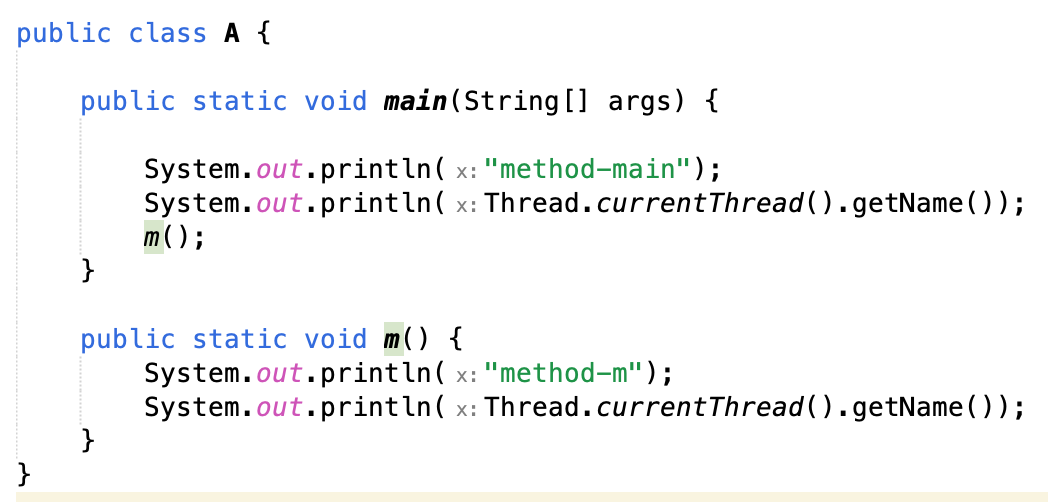
# Threads

Concurrent programming වලට තමයි අපි thread use කරන්නේ.එක වැඩක් එක thread එකකට. එකම වෙලාවේ වැඩ කීපයක් (multi treading).

Tread කියන class එක java.lang කියන package එකේ තමයි තියෙන්නේ.ඒක නිසා import කරන්න අවශ්‍ය නෑ.

Tread එකක් start කරන්න start method එක call කරන්න ඕනි.

---------------------------------------------------------------------------------------------------------------------



Thread - main

**JVM**

---------------------------------------------------------------------------------------------------------------------

Main thread ​එක අපි හදන ඕනිම app ​එකක තියෙනවා. Main thread එ​කෙන් තමයි main method ​එක call කරන්නෙ.

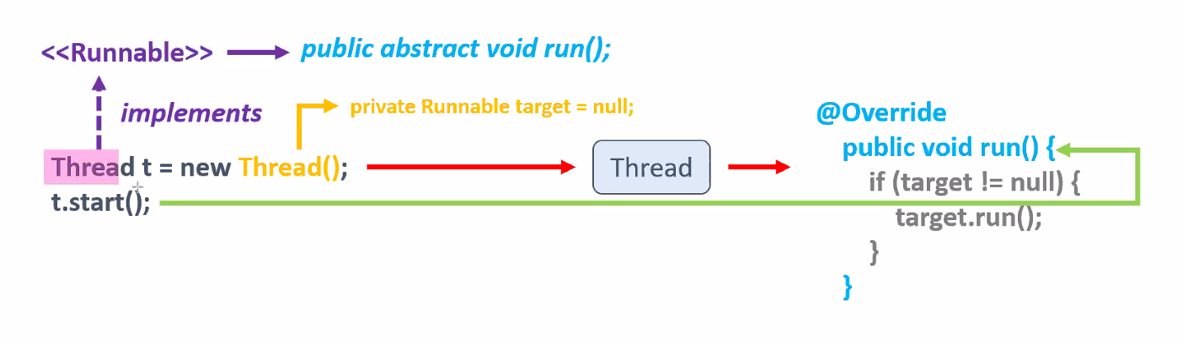
​Program ​එකක් run ​කරාම් JVM ​එක තමයි class load ​කරන්නෙ. ClassLoader object එකෙන් තමයි JVM එ​ක load ​කරන්නෙ.

