Functional Programming via C#

Dallas C# SIG February 5, 2015

Nik Clarkson (CaptainArkansas)

What is Functional Programming?

Higher Order Functions

A function as a parameter

```
int SomeFunction(Function myFunction)
{
      return myFunction(10);
}
```

A function as the return value

```
Function SomeFunction(int someValue)
{
    return x => x + someValue;
}
```

both

```
Function SomeFunction(Function myFunc)
{
    return x => myFunc(x);
}
```

Higher Order Function Examples

Referential Transparency

Functions constructing functions!

Common Higher Order Functions

Map(List, Func) : List

Applies a function to each element of a list

Fold(List, Func<Bool>) : List

Returns only elements from the list that satisfies the Boolean function

A.K.A. - Reduce

Reduce(List<T>, Func) : V

Reduces a list of elements to a single value

Filter

Fold (Reduce)

Map, Filter, & Fold in LINQ

List Comprehensions

Recursion

Activating Skynet

The more you know!

http://learnyouahaskell.com/chapters

http://fsharpforfunandprofit.com/

Why FP Matters - http://www.cse.chalmers.se/~rjmh/Papers/whyfp.html

https://projecteuler.net/

http://www.meetup.com/Dallas-Functional-Programmers/

Questions?

Explorer’s League

Contact

- Nik.Clarkson@improvingenterprises.com
- @CaptainArkansas
- https://github.com/armadilloNik/FunctionalCSharpSig