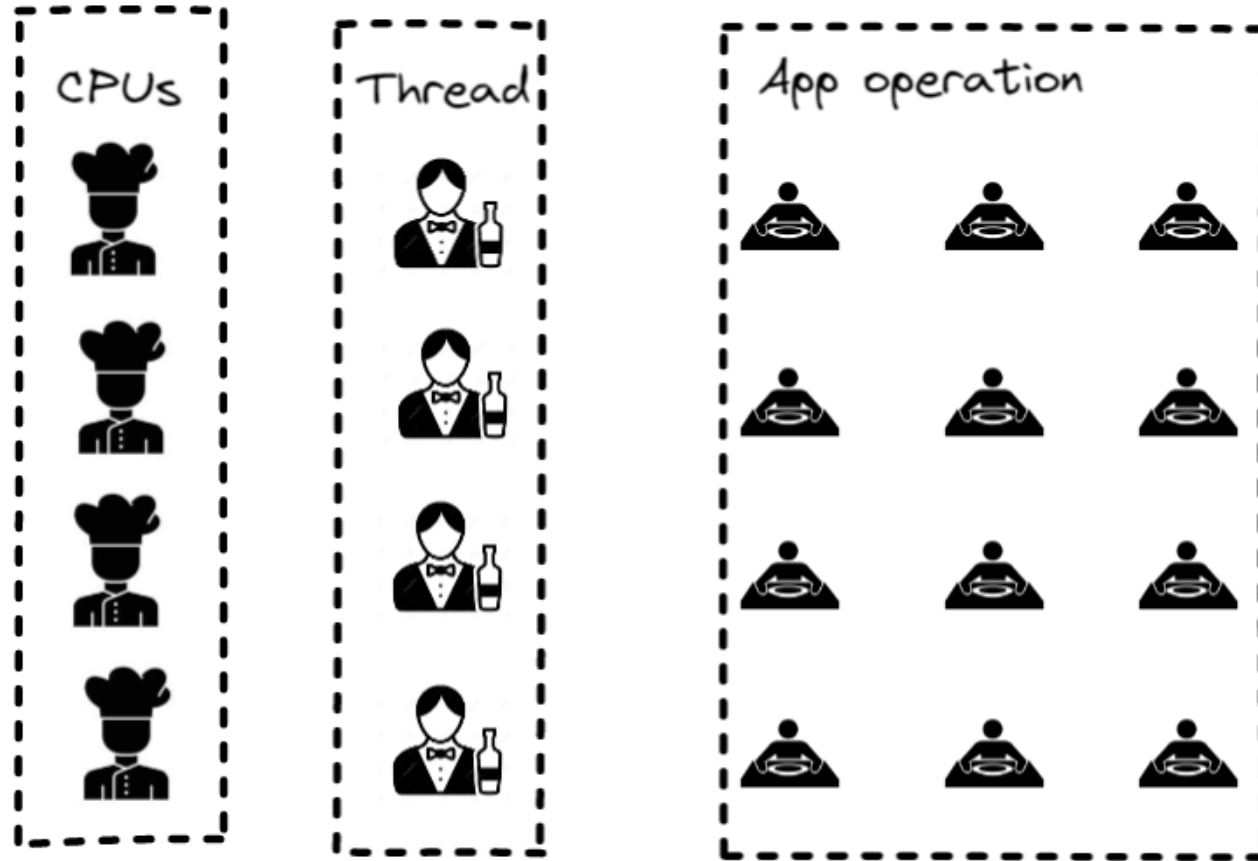


# Asynchronous programming & Working with persisted data

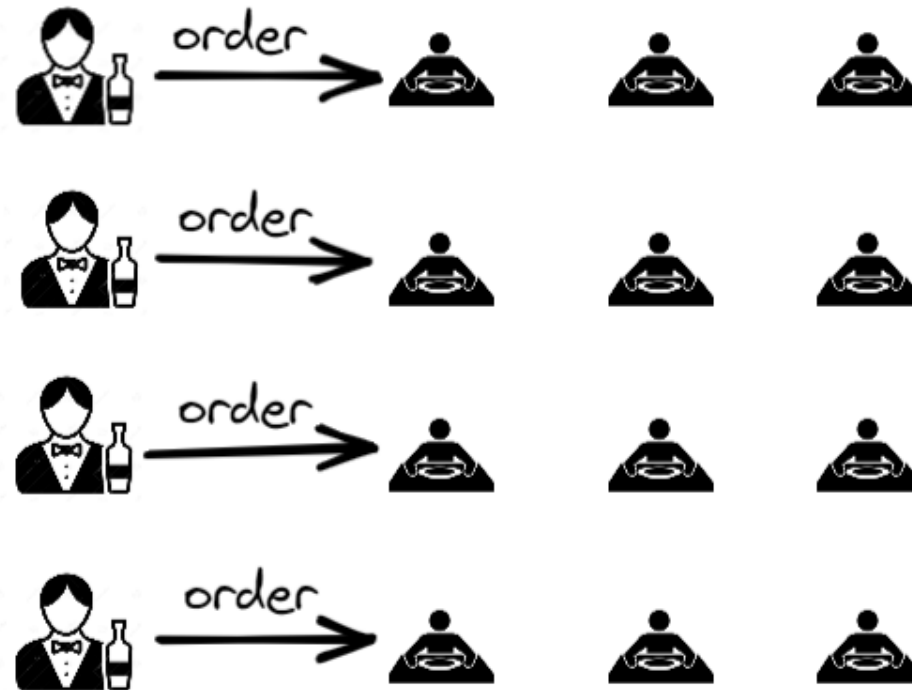
# What is asynchronous programming?

