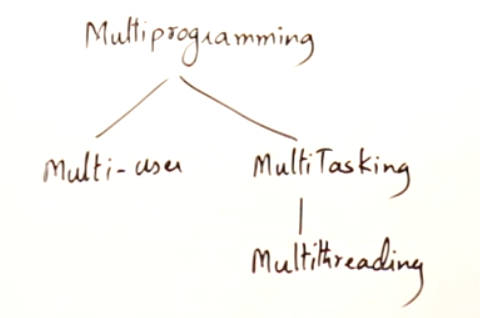
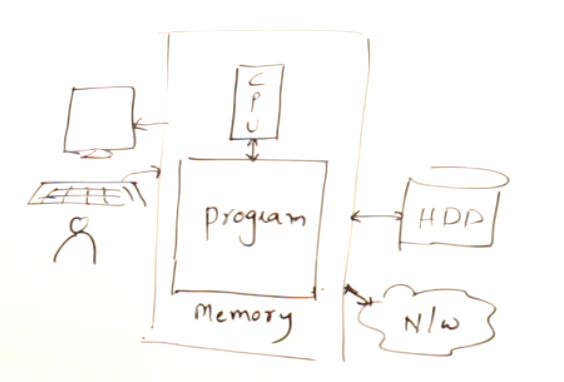
# Multi-programming



Need for multi-programming

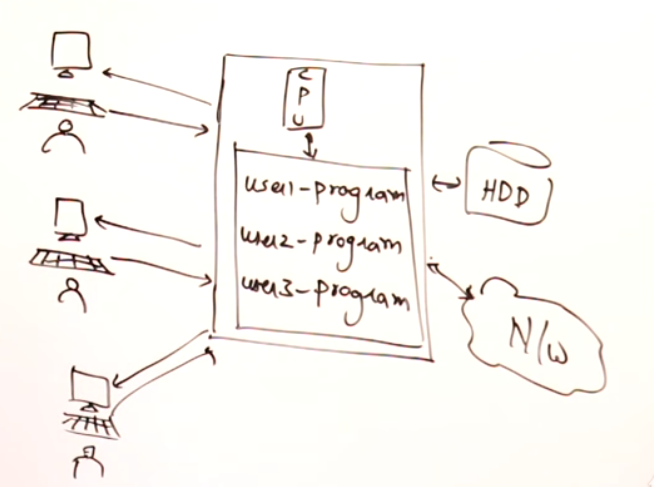


Suppose when you type 10-15 pages in MS-word which takes nearly an hour to complete, CPU will be utilized efficiently only for 5 minutes.   
CPU is utilized only for checking spelling mistakes, etc..

Rest of the time, CPU is idle. So to make use of the CPU efficienlyt, the concept of multi-programming comes into picture.

So that when you type 10-15 pages in MS-word, side by side you can also run some other programs at the background which will make use of the CPU so efficiently. Hence our aim is always to keep CPU busy all the time. This will taken care by the OS.

# Multi-user



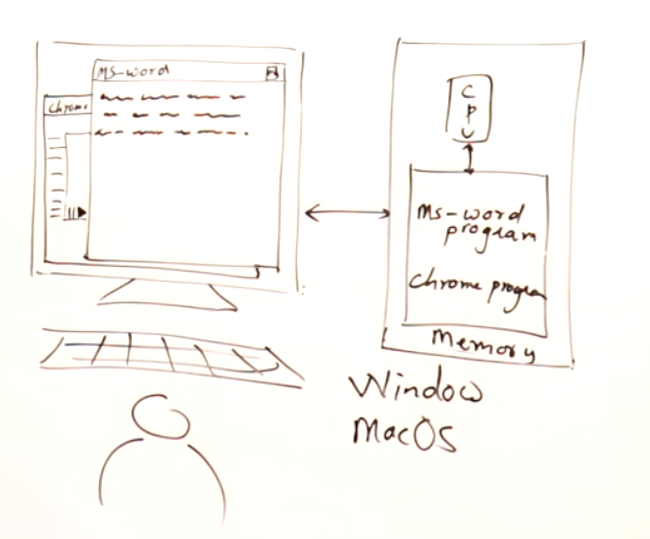


Multi-user 🡪 More than 1 user uses the same PC at the same time.

3 individual users are using the same PC, so that the CPU is always busy. These are taken care by famous OS such as Unix and Linux. These kind of multi-user practices are now outdated.

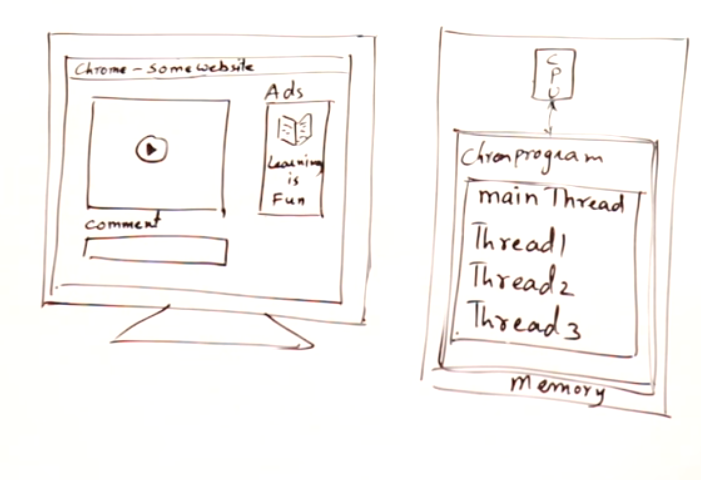
# Multi-tasking

Multi-tasking 🡪 A single user making multiple tasks on a PC simultaneously.

