CSE410 aka CSE306
Software Quality

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall

http://www.cse.buffalo.edu/faculty/alphonce/SP17/CSE410
https://piazza.com/class/iyb933z3aro2p
**plan**
(drop LEX 9/10, EXP 3/4)

<table>
<thead>
<tr>
<th></th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/17-4/21</td>
<td><strong>✓</strong></td>
<td>(open)</td>
<td><strong>✓</strong></td>
</tr>
<tr>
<td>4/24-4/28</td>
<td><strong>✓</strong></td>
<td>LEX&amp;8</td>
<td><strong>✓</strong></td>
</tr>
<tr>
<td>5/1-5/5</td>
<td>developing and debugging in a dynamic language</td>
<td><strong>✓</strong></td>
<td>developing and debugging in a dynamic language</td>
</tr>
<tr>
<td>5/8-5/12</td>
<td>Chat Client dynamic language dev &amp; deb</td>
<td><strong>✓</strong></td>
<td>Chat Server? dynamic language dev &amp; deb</td>
</tr>
</tbody>
</table>

POST PROJECT
Team formation

- teama: Ed
- teamb: Stan
- teamc: Heeba
- teamd: Dan
- teame: Valencia
Today's goal: build a chat client

- I have a chat server already built.
- It responds to these messages:

  `{From, {say, Text}}`
  `{From, join}`
  `{From, leave}`
Today's goal: build a chat client

- I have a chat server already built.
- It sends this message:

  \{say, From, Text\}
{From, {say, Text}}

- **From** - the server who sent the message
- **Text** - the text (as an atom) to broadcast to the rest of the chat
{From, join}

- From - the server who sent the message
- joins the current chat (no duplicates possible)
{From, leave}

- From - the server who sent the message
- leaves the current chat (if not in chat then no change)
How many processes in the client?
How many processes in the client?
How many processes in the client?
How many processes in the client?
How many processes in the client?
How many processes in the client?
How many processes in the client?
Capturing Pid of local process

```erlang
Pid = spawn(fun() -> listener() end)
```

Write output

```erlang
io:format("Welcome to chat!\n"),
```

Read input

```erlang
{ok,Command} = io:read("chat> "),
```

How to communicate with process registered on another machine

```erlang
Server = {chat_server, 'chat@timberlake.cse.buffalo.edu'}
Server ! {Pid, join}
```