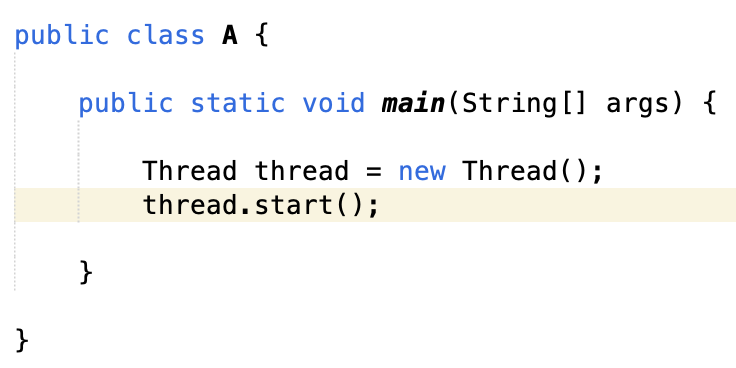
ඉස්සෙල්ලම main method එක තියෙන class ​එක load ​වෙනවා. Class ​එක load ​කරල main thread ​එක හදනව. Main thread එක හදල ඒකට භාර දෙනව main method එක execute කරන වැඩේ කරන්න.

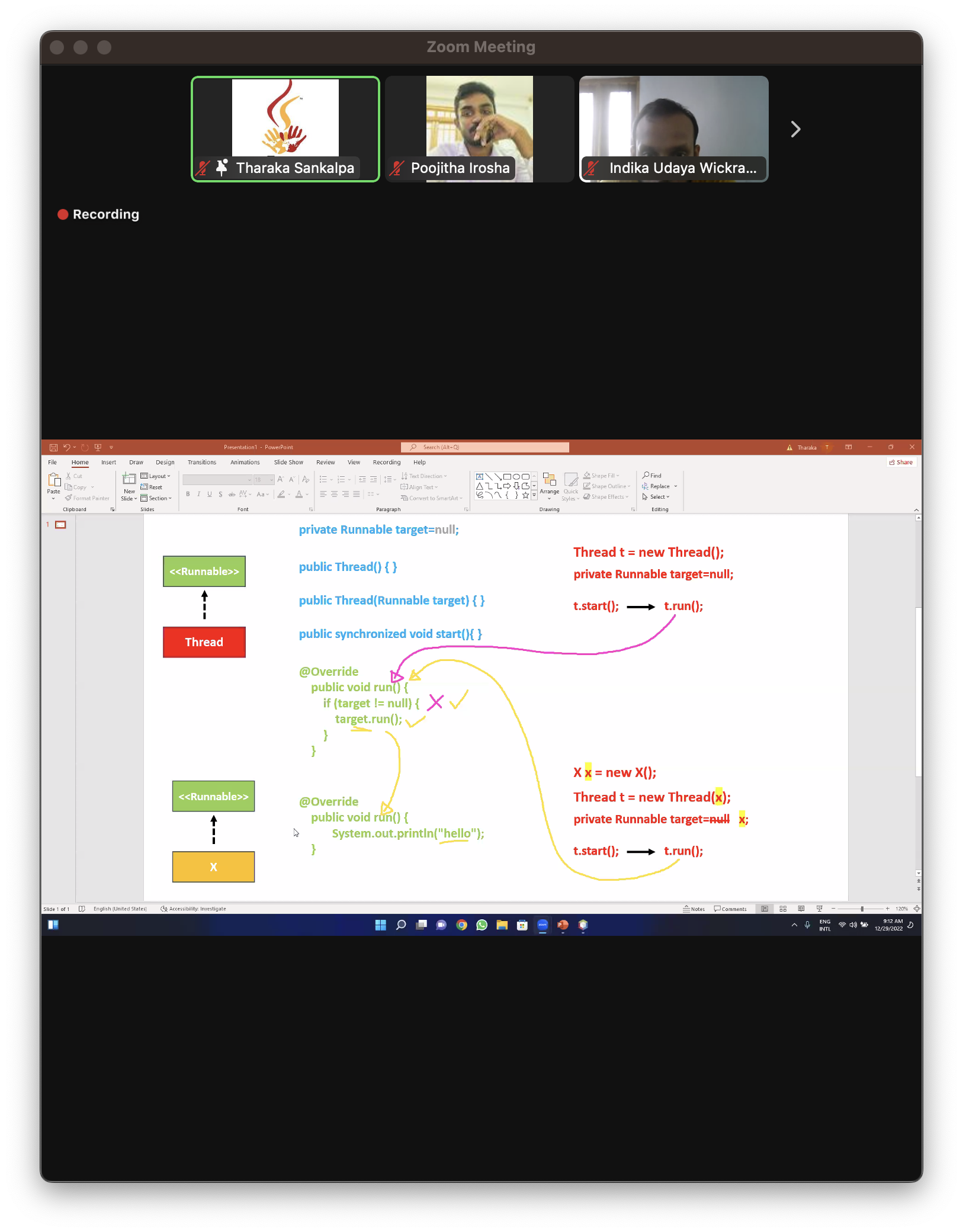
## ​අපෙ program එකක් run ​කරපු ගමන් JVM ​එක class ටික load ​කර ගන්න​වා. ඊටපස්සෙ JVM එක විසින් main thread ​එක හදනවා. එයා main method එ​ක call කරනවා.