  
Actually CPU cannot proceed all programs simultaneously, it running alternatively (i.e changing programs) at a faster rate, so we don’t feel or see the shift. Programs are running on the CPU alternatively and not simultaneously.

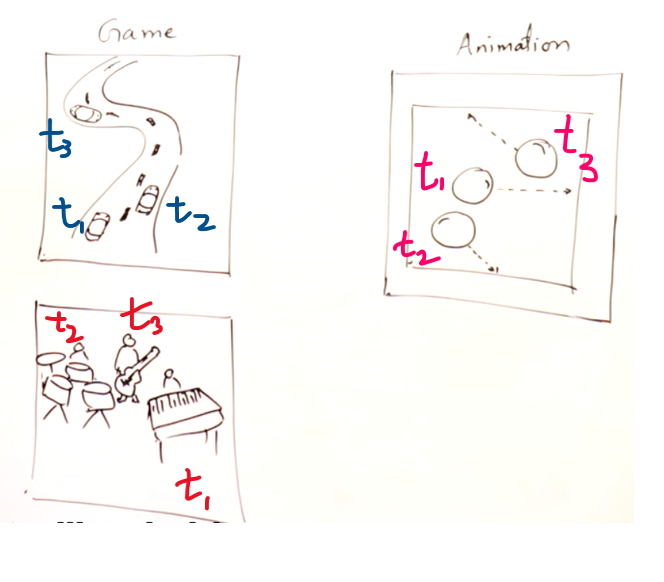


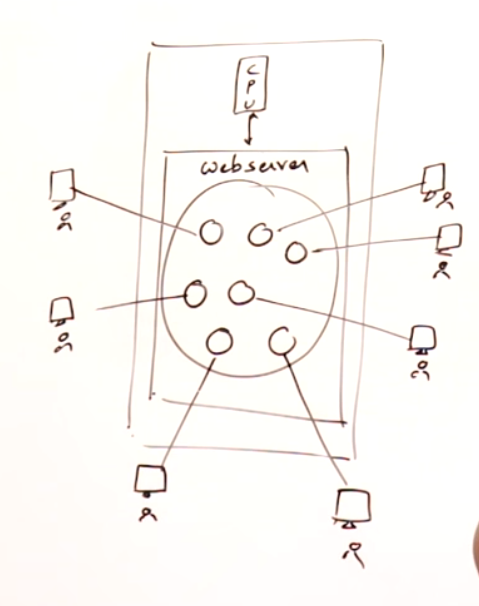
# Multi-threading



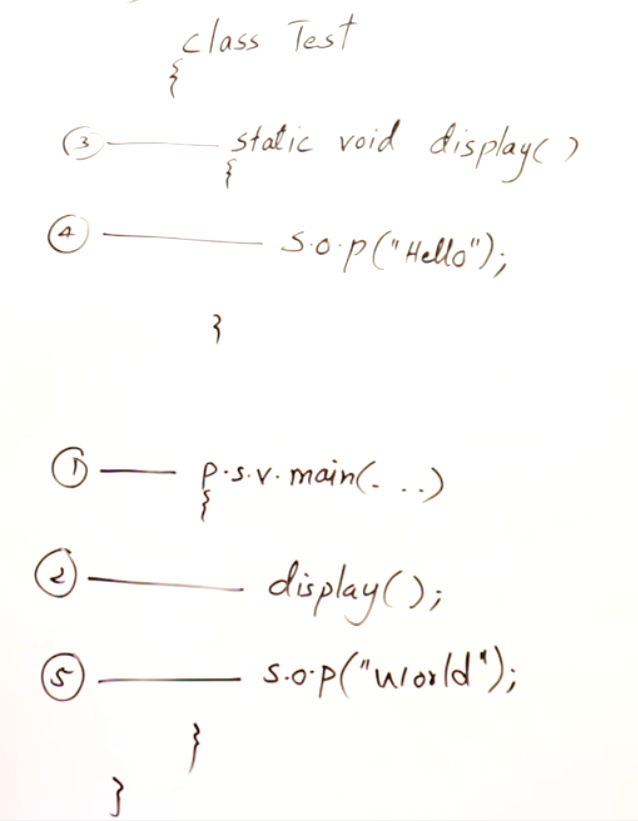
main thread () 🡪 Overall chrome application  
thread1 🡪 to play the video  
thread2 🡪 to showcase the adds  
therad3 🡪 to put the comments

So a single chrome application is divided into multiple threads and executed.  
Threads can also be executed one-by-one, by the speed of CPU we cannot able to see the shift and we are assuming that all threads are executing simultaneously.

