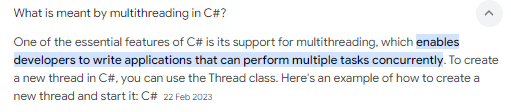
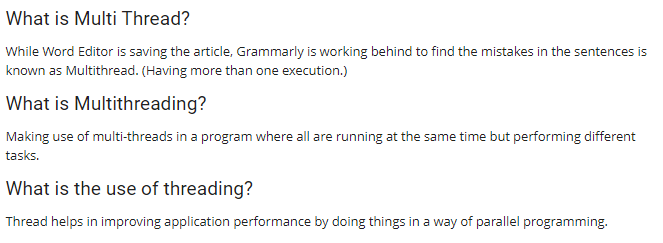
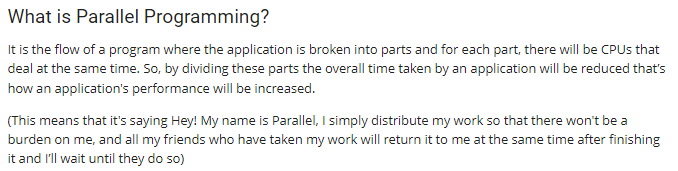
**Asynchronous Programming C#**







Threads & Tasks are two different ways for doing multithreading only

Thread is a general programming concept for all pgm languages.

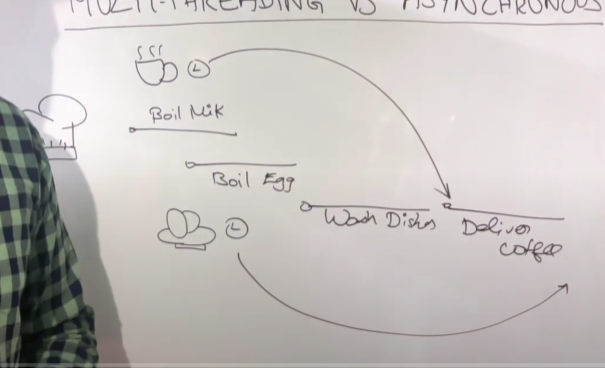
Microsoft created Task in .NET to simplify the use of threads or Tasks are like a wrapper over Threads and internally Tasks uses threads only.

**MultiThreading & Asynchronous Programming**

Both are different from each other and each have their own objective.

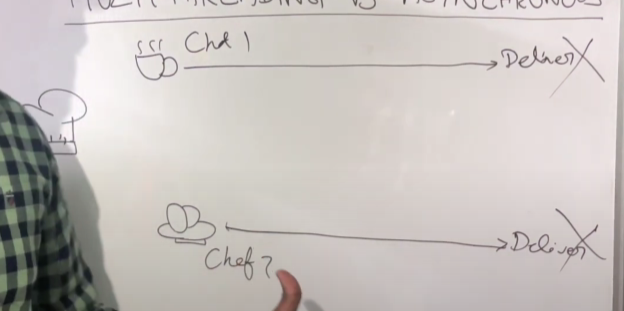
Eg: Handling two orders in Hotel.

In Async pgm, chef starts with first order and doesn’t wait for it to complete and starts the 2nd one also. Once the first order completes it receives a callback to chef and similarly for 2nd order. The chef was not ideal for any time.



MultiThread: Here upon 2 orders, check will assign these two orders to another two chefs (thread 1 & 2). Chef 1 do the work and deliver , similarly chef2.

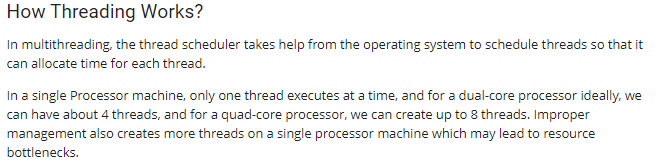
Once job is done, main chef will kill the new 2 chefs (kill the threads.)

