**Tutorial No. 2**

**Aim**:- Implement a client-server calculator program using multithreading.

**Theory**:-

Java is a *multi-threaded programming language* which means we can develop multi-threaded program using Java. A multi-threaded program contains two or more parts that can run concurrently and each part can handle a different task at the same time.

Multi-threading extends the idea of multitasking into applications where you can subdivide specific operations within a single application into individual threads. Each of the threads can run in parallel.

But we use multithreading than multiprocessing because threads share a common memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process. Java Multithreading is mostly used in games, animation etc.

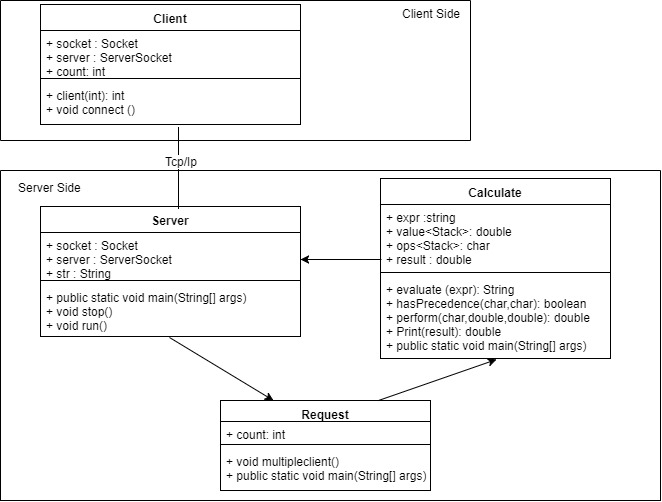
Advantages of Java Multithreading:

* Enhanced performance by decreased development time
* Simplified and streamlined program coding
* Improvised GUI responsiveness
* Simultaneous and parallelized occurrence of tasks
* Better use of cache storage by utilization of resources
* Decreased cost of maintenance

The advantage of implementing the multithreading with the client server is that at a time we can connect many client with the server and they can perform the actions simultaneously. Expected results must be concerned with the respective user so port number has been given for each client for the communication.

The problem arises when we use the simple calculator without client-server where only one client can access the server at a time. The calculations also might be done in Nano-second but still multiple clients where not possible to handle. So the multithreading does this integral part of connecting many clients with server and performs calculations.

**Class Diagram:-**

****

**Conclusion**:-

Thus we have implemented the client-server multithreaded architecture where the server thread is generated for each client to perform its calculations and return the result to the client back.