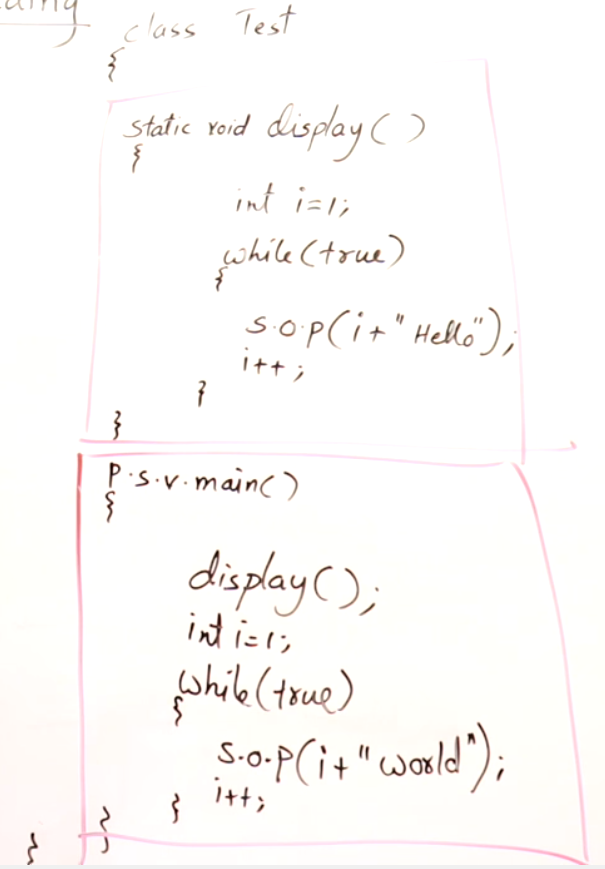


 Bubbles 🡪 threads

# Control flow of a program



This is the normal control flow while handling with functions





i+“hello” and i+“world” should run simultaneously. How to do???  
Assign thread1 to i+”hello” and thread2 to i+”world” , so that two threads can run simultaneously.  
Multi-threading concepts come into picture.

# Multi-threading using Thread class

How to achieve multi-threading ????

1. Thread class
2. Runnable Interface

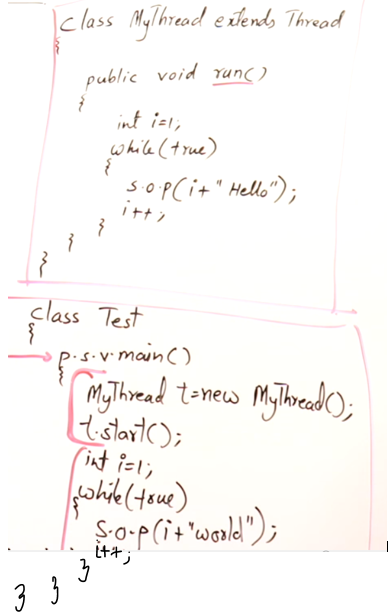
If your class is not inheriting any other class, then you can inherit thread class and can achieve multi-threading.

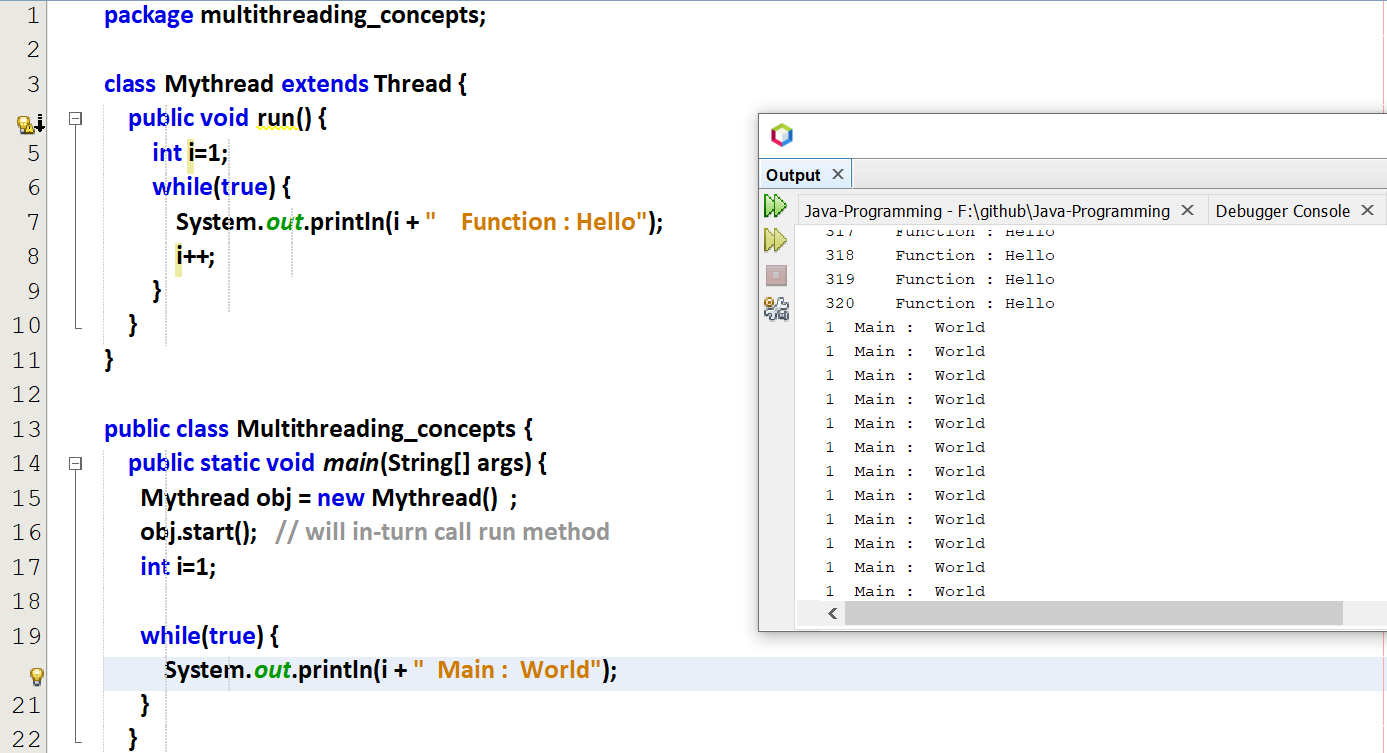
If your class is already inheriting some other class, then you cannot inherit Thread class. Since JAVA class can only inherit one class. In this case your class can implement Runnable interface and can achieve multi-threading. Since JAVA class can implement multiple interfaces but can extends(i.e inherit) only 1 class.

