Course Introduction

Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science
Table of Contents

Introduction and Logistics
Objectives

Clojure
Welcome to CS 296-25!

Topics for discussion:
- Logistics — instructor, grades, course objectives, lecture format
- Clojure
Me!

Name  Mattox Beckman

History  PhD, Fall 2003, University of Illinois at Urbana-Champaign
  Lecturer 2013–2015 Illinois Institute of Technology

Research Areas  Programming Languages, Mathematical Foundations of Computer Science, CS Education

Specialty  Partial Evaluation, Functional Programming

Professional Interests  Teaching; Partial Evaluation; Interpreters; Functional Programming; Semantics and Types; Category Theory

Personal Interests  Cooking; Go (Baduk, Wei-Qi, Igo); Theology and Philosophy; Evolution; Meditation; Kerbal Space Program; Home-brewing; ... and many many more ...
Contact Info

Instructor  Mattox Beckman

Best Contact  via email. I pretend to use inbox zero, but not on weekends.

Email Addresses  <mattox@illinois.edu>.

Office  2227 Siebel Center

Office Hours  12:00–14:00 F; 3034 ECEB
Assignment

- **Mini Machine Problems**
  - Small Clojure programs
  - Collectively worth 50%
  - Probably 4 of these.

- **Final Project**
  - Announced in a few weeks
  - Worth the other 50%

- You may collaborate with one other student.
## Grade Guarantees

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>93</td>
<td>A</td>
</tr>
<tr>
<td>90</td>
<td>A-</td>
</tr>
<tr>
<td>87</td>
<td>B+</td>
</tr>
<tr>
<td>83</td>
<td>B</td>
</tr>
<tr>
<td>80</td>
<td>B-</td>
</tr>
<tr>
<td>77</td>
<td>C+</td>
</tr>
<tr>
<td>73</td>
<td>C</td>
</tr>
<tr>
<td>70</td>
<td>C-</td>
</tr>
<tr>
<td>60</td>
<td>D</td>
</tr>
<tr>
<td>&lt;60</td>
<td>F</td>
</tr>
</tbody>
</table>

- This is just for formality—you are all “supposed” to get an A in this course.
- To get an A+ you will need to do an extended version of the final project.
Table of Contents

Introduction and Logistics
Objectives

Clojure
Quick Intro to **Clojure**

- The first thing about **Clojure**: parentheses!

<table>
<thead>
<tr>
<th>Feature</th>
<th>C++</th>
<th>Clojure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions</td>
<td>( f(x,y,z) )</td>
<td>((f x y z))</td>
</tr>
<tr>
<td>Arithmetic</td>
<td>( 2 \times x + 9 \times y + 5 )</td>
<td>((+ (\times 2 x) (\times 9 y) 5))</td>
</tr>
<tr>
<td>If</td>
<td>( \text{if} (x &gt; y) a; \text{else} b )</td>
<td>((\text{if} (&gt; x y) a b))</td>
</tr>
<tr>
<td>Array</td>
<td>( \text{int} a[3] = {1,2,3}; )</td>
<td>((\text{def} a [1 2 3]))</td>
</tr>
<tr>
<td>Function</td>
<td>( \text{int inc(int i) {} \ldots } )</td>
<td>((\text{defn inc [x] (+ x 1)}))</td>
</tr>
<tr>
<td>Lists</td>
<td>not built it</td>
<td>'(10 20 30)</td>
</tr>
</tbody>
</table>

- This will seem weird at first, but...
  - **Use a good editor!** and it will be easier to read.
  - There will be huge advantages later on. Code and data have the same form!

- Enough talk. Let’s do the activity.