Functional Programming

Introduction(Video 1):

-Types of programming:

1- Declarative Programming

2- Imperative Programming

-Types of Declarative programming :

1- Functional Programming

2-Reactive Programming

3- Domain Specific Language

4- TPL dataflow (some sort of parallel programming)

- you write declarative daily : when you write sql statement the sql server creates execution plan for you , in this case sql statement is declarative and execution plan is imperative

- imperative means the exact computation you need to finish the task, declarative means high level statements.

- OOP is based on encapsulation which encapsulates data and logic in a single unit, Functional programming is based on that every single function doesnt take input from global scope or write output to a global scope but its a singe individual unit that is easy to test and if all functions are correct then all the functions pipeline will be correct (take input → calculate → output).

- FP is easier in work division and easier to test than OOP and encapsulation.

-LINQ is a pipeline of functions in C#.

Video 2 Check Code Example1.cpp

Video 3

- pure function is the function that doesn't read data from its surroundings or write data to its surroundings , it depends only on the single input ex: AddOne , we always try to make our functions pure to use functional programming

- Higher order function : means to take or return function inside another function.

- Check Example2.cpp

Video 4

-Check Example3.cpp

Video 5

- Function composition means if you have y= f(x) and g = f(y) then you can compose both functions and have g(y) which is f(f(x))

- Function composition helps to have shorter pipeline as we can compose related functions together in one composed function

- Function composition is very useful check the end of the video

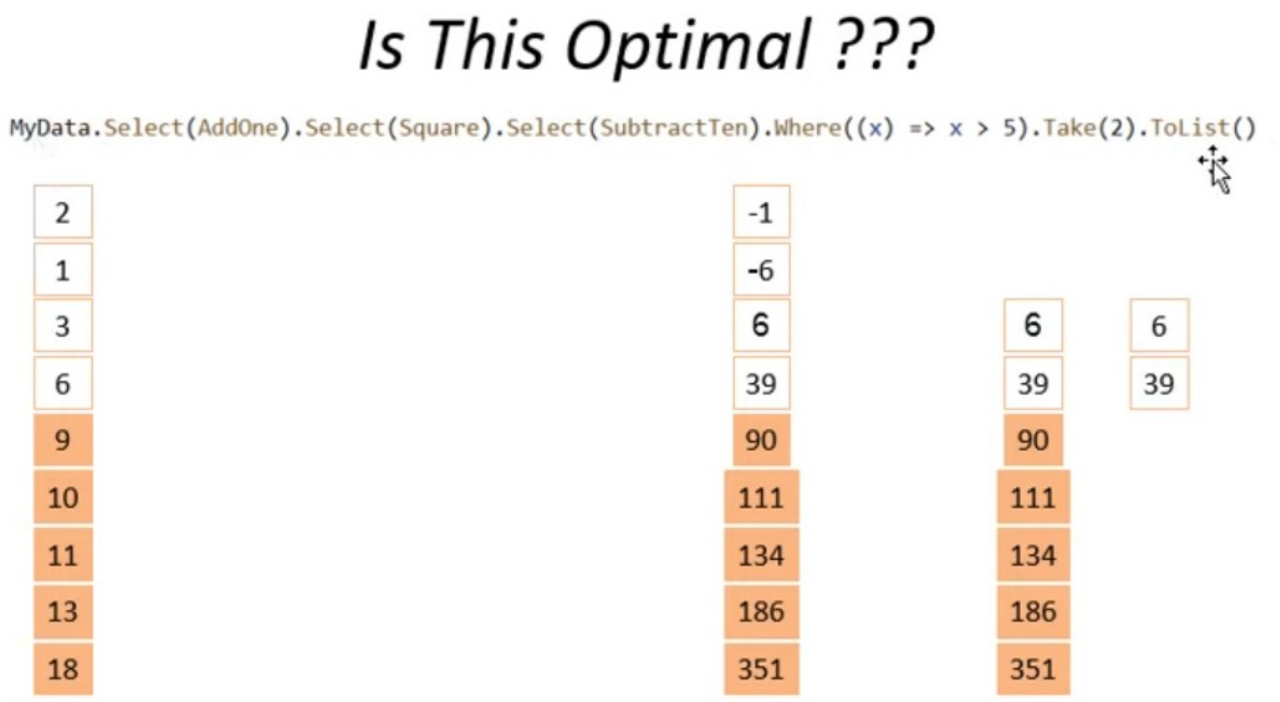
Function Closure

- normal functions are like some of instruction of code , function closure are instructions + memory

, memory is added to function to act based on its memory

How LINQ Works

- its not mydata that pushes the data to the pipeline as this will lead to extra computation,its ToList() who polls the data from the pipeline until it reaches the 2 elements its need



- Enumerator is a yield return so it helps to poll, enumerators are iterators

- I think if the pipeline have sorting it will have to push all the data

- check session code it have implementation that is same as LINQ