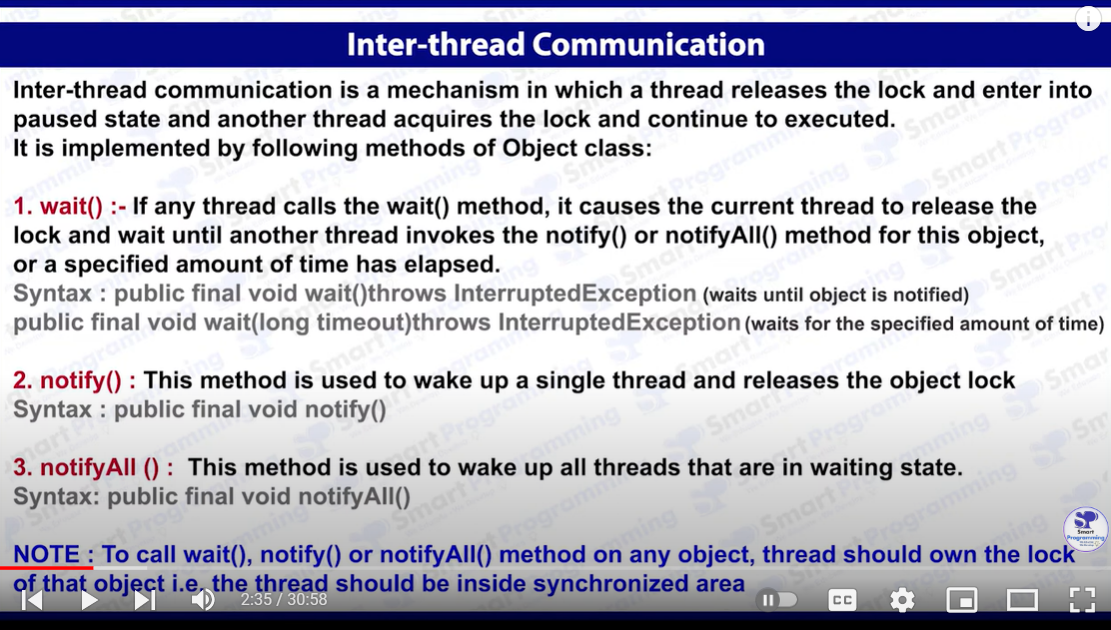
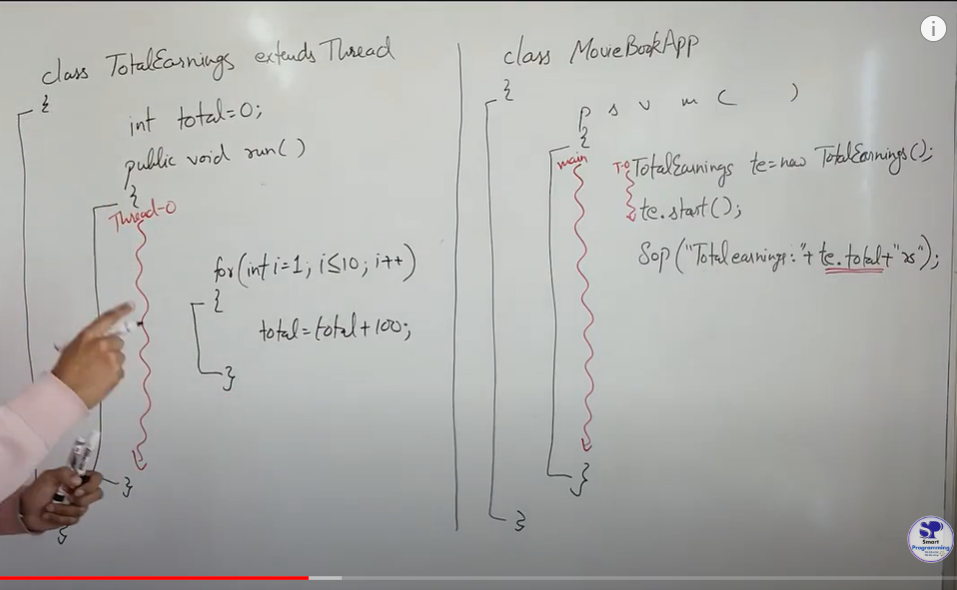
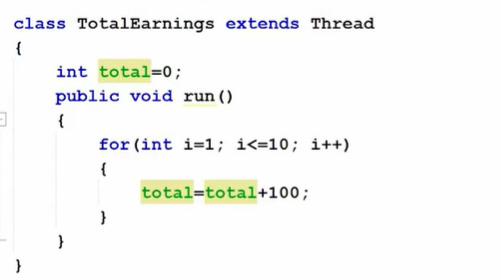
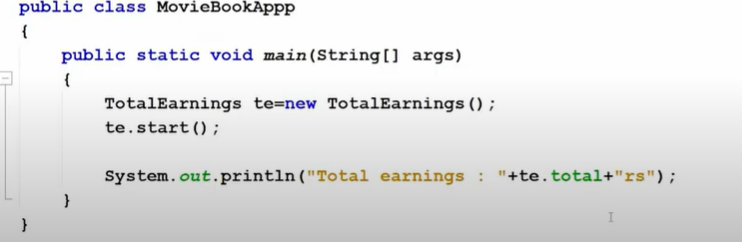
THIS METHODS ARE NOT PART OF THREAD CLASS .THIS ARE MEHODS UNDER OBJECT CLASS.