What-ever logic we want to execute in a thread, we should write that logic inside the run().  
We are over-riding the run() since run() is already defined in Thread class.  
main() 🡪 starting point of a program.  
run() 🡪 starting point of a thread.

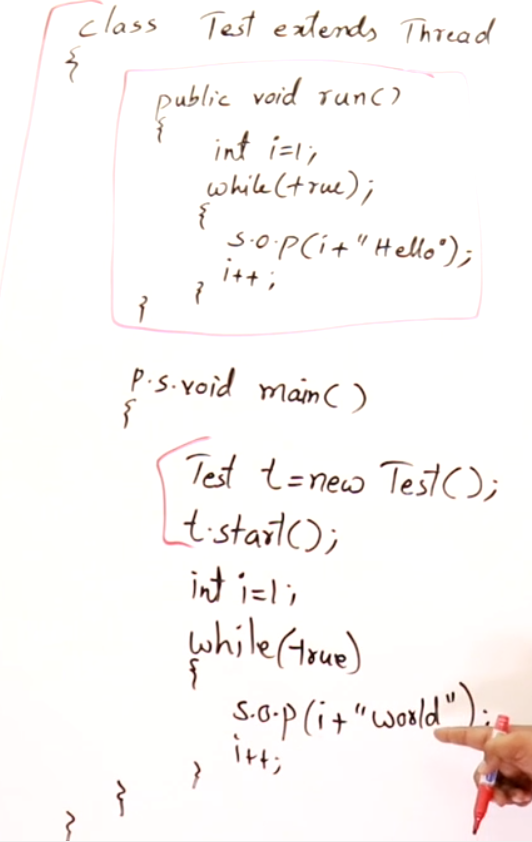
1)Create an object of the class which you want to execute as a thread.  
2) Then obj.start(), start() is already defined in Thread class and it knows how to start a thread and class the run()