Static blocks load කරන්නෙත් main thread එකමයි.2

---------------------------------------------------------------------------------------------------------------------







---------------------------------------------------------------------------------------------------------------------

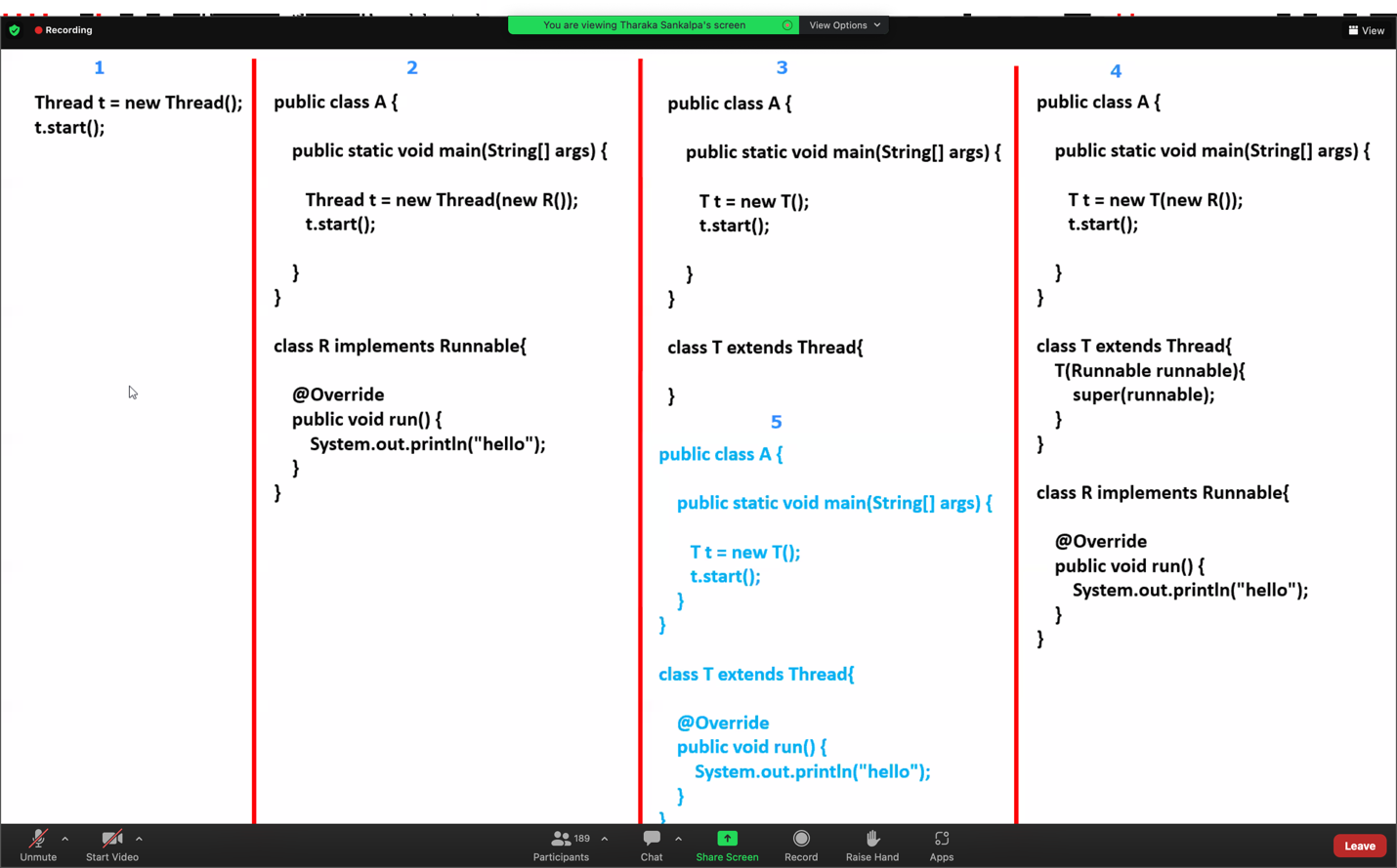
Thread ​එකේ Runnable target ​එක null ​නිසා ඒකෙන් අපිට වැඩක් ​නෑ. අපි වෙනම Runnable ​කෙනෙක් හදල සෙට් කරල තම​යි අපිට වෙන්න ඕනි වැඩේ කර ගන්නෙ.

