Big...

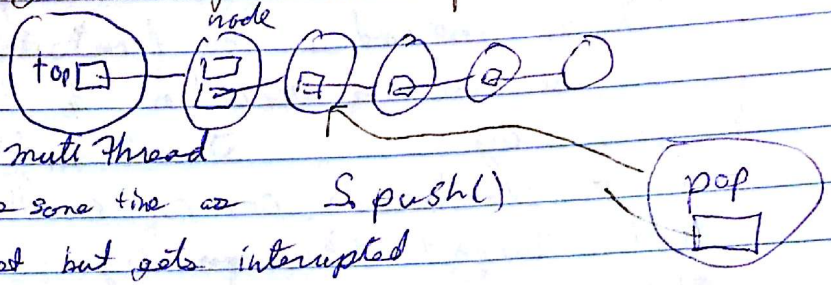
Cp213 Lecture Start Nov 18th 2016 CE

Threads are sometimes interupted and won't run

↳ Use **Synchronized** keyword on method

↳ stops more than one Thread from Calling Increment & "entering Critical Area" by locking the object until first thread finishes

Example Stack



Say the stack is multi Thread

S.pop() happens same time as S.push()

if pop goes first but gets interupted

push goes completely - Finishes adding it's node

then pop finishes removing the original node it was removing And the pushed node

Only methods that change the stack need to be synchronized

S = 2 = 2

S = 2 = 1

If only one piece of a method needs to be Synchronized Can create a **Synchronized Block** `Synchronized (this) {`

`// the code you want Synchronized`

`}`

Cp213 Lecture Start Nov 21st 2016 CE

Synchronization stops a thread entering the critical area until another Thread leaves (preserves Atomization)

Wait() → object Value → 0

↳ waits until another thread Updates Value

NotifyAll(); → wakes up all Threads

→ you Can use if or while loop outside of Wait() however it will wake up at the wait stmt so while is profered

Bonus marks For thread use in Assign 4