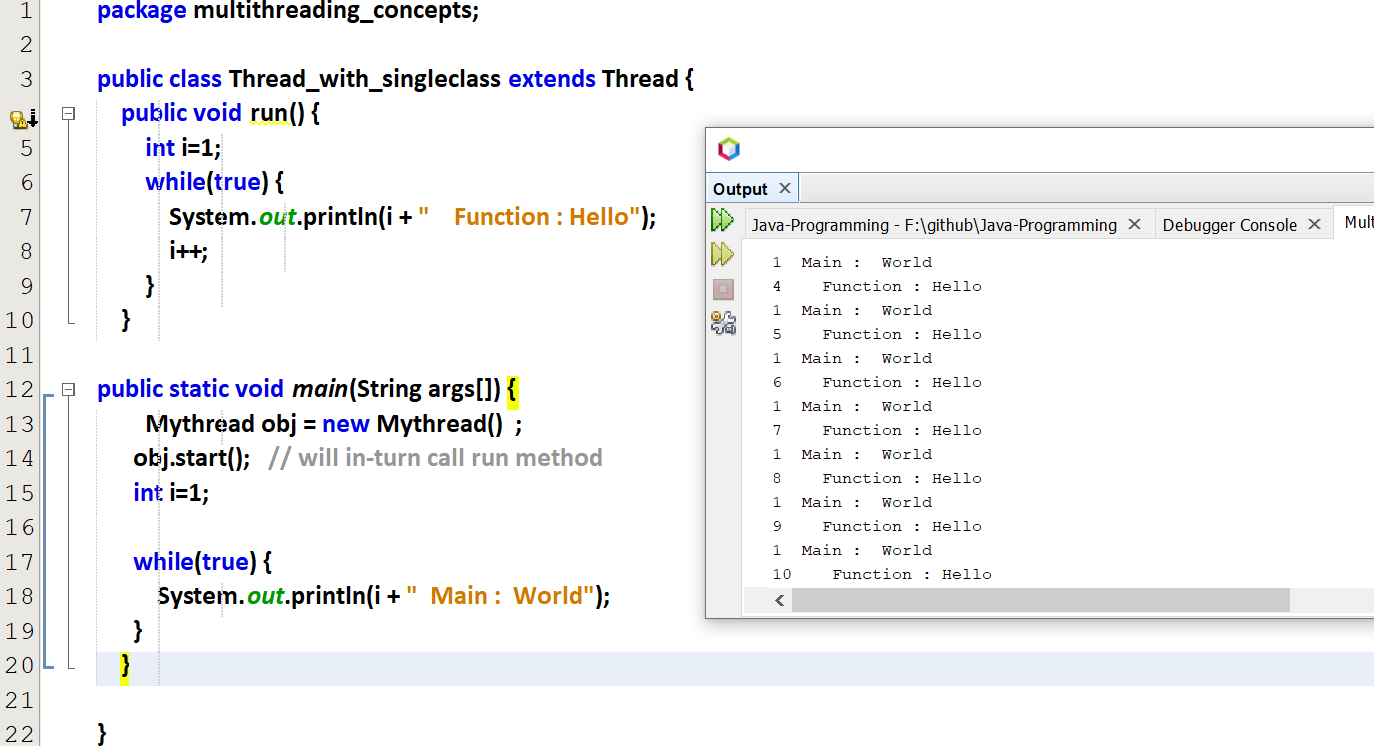
## Using two classes





## Using a single class





# Multi-threading using Runnable Interface

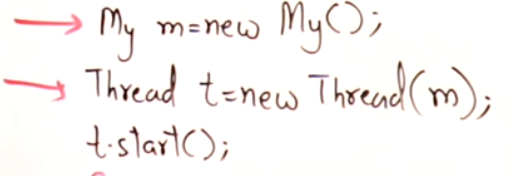


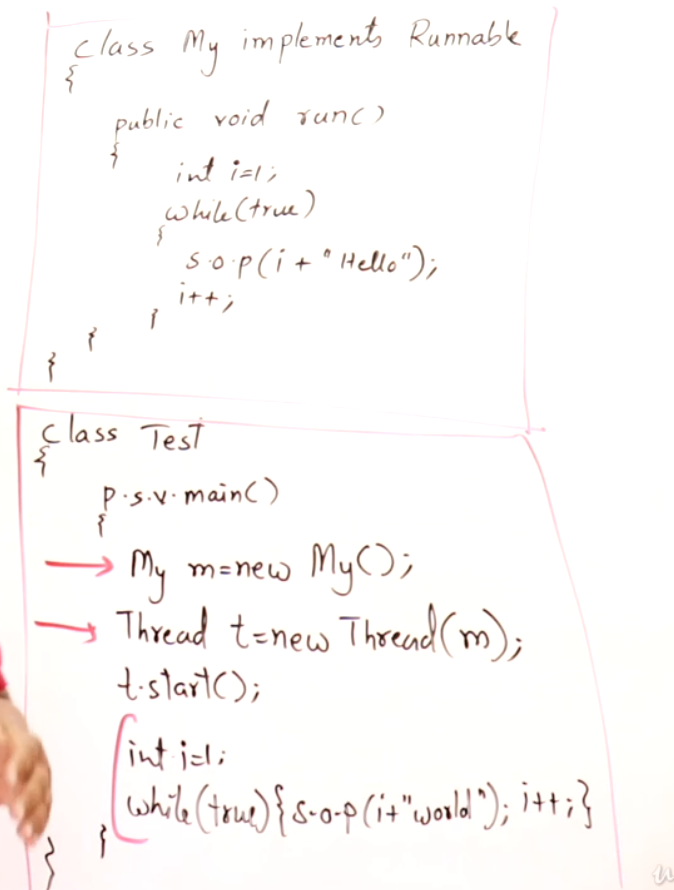
When a class implements an interface, It should over-ride all the methods in that interface.  
So it is very-very important to over-ride run(). And also only run() is available in Runnable interface.

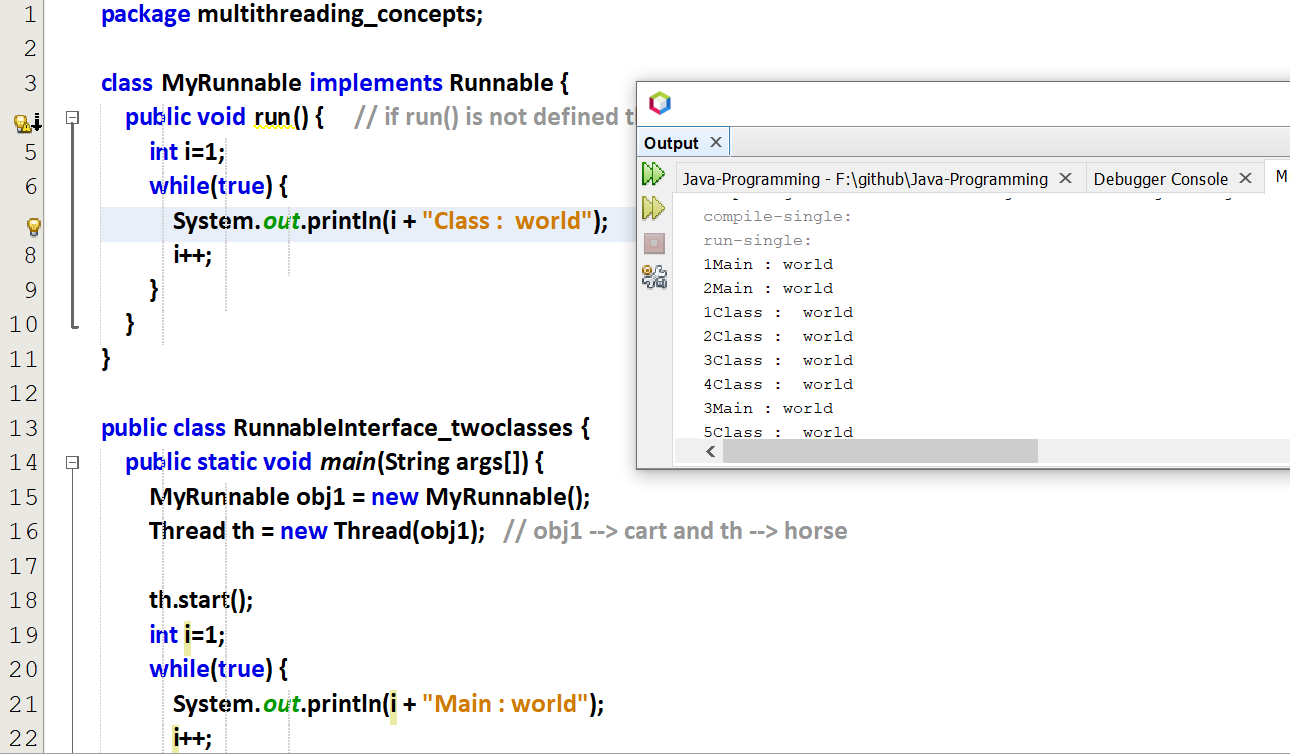
## Using two class

Runnable interface is capable of running a thread, but it cannot run by-itself. So it seeks the help of Thread class.