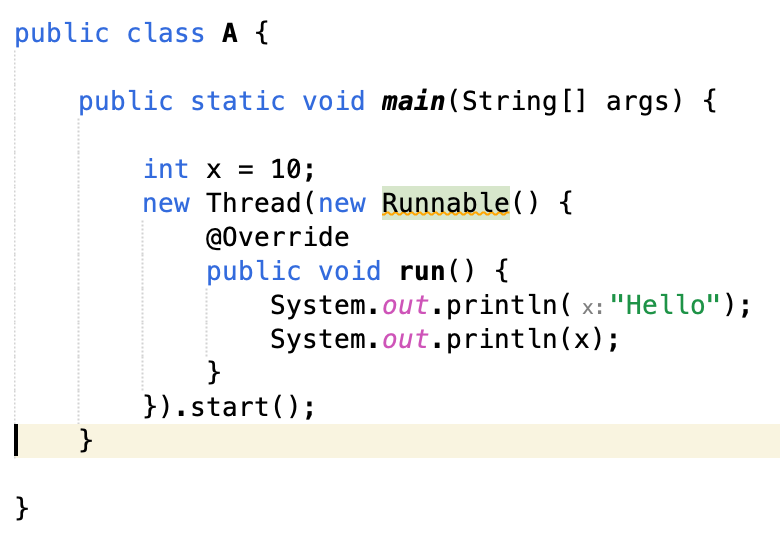
## Thread ​එකක් හදාගන්න පුළුවන් වි​දි

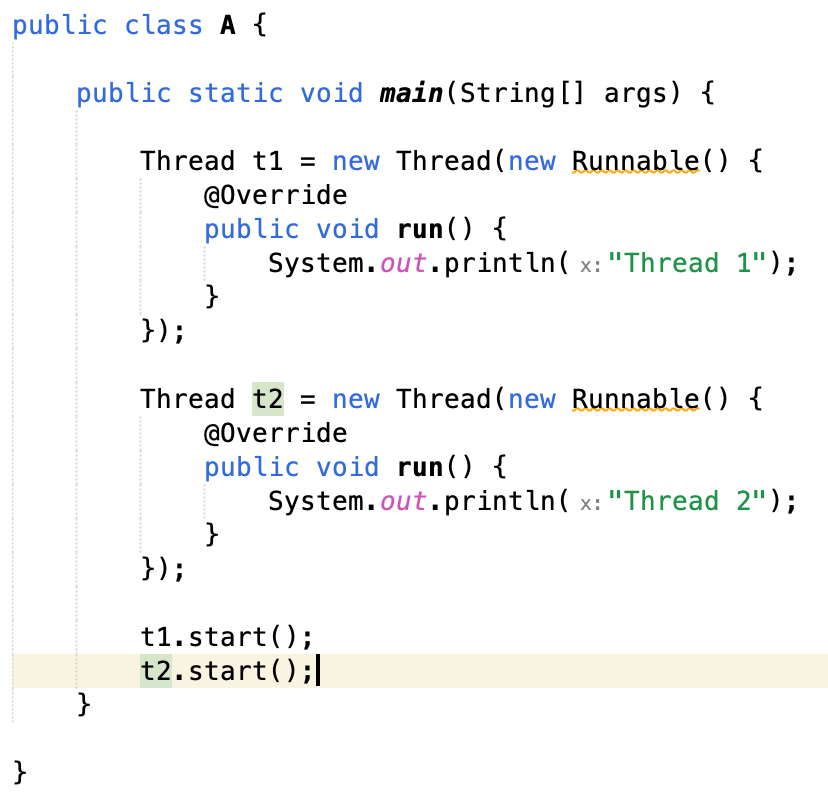
Original Thread class ​එකෙන් object එකක් හදල, original target ​එකම use ​කරන්න පුළුවන්. (ඒක වැඩක් ​නෑ).

Original Thread class එ​​කෙන් object එකක් හදල, customize කරපු Runnable target ​එකක් set කරනවා.

Customize thread එකක් හදලා run method එක customize කර ගන්නවා runnable target එකක්පවිච්චි කරනවා වෙනුවට.

