Asynchronous Programming

We could define asynchronous programming as the way of executing programming code in a thread without having to wait for an I/O-bound or CPU-bound task to finish.

Asynchronous programming in C# is a programming paradigm that allows your code to perform tasks concurrently without blocking the main execution thread. This is particularly useful when you need to perform time-consuming operations, such as I/O-bound tasks (e.g., reading from a file, making a web request), without freezing the user interface or blocking other operations.

**Key Concepts in Asynchronous Programming in C#:**

1. **async and await Keywords**:
   * **async**: Marks a method as asynchronous, allowing it to contain await statements. The method returns a Task or Task<T>.
   * **await**: Suspends the execution of the method until the awaited task completes, without blocking the thread. The method continues executing after the awaited task is done.
2. **Task and Task<T>**:
   * **Task**: Represents an ongoing or future operation. It’s a way to manage asynchronous operations and get notified when they complete.
   * **Task<T>**: Represents an asynchronous operation that returns a value of type T.