Get to know each other!

Which do you like better: *vanilla* or *chocolate* ice cream sandwiches?
Announcements

Course website is www.teach.cs.toronto.edu/~csc148h/fall/

Course email address is csc148-2018-09@cs.toronto.edu

Ramp-up materials posted

Jaisie is holding extra office hours today 3-6pm in BA2230

Lab room assignments have been posted
How to use active learning lectures

Review the prep before class so that it’s fresh in your mind.
Take your own notes – don’t copy things down blindly.
Do the worksheets – don’t wait for us to take it up.  
(often we won’t)
Switch between working by yourself and with your classmates.
Ask questions! Your TAs are here to help.
Complete and/or review lecture materials after class.
<table>
<thead>
<tr>
<th>Key terms and phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
</tr>
<tr>
<td>id/type/value</td>
</tr>
<tr>
<td>variable</td>
</tr>
<tr>
<td>refers to</td>
</tr>
<tr>
<td>assignment statement</td>
</tr>
<tr>
<td>alias</td>
</tr>
<tr>
<td>immutable</td>
</tr>
<tr>
<td>mutable</td>
</tr>
<tr>
<td>mutate</td>
</tr>
<tr>
<td>re-assign</td>
</tr>
</tbody>
</table>
for loop gotcha

```python
lst = [1, 2, 3]

for item in lst:
    item = item + 1

print(lst)
```

*Do not assign to the loop variable in a for loop.*
The Python Memory Model: Functions

WHAT HAPPENS WHEN WE CALL A FUNCTION?
Example 1

def mess_about(n: int, s: str) -> None:
    message = s * n
    s = message

if __name__ == '__main__':
    count = 13
    word = 'nonsense'
    mess_about(count, word)
def mess_about(n: int, s: str) -> None:
    message = s * n
    s = message

if __name__ == '__main__':
    count = 13
    word = 'nonsense'
    mess_about(count, word)
Example 1

def mess_about(n: int, s: str) -> None:
    message = s * n
    s = message

if __name__ == '__main__':
    count = 13
    word = 'nonsense'
    mess_about(count, word)
Example 1

def mess_about(n: int, s: str) -> None:
    message = s * n
    s = message

if __name__ == '__main__':
    count = 13
    word = 'nonsense'
    mess_about(count, word)
Example 1

def mess_about(n: int, s: str) -> None:
    message = s * n
    s = message

if __name__ == '__main__':
    count = 13
    word = 'nonsense'
    mess_about(count, word)
New terminology

**frame**

data about a function call (including local variables)

**call stack**
a collection of the frames of the functions that are currently running
A complete memory model shows...

On the left, the call stack, whose frames show the program’s variables and which ids they store

On the right, the object space, where all objects are shown (id, type, value)
Function calls and the memory model

When a function is called, its parameters become *aliases* of the arguments passed to the function call.

If you understand this, you can do the next worksheet!
Function gotcha

def f(x, ...):
    x = 10
    ...
    # Other code

    if __name__ == '__main__':
        f(148, ...)

Do not assign to a function parameter (without using it first).
Your first lab is this Thursday!

Review lab policies and rooms on the course syllabus.

*Before* the first lab, go to your lab room and login to a Teaching Lab machine to reset your password.

Labs are graded based on **attendance**.
- Show up on time
- Work hard on the lab exercises
- Complete the three-stage quiz at the end of the lab

For the first lab, you *must* work with a partner.