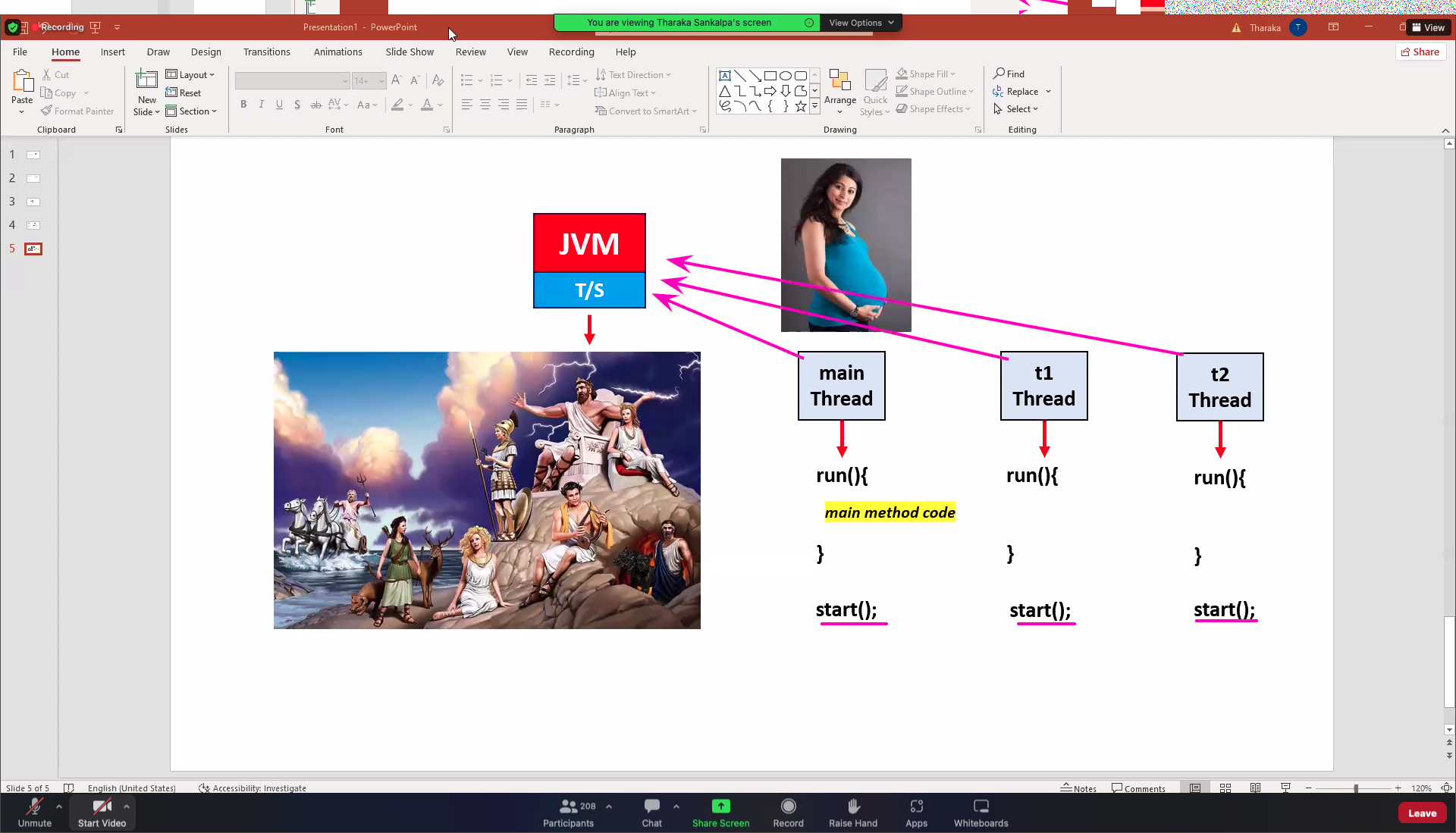
****

****

* ​මේ program ​එකේ threads 3ක් තියෙනවා.
* මුලින්ම main thread එක හැදෙනවා, ඊට පස්සෙ t1 thread හැදෙනවා, ඊටත් පස්සෙ t2 thread ​එක හැදෙනවා.
* මුලින්ම run වෙන්නෙ main thread ​එක, ​හැබැයි t1 t2 ​දෙන්නගෙන් කවුරු ඊළගට වැඩ කරයි ද කියල කියන්න ​බෑ.

JVM එකේ component එකක් තියෙනවා Thread Scheduler ​කිය​ල.

එක වෙලාවක chance එක දෙන්නෙ එක්කෙනෙක්ට විතරයි. කෙටි කාලයක් ඇතුළත ගොඩක් කට්ටියට chances ​දෙන නිසා, එකම වෙලාවෙ වැඩ කීපයක් වෙනව අපිට බලාගන්න පුළුවන්.



​Thread Scheduler - ​සර්ව බලධාරී දෙවියන් වහන්සේ.

Main thread - අම්මා.

t1, t2 - ​ළමයි.