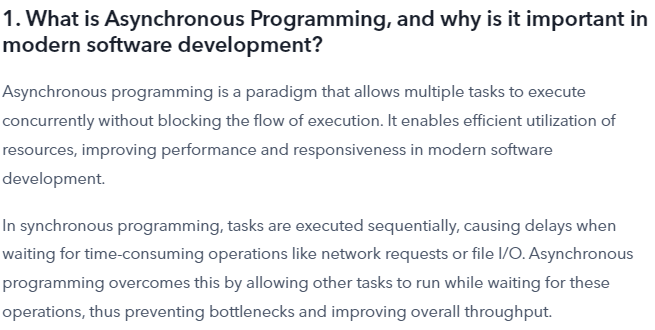
;

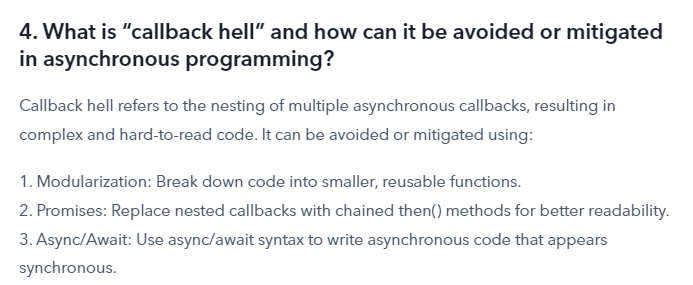
ThreadPool is used for managing all the threads and use it for different purposes.

Multithreading – all about managing the workers threads.

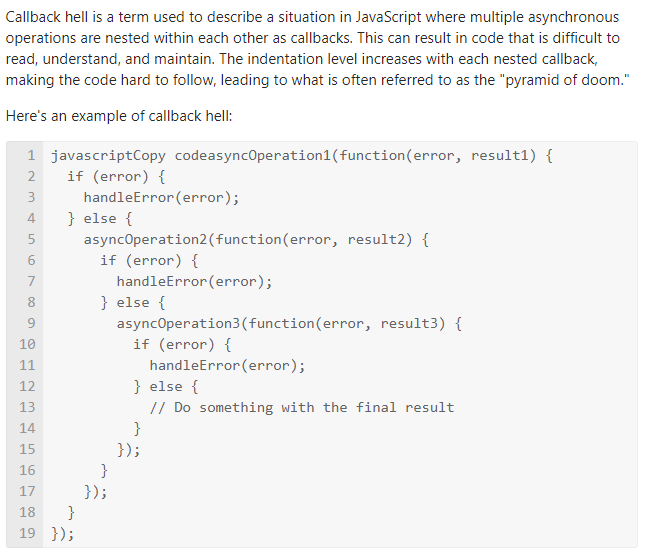
Async – managing the tasks, efficiently manage the task.

