

## What is Clojure

Clojure is a variant of LISP, one of the oldest computer languages that is still in active use today. It is multi-paradigm, supporting mutable and immutable data (more on that below), functional, object-oriented, and even procedural programming.

It is available for free, and runs on Linux, Mac, and Windows. It targets the Java Virtual Machine, and is able to interoperate with any Java program.

It is (in)famous for a lack of syntax; parenthesis are used for nearly everything, and the position of a sub-expression within another determines its intent rather than keywords. For example, instead of

```
if (x>4) printf("Hi");
    else printf("%d",cos(x)) ;
```

we would use

```
(if (> x 4) (printf "hi") (printf "~a" (cos x)))
```

The simplicity of the language allows for many advantages that become apparent later on. Clojure, in particular, has built-in support for some very modern data-structures, making it very easy to write complex programs quickly. One of my former students, acting on my recommendation to learn this language, emailed me to say that this was the most fun he had ever had programming.

## Why use it for Data Structures?

The world of computer science is changing. While computers have been able to multitask for quite some time, they typically had done this using a single CPU. As a result, data structures and algorithms usually assumed one CPU core manipulating the data. Until about ten years ago, only large institutions could afford to have multi-CPU machines.

Today, multicore CPUs are common; it is hard to buy a machine that does not have at least two cores, and even smart phones have two or four cores on board. Manipulating data while multiple cores are running simultaneously requires a different mindset than for a single core.

One of the major changes of mindset is a shift from *mutable* data structures (i.e, you can update a structure after you have created it) to *immutable* structures, which allow update only by recreating parts of the affected structure. The ability to preserve history and to rely on the integrity of the data makes it much easier to make full use of parallelism, and this style is rapidly being adopted in industry.

In order to be well prepared, a programmer will need to know how to manipulate data structures in both their mutable and immutable versions.

Clojure helps us here because it enables us to use several paradigms of programming. It has a more expressive object system than newer languages like Java. But unlike Java, it is not limited to object oriented style; it also allows us to express solutions in functional style, which tends to favor immutable data, and is especially useful in multi-core environments.

By using multiple paradigms, we hope to help students gain flexibility in approaching programming problems, and to have an appreciation for tradeoffs in structuring data that their peers will not gain from using languages that lock them into an object-oriented way of thinking.

Because there is a lot of excitement about Clojure, there are *many* resources on the web—documentation, libraries, and discussion groups—for those trying to learn Clojure or to solve problems.

## Isn't it hard to learn LISP or Clojure?

It is not necessarily more difficult than other languages to learn. There is a learning curve simply because it does not look like other languages, but most students get used to it quickly. After that, the syntax is quite minimal; a programmer only needs to be sure to balance the parentheses.

## What about other universities?

There is a wide range of languages used to teach data structures. C++ and Java are common, but it's possible to find languages such as Haskell, Python, and other dialects Lisp/Scheme. There are several universities that teach data structures purely as theory, with no programming other than pseudo code.

Racket, another dialect of LISP, is used at universities such as San Jose State, Brown, Worcester Polytechnic Institute, Utah, North Eastern University, and Adelphi. The language sees a range of use, from CS1 to classes in programming languages.

From a teacher's perspective, I can offer one observation: when teaching a junior to senior level course, the students who had introductory programming based on Scheme-like languages performed on average a letter grade higher than those who had a different language.

## Is it applicable to industry?

There is still active development in the LISP world. Clojure, being a JVM-based language, is causing a lot of excitement. As it was released for the first time in 2009, it's too early to tell how much industry adoption it will have.

However, early anecdote report a lot of use, especially in software that was already written in Java, since it is fully interoperable with it. There are also versions of Clojure that target .Net, JavaScript, and Python.

Some industry success stories include Paul Graham's Beating the Averages, about how he used LISP to write the first interactive web service, and was able to stay ahead of his competitors indefinitely. Dialects of LISP are used by many companies, including American Express, Orbitz, PricewaterhouseCoopers, and Nichimen Graphics.

Finally, we note that LISP concepts are increasingly being incorporated into non-lisp languages that have lacked them: the most notable is that the lambda construct is being added to Java 8. In addition to Clojure, the JVM language SCALA, which is inspired by HASKELL, emphasizes the use of immutable data structures.

## How can I prepare for the class?

There are three practical steps a student could take to prepare for this class. All of them are optional, as we assume that nobody has used this language before and take our time introducing it.

First, get the Clojure compiler. The fastest way to do this is to install a program called `lein`, or Leiningen. You can get this from <http://leiningen.org/>.

There are many Clojure tutorials online, including videos on Youtube. We are in the process of writing our own tutorial as well. Contact me, and let me know about your computing background, and I will be able to recommend some tutorials for you.