Run() - දෙයිව​ය

​එක්කෙනෙක්ගෙ දෙයිවය line by line තමයි execute කරන්නෙ.

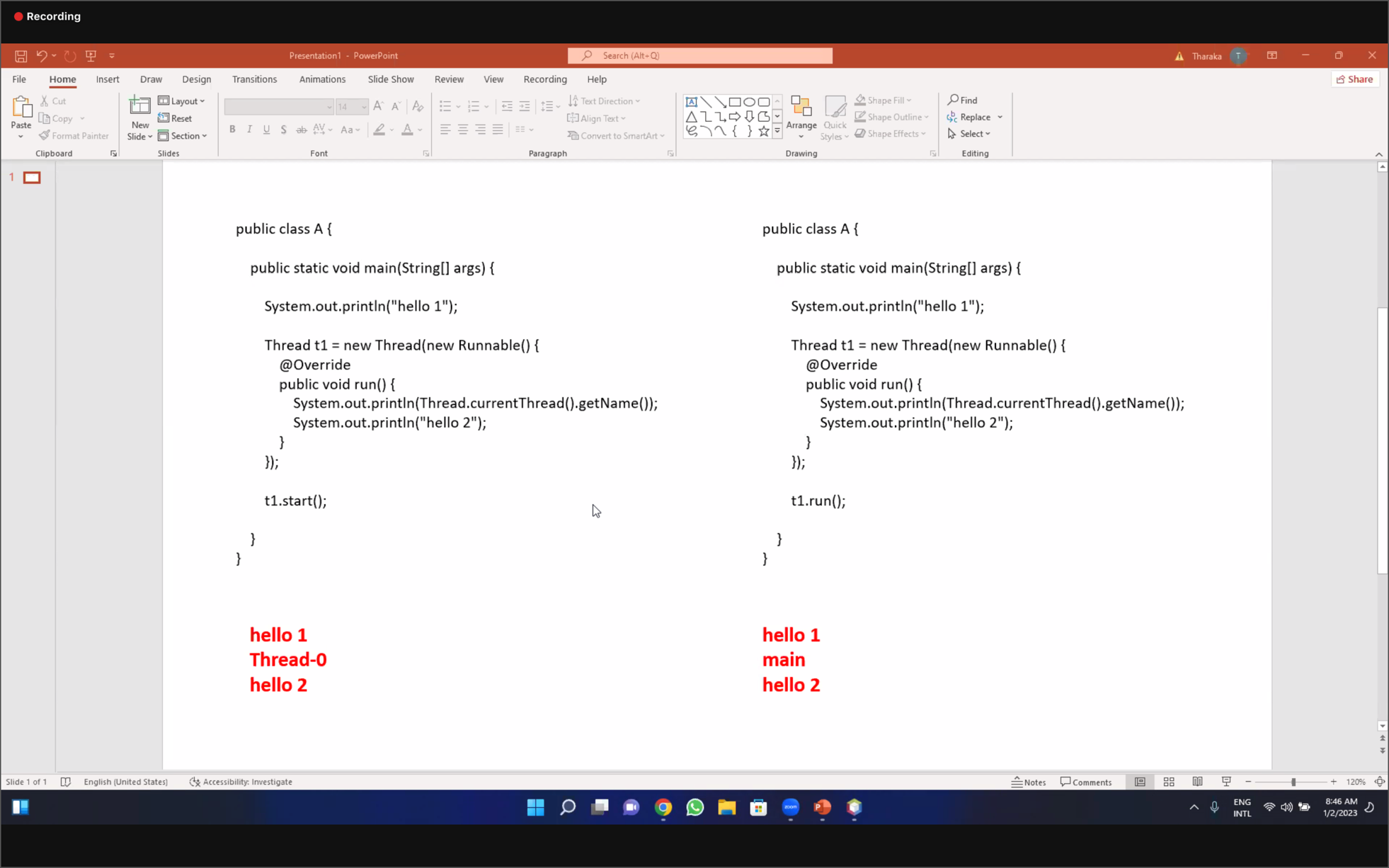
start() - ​ඉපදෙනව, ​ඉපදුනාම තමයි ​සර්ව බලධාරී දෙවියන් එක්ක සම්බන්ද වෙන්නෙ.