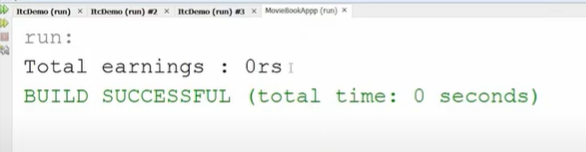




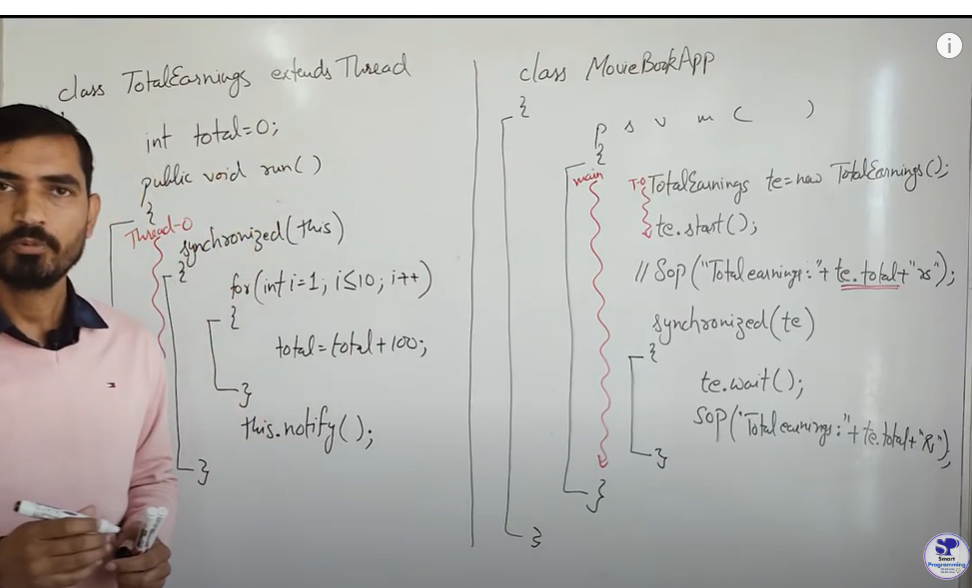
LIKE WE WANT TO GET TOTAL EARNINGS COLLECTED IN A MOVIE , SO WE USED TE.START(); FOR THAT BUT IT MAY HAPPEN THAT BEFORE TE.START() CALCULATES MAIN METHOD EXECUTE SOP AND A RANDOM VALUE WILL BE PRINTED ( UNEXPECTED BEHAVIOUR )



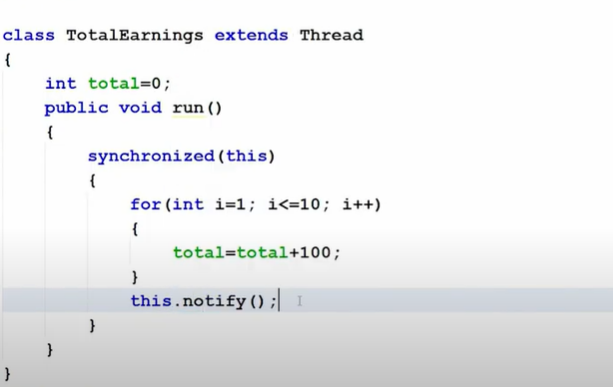


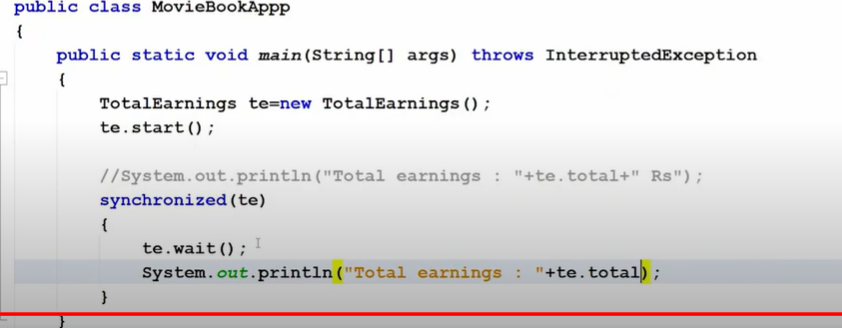


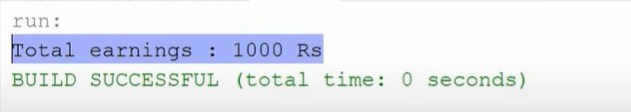
STEPS TO HANDLE : USE SYNCHRONIZE BLOCK INSIDE MAIN METHOD TO HOLD THE LOCK AND PASS THREAD AS REFERENCE , WAIT TILL ANY NOTIFICATION IS NOT RECEIVED.



CODE::







MovieBookAppp (Main Thread):

The main thread starts the TotalEarnings thread (te.start()).

The main thread then enters a synchronized(te) block, where it holds the lock on the TotalEarnings object (te).

The main thread calls te.wait() inside the synchronized block, which causes it to release the lock on te and go into a waiting state.

The main thread waits until it gets notified by the TotalEarnings thread.

TotalEarnings (Worker Thread):

The TotalEarnings thread starts executing its run() method.

The thread enters a synchronized(this) block, meaning it acquires the lock on itself (this refers to the TotalEarnings object, i.e., te).

The loop runs 10 times, adding 100 to the total variable on each iteration.

After the loop, the TotalEarnings thread calls notify(), which wakes up the waiting main thread.

Once the TotalEarnings thread finishes execution, it releases the lock on te.