**singleton Design Pattern:**

Define a class that has only one instance and provides a global point

of access that instance.

DESCRIPTION : -- Singleton design pattern is a creational design pattern and one of the main design pattern to learn, it gives great flexibility over global variables, but this design pattern should be build with good care otherwise developer may fall into lots of problem. And one of the main problem could be concurrency, when two or more threads are trying to access the same object which you created as singleton might end-up corrupting the data. SO one need to be very careful which design Singleton Design Pattern. In Singleton design pattern in c++ video tutorial i have tried to explain with the help of definition, requirements, advantages, usage, UML diagram, single design pattern in c++ live example.

**Requirements:**

-> One and only one instance

-> Global access (Anyone can use this object anywhere in the program)

-> No ownership

-> Lazy Initialization

**Advantages:**

a) save memory

-> Only one object is required so why to create so many objects.

b) single access point-

-> Logger (Alll the log prints), database connection

c) Flexibility

-> change anytime you want to

**Usages:**

a) MultiThreaded

- threadpool

Ex: Get a thread from the threadpool and once done with the work,

returning a thread back to the pool. This threadpool is implemented

using singleton.

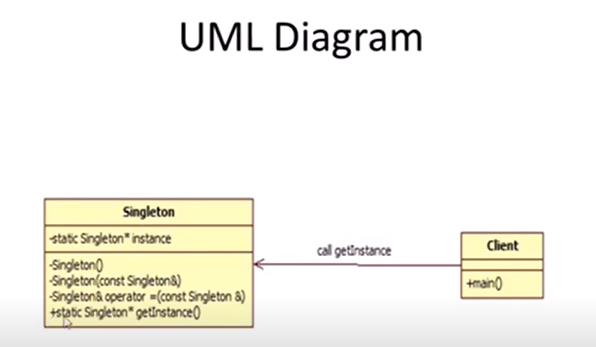
Because we can't have two thread pool's at the same time

b) Database Application

- Logging

c) configuration settings

- Game setting, application setting, etc.



How to create:  
To create singleton class we must have

* static member (can be directly accessed without a object)
* private constructor (Because we don’t want to allow multiple creation of object’s when using a singleton class)
* static function (which will return as the object and the Access specifier is public)