Chapter 1 Concurrency an overview.

* **Introduction to concurrency**

1. Concurrency is doing more that one thing at the same time
2. Multithreading is a form of concurrency that uses multiple threads
3. Multithreading lives on the thread pool, a useful place to queue work
4. Thread pool enables another form of concurrency: Parallel processing
5. Parallel processing is one type of multithreading, multithreading is one type of concurrency
6. Async programming a form of concurrency that uses promises [features] or call-backs
7. Another form of concurrency is reactive programming.

* **Introduction to Async Programming**

1. Has two benefits for GUI free the main UI Thread and for server applications free the requests threads
2. When async keyword added to methods it performs double purposes

It allows await keyword within that method and signals the compiler to generate a state machine for that method

1. Reactive programming related to asynchronous events while async programming based on asynchronous operations.
2. An async method start execution synchronously just like any other method, the await keyword performs asynchronous wait. First check if the operation is complete if yes then it completes executing asynchronously, otherwise it will pause and return incomplete task.
3. When you wait a task, a context is captured when the await decides to pause the method, this is the current “SynchronizationContext”, then method will continue executing “when resuming” on that captured context usually the UI context.
4. Asp.net classis uses “Request Context”, but .net core uses “Thread Pool “Context.
5. Use configureAwait method to change the default behaviour
6. Avoid use task.Wait() could cause a DeakLock if method called from UI Thread or asp.net classic context.

* **Introduction to Parallel programming**

1. Use parallel.ForEach , or PLINQ values.AsParallel , parallel.For
2. Use parallel.Invoke()
3. Use AggregateException to handle multiple exceptions when running parallel tasks.

* **Introduction to Reactive Programming**

1. Subscriptions behave differently with hot and cold observables.
2. LINQ to entity and Objects is Pull model , however LINQ to events “System.Reactive” uses push model where events arrive and travel through the query.
3. Reactive programming is based on the notion of observables streams.

* **Introduction to Dataflows**

1. TPL Dataflow is a mix of async and parallel technologies.
2. TPL Dataflow is commonly used as a pipeline
3. Dataflow block is the building block of dataflow [could be source block (producing) or target block (receiving)
4. Use AggregateException.Flatten to handle multiple exceptions caused by the mesh or pipeline.

* **Introduction to Multithreaded programming**

1. A thread is an independent executor.
2. New use Thread, only if you need STA thread for COM interop.

* **Collection of Concurrent Applications**

1. Use Concurrent collections and immutable collections.

* **Summary of Key technologies**

System.Reactive

System. Collection. Immutable

System. Threading. Tasks. Dataflow

Chapter 2 Async Basics

* **Introduction to concurrency**