Thread ​එකකට නමක් දාගන්න පුළුවන් විදි

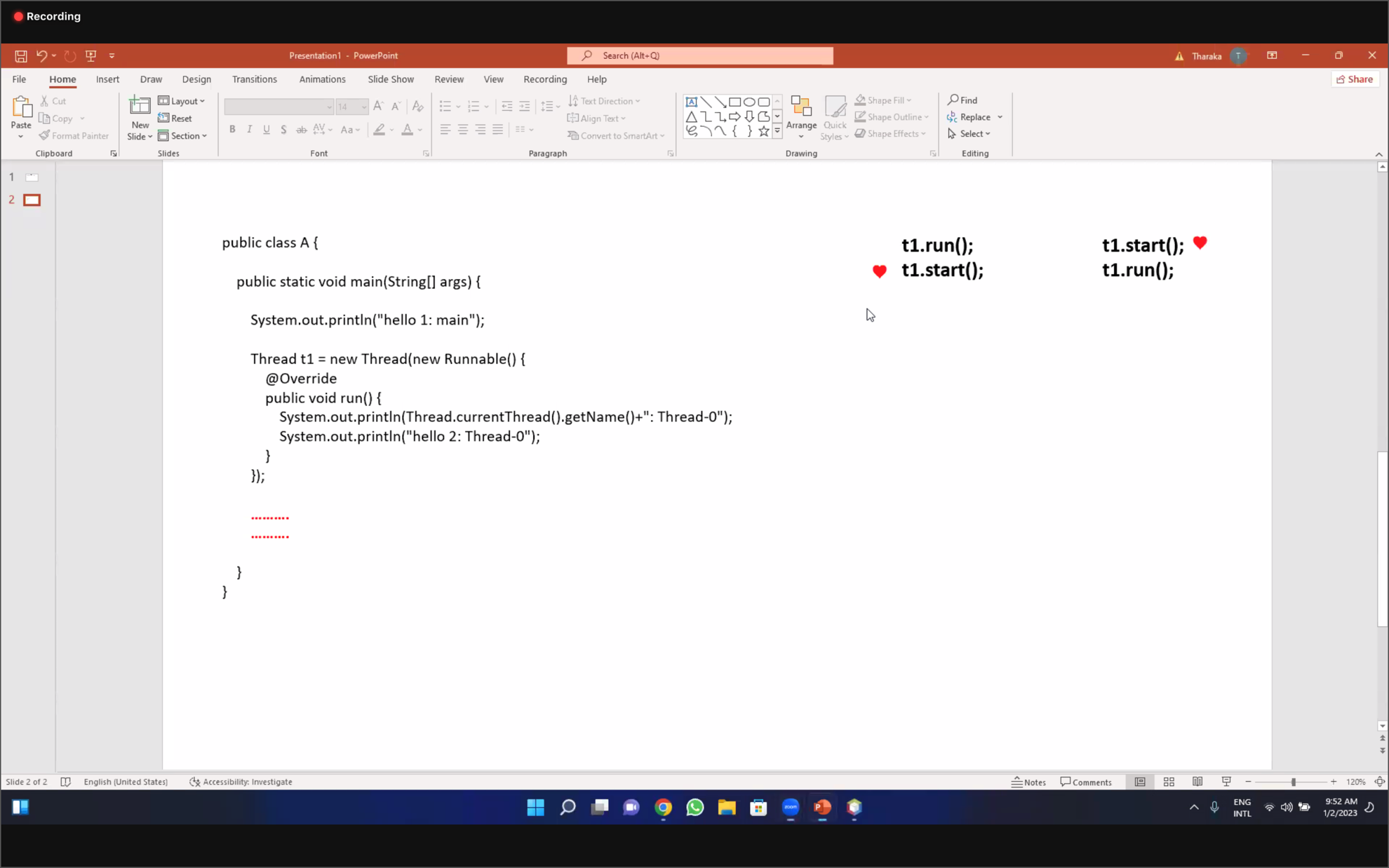
**Constructor**: *public Thread(Runnable target, String name) {}*

**Setter Method:** *public final synchronized void setName(String name);*

Thread එකකට නමක් දුන්නෙ නැත්තන් default නමක් වැටෙනව “Thread-0, Thread-1, Threed-2,..” ​කියල.



* ​මේ අවස්ථා දෙකේදිම thread obj 2ක් හැදෙනවා.
* හැබැයි thread-0 ​කියන thread එක start කරොත් විතරයි thread scheduler​ට connect ​වෙලා thread එකක් විදියට වැඩ කරන්නෙ.
* Run එක call ​කරනව කියන්නෙ main thread එ​කේ දෛවයෙන් t1 object ​එකේ method calling එකක් වෙන එක විතරයි.