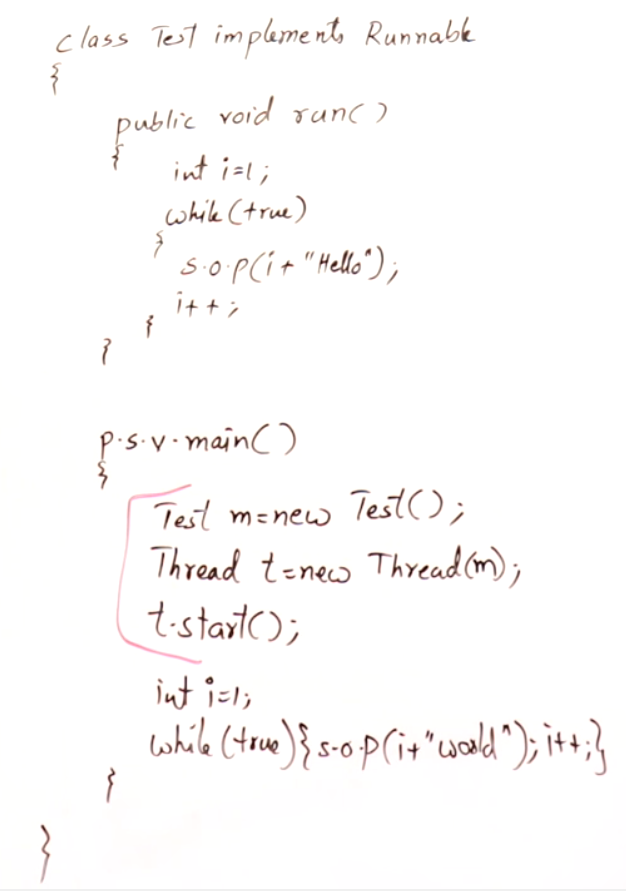
t 🡪 horse  
m 🡪 cart attached to that horse  
So that when the horse runs, it will also drag the cart along with it.

