        Foo fooFoo = fooBar(foo);
        System.out.println(foo.getX() + "-");
        System.out.println(fooFoo.getX() + "-");
        foo = fooBar( fooFoo);
        System.out.println( foo.getX() + "-");
        System.out.println(fooFoo.getX());
    }
}
```

What is the output of the program shown in the exhibit?

A. 300-100-100-100-100

B. 300-300-100-100-100

C. 300-300-300-100-100

D. 300-300-300-300-100

Answer: B

Q>

Ln 111, Col 1 100%

21°C Mostly clear

Search

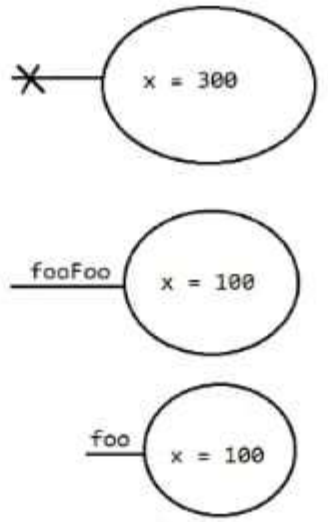
2_12_2022_snippets_images - Paint

File View

Clipboard Image Tools Brushes Shapes Size Colours

```
class Foo {
     private int x;
     public Foo( int x ){ this.x = x;}
     public void setX( int x ) { this.x = x; }
     public int getX(){ return x;}
}

public class Gamma {
     static Foo fooBar(Foo foo) {
          foo = new Foo(100);
          return foo;
     }
     public static void main(String[] args) {
           Foo foo = new Foo( 300 );
          System.out.println( foo.getX() + "-");
          Foo fooFoo = fooBar(foo);
          System.out.println(foo.getX() + "-");
          System.out.println(fooFoo.getX() + "-");
          foo = fooBar( fooFoo);
          System.out.println( foo.getX() + "-");
          System.out.println(fooFoo.getX());
     }
}
```

300-300-100 -100-100



1087, 809px 276 × 41px 3490 × 3885px Size: 72.1KB

21°C Mostly clear

Search

2_12_2022_snippets_session_notes - Notepad

File Edit View

Q>

```
interface Fish {}
class Perch implements Fish {}
class Walleye extends Perch {}
class Bluegill {}
public class Fisherman {
 public static void main(String[] args) {
      Fish f = new Walleye();
      Walleye w = new Walleye();
      Bluegill b = new Bluegill();
      if (f instanceof Perch)
            System.out.print("f-p ");
      if (w instanceof Fish)
            System.out.print("w-f ");
      if (b instanceof Fish)
            System.out.print("b-f ");
 }
}
```

What is the result?

A. w-f

B. f-p w-f

Ln 114, Col 17 100%

21°C Mostly clear

Search