- A way of executing programming code in a thread without having to wait task to finish
- Idea of asynchronous programming is to divide our logic into awaitable tasks
- Benefits: better responsiveness of the app, better performance, avoiding bottlenecks

# Example



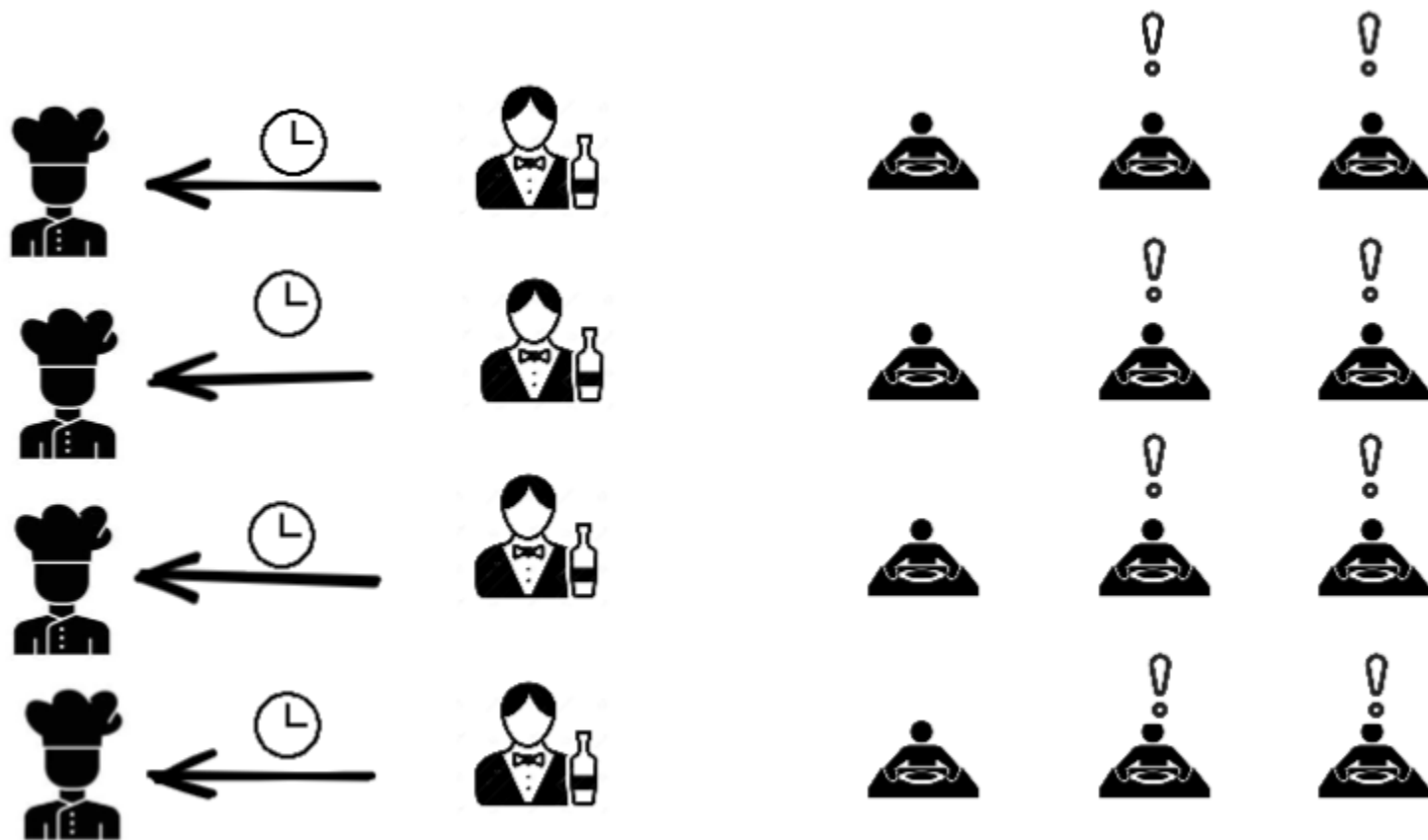
# Example



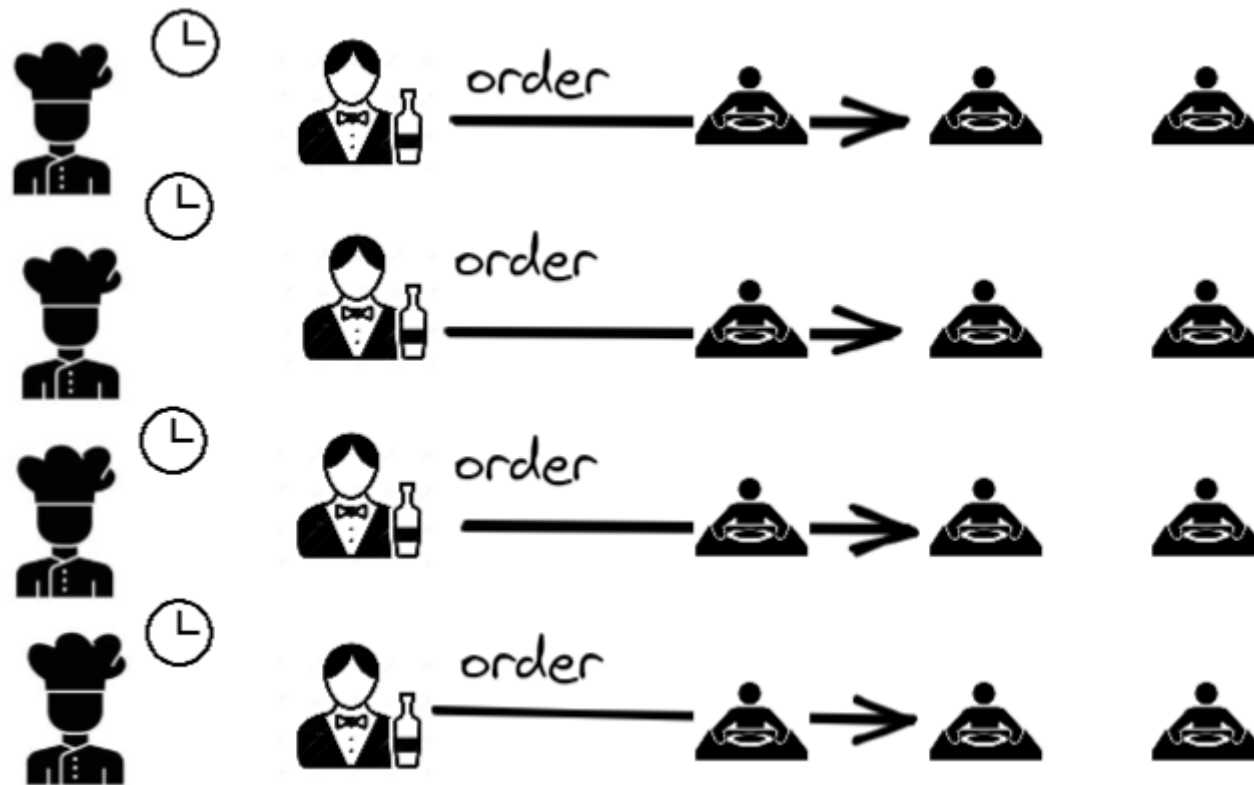
# Example



# Synchronous



# Asynchronous



# Intro to Async, Await and Tasks

- Asynchronous programming in C# can be accomplished by using Task based asynchronous patterns
- Async/Await enable us to see asynchronous code as synchronous (code is much cleaner and easier to read)
- It is an abstraction over asynchronous code
- An asynchronous operation occurs in parallel and relieves the calling thread of the work



# Async & Await Demo

# Things to remember

- **await** converts **Task<T>** to **<T>** (**Task** to **void**)
- Using **await** means that awaited method has to change signature to **async Task<T>**
- Never create an method **async void** but always **async Task<T>**
- You will get warning if you are calling **async Task<T>** method without using **async**
- Async all the way (to the top)

# NuGet

- NuGet is the package manager for .NET.
- The NuGet Gallery is the central package repository used by all package authors and consumers

# NuGet Demo

# Dapper

- Tool that enables us to easily create, read, update and delete data
- It provide us with easy way of mapping database data to objects
- It has two main methods `QueryAsync<T>` (Get), `ExecuteAsync` (Insert, Update, Delete)

# Dapper Demo

# Let`s code!

Go to `/week5/example.md`