**Note particularly where he talks about functional programming and its evolution.**

For dynamic languages:

There are undefined regions, and they will run/may run in an unexpected way.

For the regions that are supposed to work, the programmers believe that they should work as expected.

For static languages:

The regions that are outside of the regions that the programmers assume should work, they are wrong, or impossible. In addition to that, there are regions inside the defined region that are undefined, which would require the static programmers to run tests and see what the behavior is for these regions.

Inference is required for both dynamic and static languages.

Null exist in dynamic languages but not static languages

2015:

Null is needed for programming because people believe it is the only way to represent nothingness.

2010:

Functional programming is slow because people believe there is no faster immutable types.

2000:

Garbage collection is never going to be practical for real world software because people believe that there is no better algorithm exist for GC

Type: Correctness comes from categories

Test: Correctness comes from examples.

**Note particularly where he mentions Rich Hickey, whose invention is the subject of this course.😎**

Rich Hickey was the one who invented Clojure.