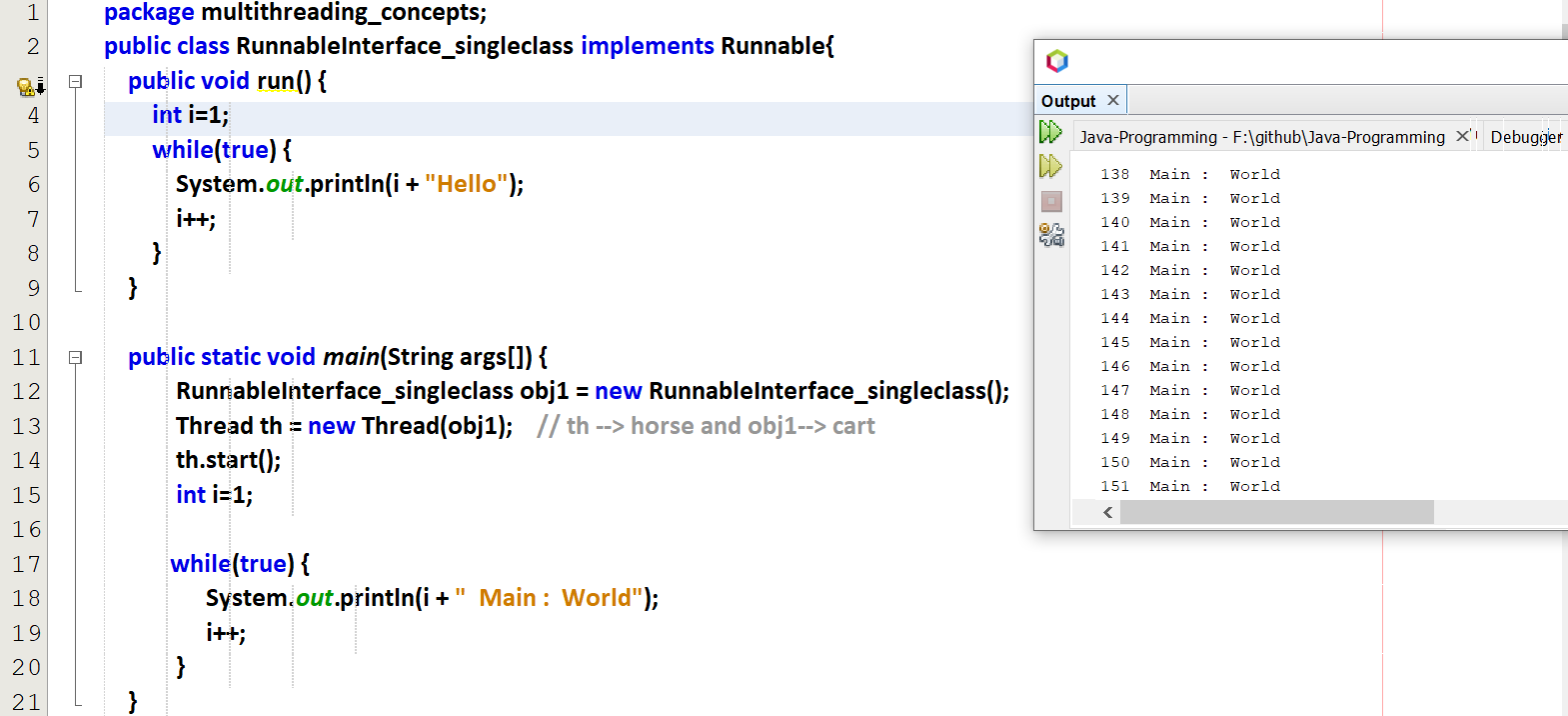




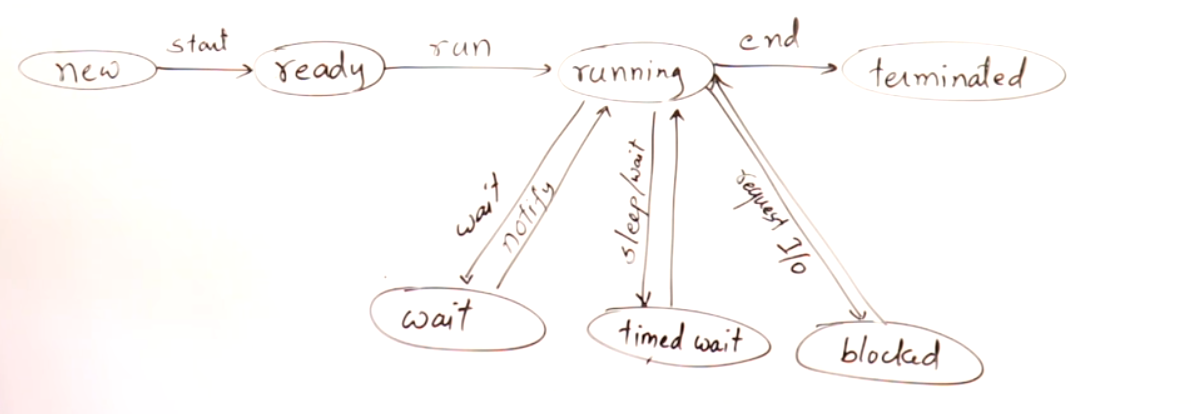


## Using single class



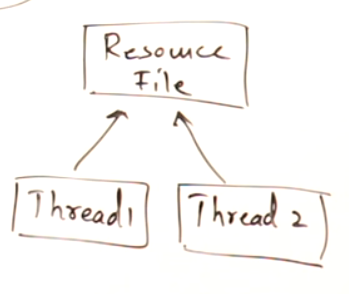


# States of a thread



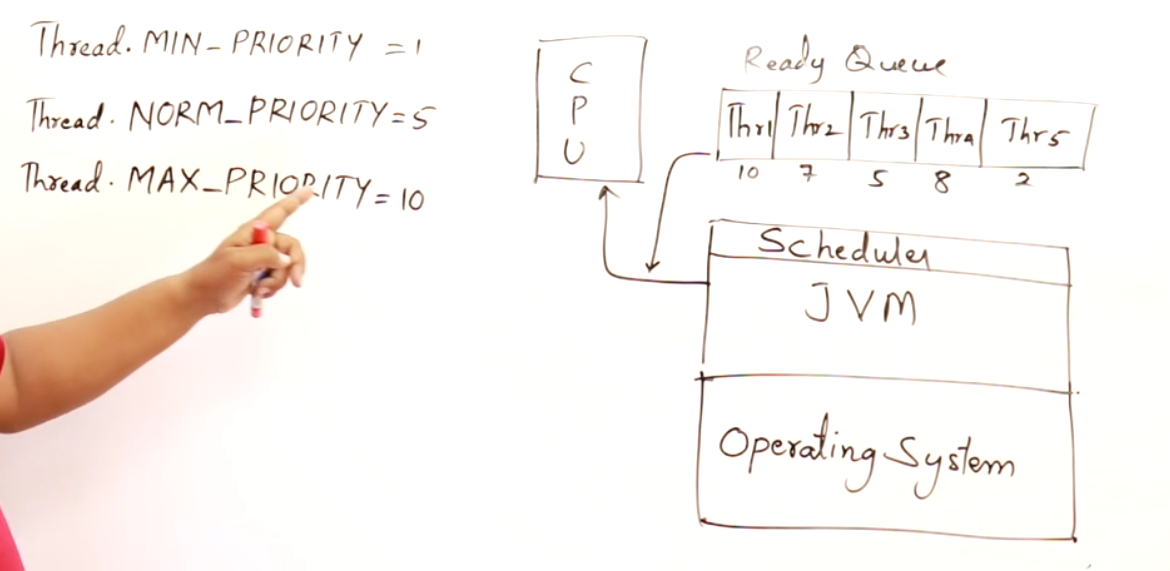
If we call start, in-turn it will call run()  
If a thread is terminated we cannot able to restart that thread, instead we have to create a new thread and start the process.

Resource should be accessed by one thread at a time.



