

Assignment 0 is due Friday

I hope it is going well. Some tips / reminders:

- If you haven't handed any version of it in yet, do that *now*.
 - This removes the risk of battling the clock on the due date
 - You can hand in a better version later.
- Don't neglect writing your own tests.
- Before your final submission:
 - Run pyTA, re-run your tests, remove TODOs.
- After your final submission: Run the MarkUs tests again.

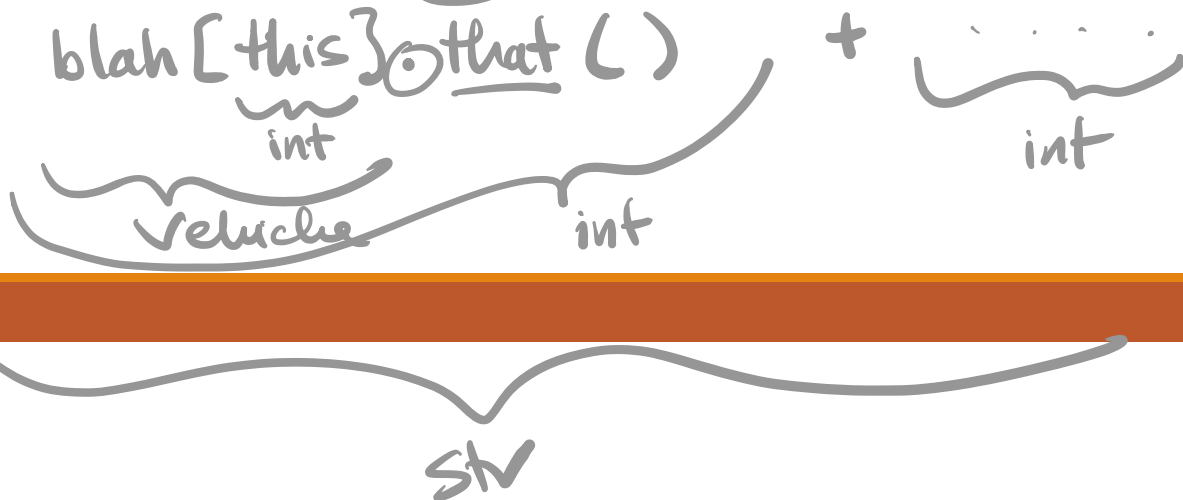
About solving bugs

Use the evidence in front of you.

Make more evidence by writing and running more tests.

Be skeptical about your own code.

Tip: always know what type of object you have in your hand.



Academic Integrity

Now is the time when you may be tempted to ask for inappropriate help.

Or to give inappropriate help.

It's not worth the risk.



PROGRAM EXPLORATION DAYS

FIND YOUR PATH

February 8: Computer Science

February 9: Math & Physical Sciences

February 15: Life Sciences

February 16: Social Sciences & Humanities

Register: uoft.me/programexploration



UNIVERSITY OF TORONTO
FACULTY OF ARTS & SCIENCE



TECHNOLOGY LEADERSHIP INITIATIVE

For undergraduate students interested in the challenge of a program that offers industry-integrated classroom learning, tailored leadership training for technologists, and industry internships, the Technology Leadership Initiative is a great way to accelerate your professional career.

[LEARN ABOUT THE PROGRAM](#)

We are now accepting applications for the fourth cohort of the Technology Leadership Initiative - ensure you submit your application by March 21, 2022.

Accelerate Your Success

The Technology Leadership initiative combines the academic excellence of the Computer Science program at the University of Toronto with a leadership curriculum taught in workshops, simulated in the classroom, and experienced through summer internships. Students develop leadership skills through three learning verticals.



Industry-Integrated Learning

Explore solutions to real-world problems woven into our academic curriculum



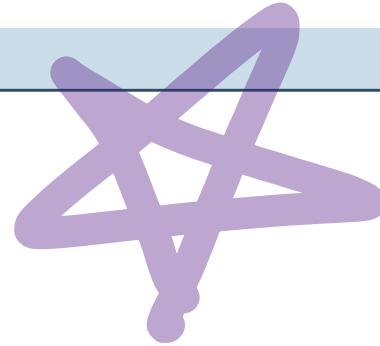
Leadership Skills Development

Learn to implement your vision in team-based settings guided by leading industry experts



Internships

Apply what you have learned while engaging in work placements with diverse companies



TLI information session:
February 4th, 2022 at 12-1pm EST

RSVP required

↑ sign-up.

```
Linked List
=====
CSC448 Winter 2019
Department of Computer Science,
University of Toronto

Module Description
=====
This module contains the code for a linked list implementation with two classes,
LinkedList and _Node.

Code template we can use:

curr = self._first
while curr is not None:
    # ... curr.item ...
    curr = curr.next

from __future__ import annotations
from typing import Any, Callable, Optional, Union

class _Node:
    """A node in a linked list.

    Note that this is considered a "private class", one which is only meant
    to be used by the code in this module. For the LinkedList class, _Node and _Node.item
    are not part of the public API.
    """
```

Takeaways

Code templates are useful.
Code templates aren't everything.

Writing a stopping condition is often easier to understand than writing a loop condition.

```
LinkedList
=====
CSC448 Winter 2019
Department of Computer Science,
University of Toronto

Module Description
=====
This module contains the code for a linked list implementation with two classes,
LinkedList and _Node.

Code template we can use:

curr = self._first
while curr is not None:
    # ... curr.item ...
    curr = curr.next

from __future__ import annotations
from typing import Any, Callable, Optional, Union

class _Node:
    """A node in a linked list.

    Note that this is considered a "private class", one which is only meant
    to be used by the code in this module. For the LinkedList class, _Node and _Node.item
    are not part of the public API.
    """
```

```
LinkedList
=====
CSC448 Winter 2019
Department of Computer Science,
University of Toronto

Module Description
=====
This module contains the code for a linked list implementation with two classes,
LinkedList and _Node.

Code template we can use:

curr = self._first
while curr is not None:
    # ... curr.item ...
    curr = curr.next

from __future__ import annotations
from typing import Any, Callable, Optional, Union

class _Node:
    """A node in a linked list.

    Note that this is considered a "private class", one which is only meant
    to be used by the code in this module. For the LinkedList class, _Node and _Node.item
    are not part of the public API.
    """
```

```
LinkedList
=====
CSC448 Winter 2019
Department of Computer Science,
University of Toronto

Module Description
=====
This module contains the code for a linked list implementation with two classes,
LinkedList and _Node.

Code template we can use:

curr = self._first
while curr is not None:
    # ... curr.item ...
    curr = curr.next

from __future__ import annotations
from typing import Any, Callable, Optional, Union

class _Node:
    """A node in a linked list.

    Note that this is considered a "private class", one which is only meant
    to be used by the code in this module. For the LinkedList class, _Node and _Node.item
    are not part of the public API.
    """
```



