"There are two ways of constructing software. One way is to make it so simple, that there are obviously no deficiencies, The other way is to make it so complicated that there are no obvious deficiencies."

Goal: Obviously No Deficiencies

Readable

Reuseable
Goal: Obviously No Deficiencies

Modifiable

Predictable

Goal: Obviously No Deficiencies

Checkable

Yes, but how?
Goal: Obviously No Deficiencies

Functional Programming (?)

No Assignment.
No Mutation.
No Loops.

So, Who Uses FP ?

PL Researchers.

So, Who Uses FP ?

Functional Programming ?

Readable
Reusable
Modifiable
Predictable
Checkable
Parallelizable

So, Who Uses FP ?

Functional Programming ?

Readable
Reusable
Modifiable
Predictable
Checkable
Parallelizable
So, Who Uses FP?

Google
MapReduce
Facebook
Erlang

So, Who Uses FP?

Microsoft
F#
Twitter
Scala
So, Who Uses FP?

Wall Street

CSE 230: Grading

[10%] Class Participation
[60%] Pair Assignments
[30%] Take-home Final

CSE 230: Outline

1. FP & Abstraction
   - Readable
   - Reusable
   - Modifiable
   - Predictable
   - Checkable

2. Types & Analysis