**t1.run();**

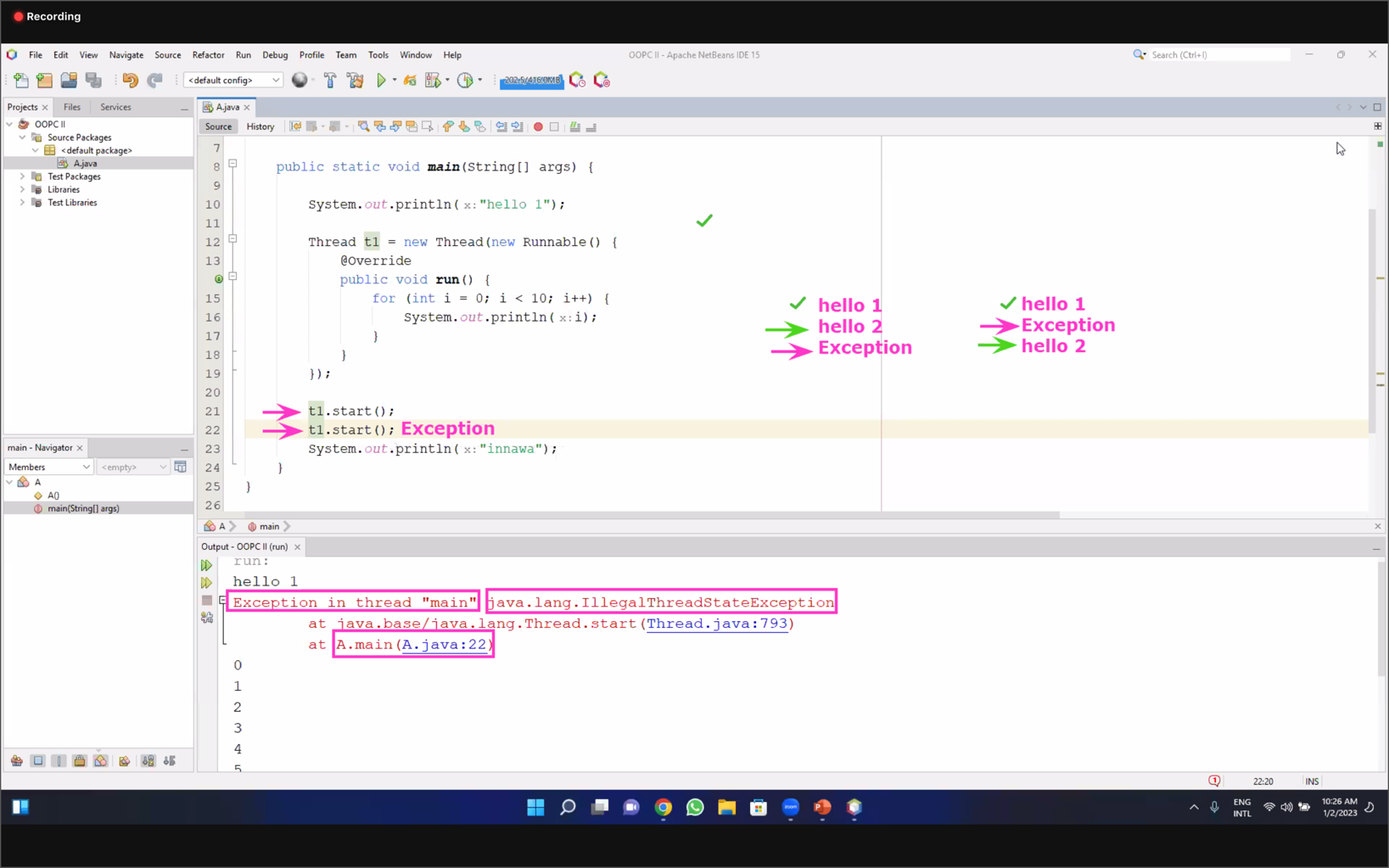
**t1.start();**

main thread ​එක t1 ​object එකේ run method ​එක call ​කරල ඉවර උනාම තමයි t1 object ​එක thread ​එකක් විදියට වැඩ කරල run method එක execute වෙන්නෙ.

**t1.start();**

**t1.run();**

t1 object ​එක thread ​එකක් විදියට වැඩ කරන්න ගත්තට පස්සෙ තමයි main thread ​එක t1 object එ​කේ run method ​එක call ​කරන්නෙ. Thread ​දෙකෙන් කොයිවිදියට මේ method දෙක call ​වෙයිද කියල අපිට මෙතනදි කියන්න බෑ.



​එක thread ​එකක exception එකක් ආවොත් ​ඒ thread ​එක තමයි නවතින්නෙ අනිත් thread ​වලට බලපෑමක් ​නෑ.

එකම thread ​එක දෙපාරක් start ​කරන්න බෑ එහෙම කරොත් IllegalThreadStateException ​එකක් එනව.

