



```

        synchronized(Singleton.class){
            if(singleton==null){
                System.out.println(Thread.currentThread().getName());
                singleton = new Singleton(); //(That is demanded/Lazy
initialization)
            }
        }
    }
    return singleton;
}

public static void main(String[] args){

    Thread task1 = new Thread(new Runnable() {
        @Override
        public void run() {
            System.out.println("Frist Thread>>>>" + Singleton.getInstance());
        }
    }, "Frist");

    Thread task2 = new Thread(new Runnable() {
        @Override
        public void run() {
            System.out.println("Second Thread>>>>" + Singleton.getInstance());
        }
    }, "Second");
    task1.start();
    task2.start();
}
}

```

### Output:

Frist

I am constructor

Frist Thread>>>>designPattern.Singleton@2ae9093c

Second Thread>>>>designPattern.Singleton@2ae9093c

**Please add on comments and like this lesson, if it gave you some revision or refreshment on Singleton class concepts.**