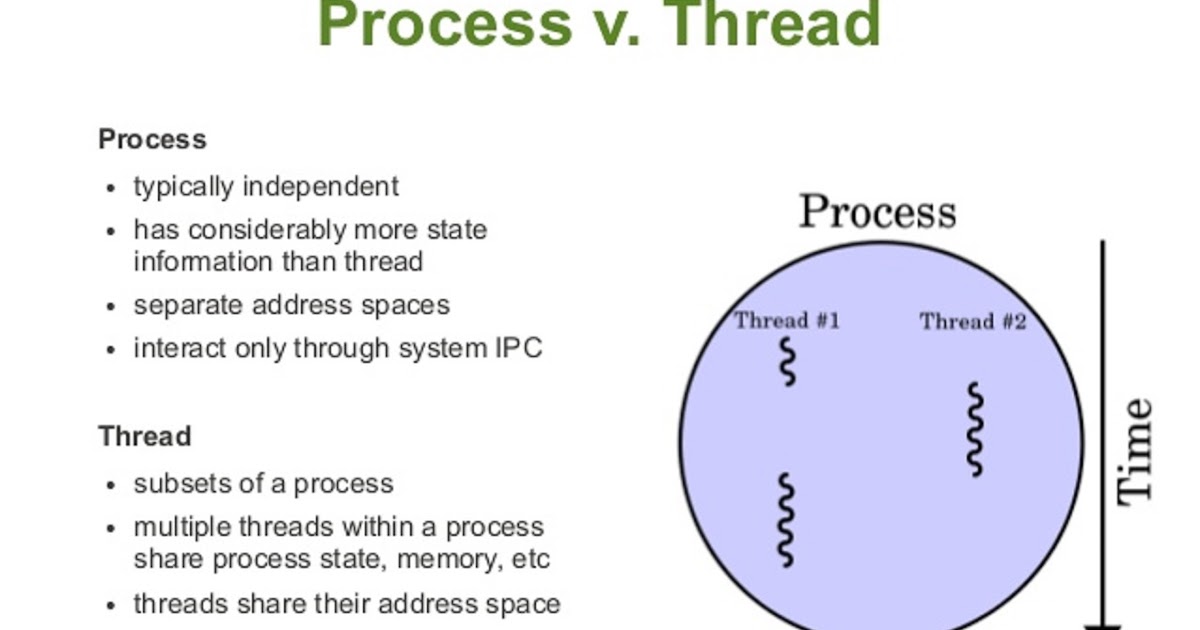
Multithreading Implementation

Thread:

* A thread is a lightweight sub process, the smallest unit of processing. *(It is a separate path of execution.)*
* Threads are independent. *(If there occurs exception in one thread, it doesn't affect other threads. It uses a shared memory area.)*

## Process:

* **A process is an instance of a program that is being executed.**
* When we run a program, it does not execute directly. It takes some time to follow all the steps required to execute the program, and following these execution steps is known as a process.



# 

# Multithreading in Java

* **Multithreading in**[**Java**](https://www.javatpoint.com/java-tutorial) is a process of executing multiple threads simultaneously.
* However, we use multithreading more than multiprocessing because threads use a shared memory area. *(They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.)*

Advantages & Uses of Multithreading are

* Java Multithreading is mostly used in games, animation, etc.
* Multithreading saves time as you can perform multiple operations together.
* The threads are independent, so it does not block the user to perform multiple operations at the same time and also, if an exception occurs in a single thread, it does not affect other threads.

Threads can be created by using two mechanisms:

**Note:** It extends from java.lang.Thread class **(Pre-defined class)**

* Extending Thread class **[E.g. class class\_name extends Thread]**
* Implementing Runnable Interface **[E.g. class** **class\_name implements Runnable]**