<table>
<thead>
<tr>
<th>Fall 2020</th>
<th>Language</th>
<th>Assignment (due at end of week)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug 24–28</td>
<td>Forth</td>
<td></td>
</tr>
<tr>
<td>Aug 31–Sep 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sep 7–11 (Labor Day)</td>
<td>Forth</td>
<td></td>
</tr>
<tr>
<td>Sep 14–18</td>
<td>Standard ML</td>
<td></td>
</tr>
<tr>
<td>Sep 21–25</td>
<td>Standard ML</td>
<td></td>
</tr>
<tr>
<td>Sep 28–Oct 2</td>
<td>Standard ML</td>
<td></td>
</tr>
<tr>
<td>Oct 5–9</td>
<td>Simple interp</td>
<td>Standard ML</td>
</tr>
<tr>
<td>Oct 12–16 (Fall break)</td>
<td>Lisp interp</td>
<td>Simple interp</td>
</tr>
<tr>
<td>Oct 19–23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oct 26–30</td>
<td>Scheme interp</td>
<td>Lisp interp</td>
</tr>
<tr>
<td>Nov 2–6</td>
<td>Prolog</td>
<td></td>
</tr>
<tr>
<td>Nov 9–13</td>
<td>Prolog</td>
<td></td>
</tr>
<tr>
<td>Nov 16–20</td>
<td>Prolog</td>
<td></td>
</tr>
<tr>
<td>Nov 23–27 (Thanksgiving)</td>
<td>TBD</td>
<td>Prolog</td>
</tr>
</tbody>
</table>

Changes to the schedule will be announced in class.

Resources

- Syllabus
- Examples from class
- Setting up ssh to connect to cs3520.cs.dixie.edu without typing a password:
  - Written instructions
  - Screencast demo (note, the written instructions are slightly simpler—open that page and follow along while you watch the screencast).

Languages

- Forth
  - Learn X in Y Minutes: Forth
  - Easy Forth
  - Starting Forth
  - jonesforth (assembly part)
  - jonesforth (forth part)
- Standard ML slides
- Prolog slides part 1 (first look, rules, operators, lists)
- Prolog slides part 2 (second look, unification, execution model, adventure game)
- Prolog slides part 3 (cost models)
- Prolog slides part 4 (third look, numeric computation, knapsack, 8-queens)
- Lua 5.1 Reference Manual
- Learn X in Y minutes: Lua
- Lua: Passing a Language through the Eye of a Needle
- Language shootout size vs speed

Assignments

See the Canvas listings for assignments and due dates. All homework is submitted using CodeGrinder unless otherwise noted.

Final project languages

In place of a final exam, each student will learn one additional language, write some code in that language, and present it to the rest of the class. Here are a few potential choices:
- Factor
- Smalltalk
- Haskell
- OCaml or F#
- Clojure
- Common Lisp
- Rust
- Perl
- Erlang or Elixir
- J
- Tcl