Similar to a printer used in office, where there will be only 1 printer and all other users wants to access the printer, so they must wait in a queue and take the printout one-by-one at a time.

# Thread Priorities

  
Scheduler in JVM will maintain the ready queue.

A thread having higher priority will be executed first.

## **Eg: MS-Word**

t1 🡪 getting input from the user  
t2 🡪 spell check  
t3 🡪 autosaving the document

In the above operations how the priorities are provided ???  
If the 1st priority is given to t3, then what all you type will not apply on the screen time to time, there will be slight delay in displaying which will irritate the user while typing.  
So 1st priority 🡪 t1 , 2nd priority 🡪 t2 , 3rd priority 🡪 t3  
if this takes place in-order then there will be no problem in MS-Word.

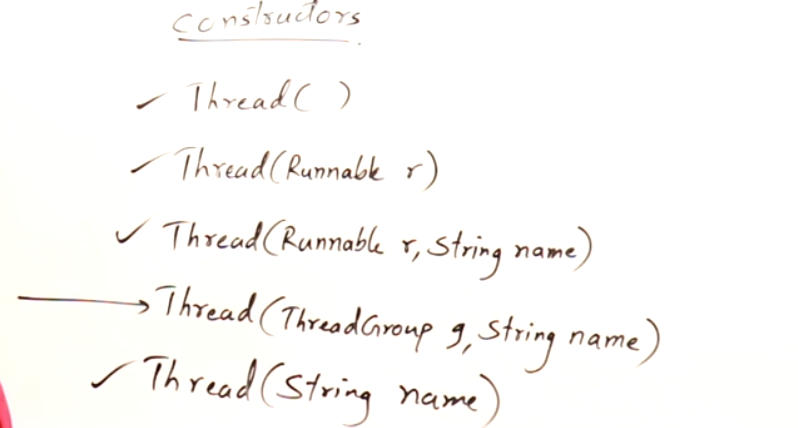
## **Eg: Chrome Browser**

t1 🡪 pulling the data from the server  
t2 🡪 rendering the pulled data to the display

Some-times it takes for a page to reload to some extent. This is because without pulling the complete data from the web-server, it cannot able to render that pulled data into the monitor. So the web-pages loads and after that only it renders on the monitor display.  
So 1st priority 🡪 pulling the data from the server  
2nd priority 🡪 rendering the pulled data to the display

# Thread Class

## Constructors in thread class

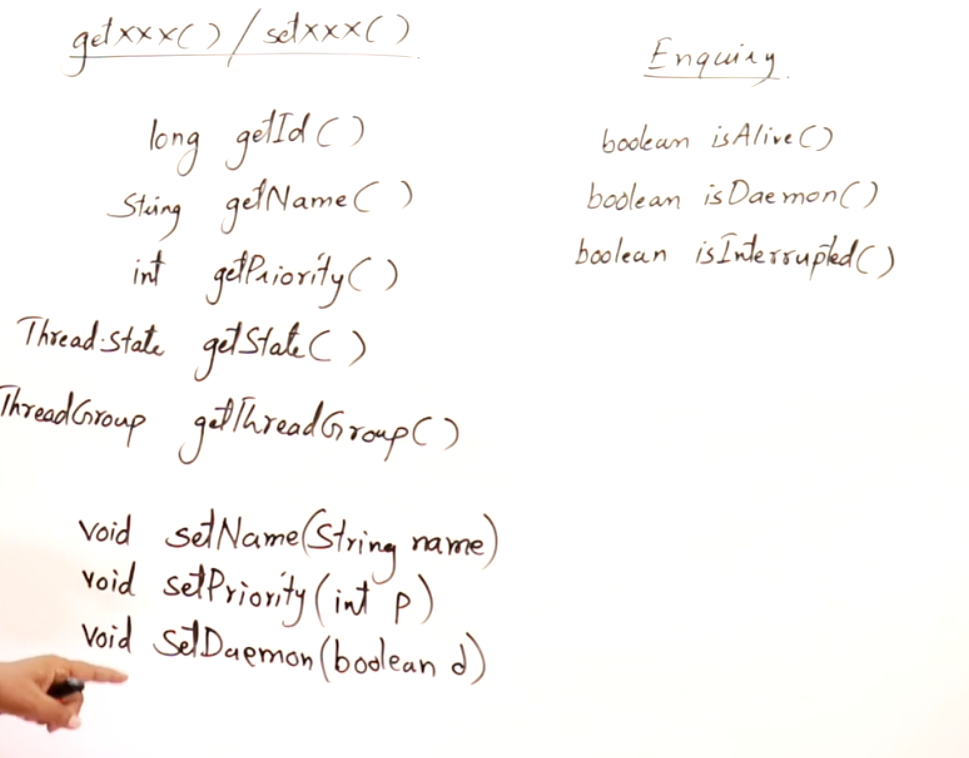


ThreadGroup 🡪 we can create a group of threads and make a sync among them.  
Eg: While designing a game, we assign all threads to all small-small operations through out the game and have a complete control over-it. So if we pause that game then all the threads will be stopped.

We can get id of a thread, but we cannot set id of a thread.  
We can get and set a name to a thread.

Daemon thread 🡪 A thread which is having very least priority.  
Eg: In MS-Word, autosave option is done only at the last, so the autosave option is set to as demon thread. So that it happens at the last.  
If we give the boolean d 🡪 true [ Demon ]   
If we give the boolean d 🡪 false [ Not Demon ]

## Get and Set method in thread class



## Instance method in thread class