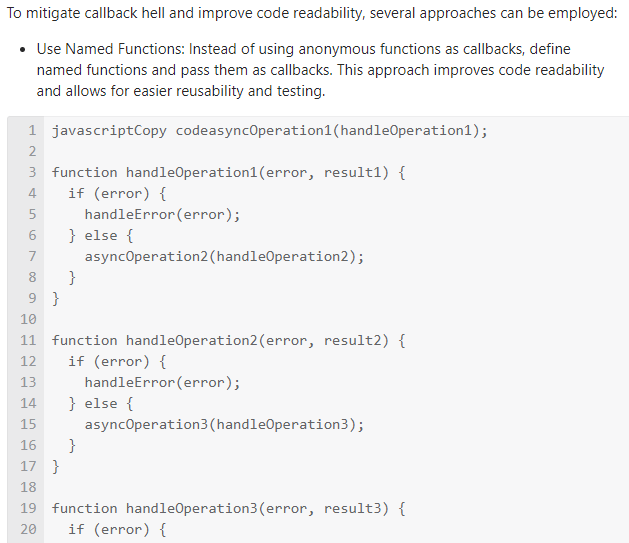


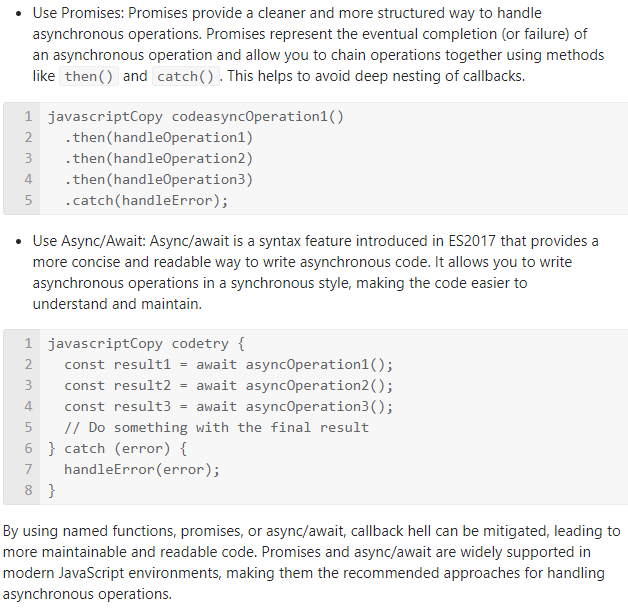


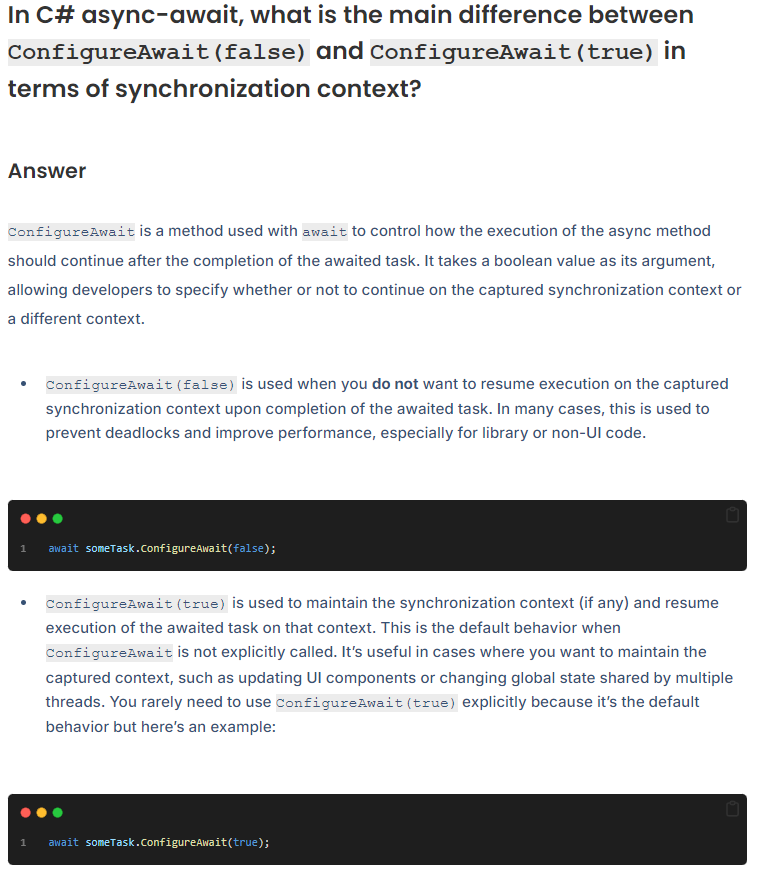


**CALL BACK HELL**









ConfigureAwait(false) tells it that it does not need the context, so the code can be run anywhere. It could be any thread that runs it. If the await is followed by a code that accesses the UI, the task should be appended with . ConfigureAwait(true)

