CSCA48 Winter 2016

Week 9 – Binary Tree & Term Test #2

Bo(Kenny) Zhao

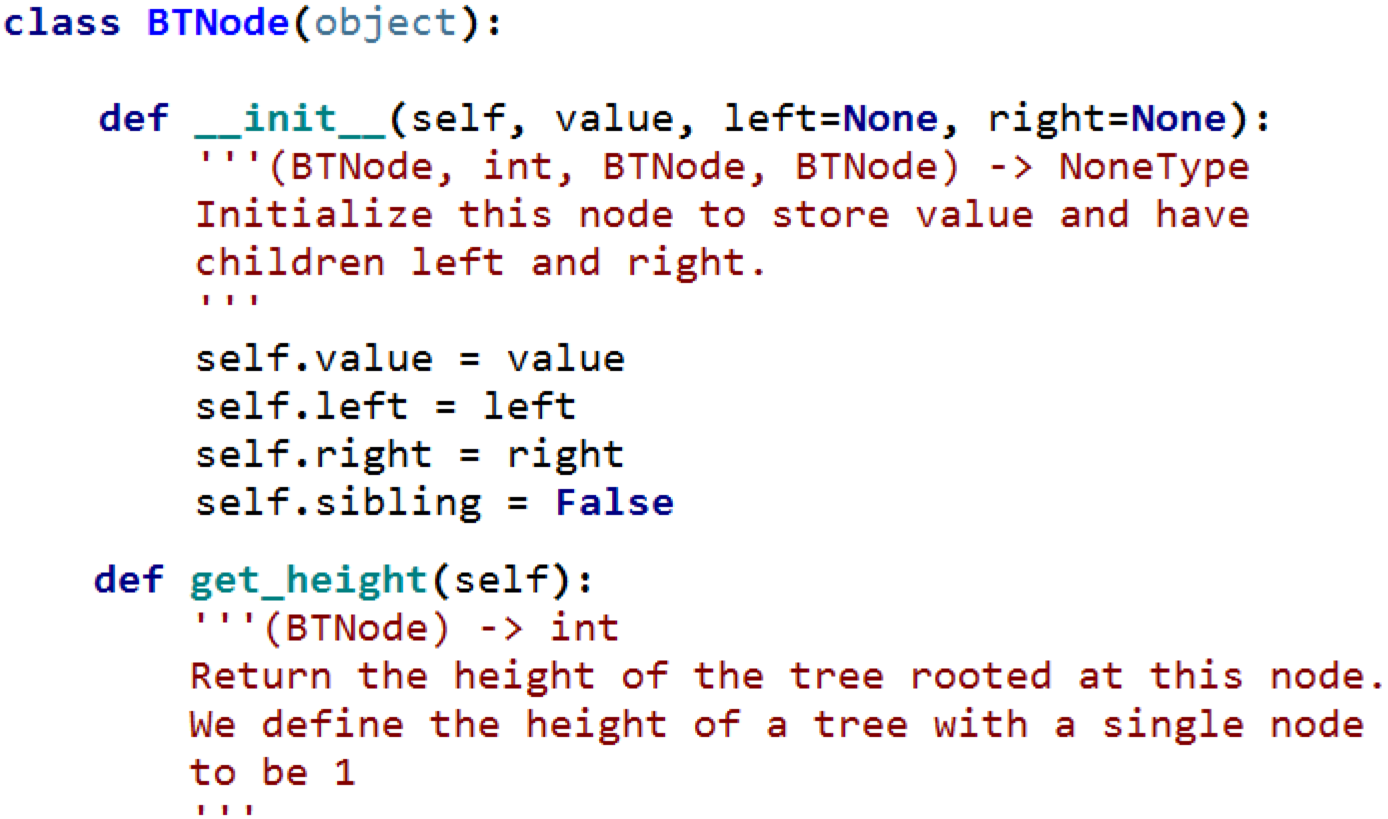
University of Toronto Scarborough

March 9, 2016

Today’s Plan

- Implement two methods in BTNode class (in small groups)
    1. get_height()
    2. set_sibling()
- Term test #2
    1. Topics
    2. How to prepare
- Quiz 4

Implement the method get_height()

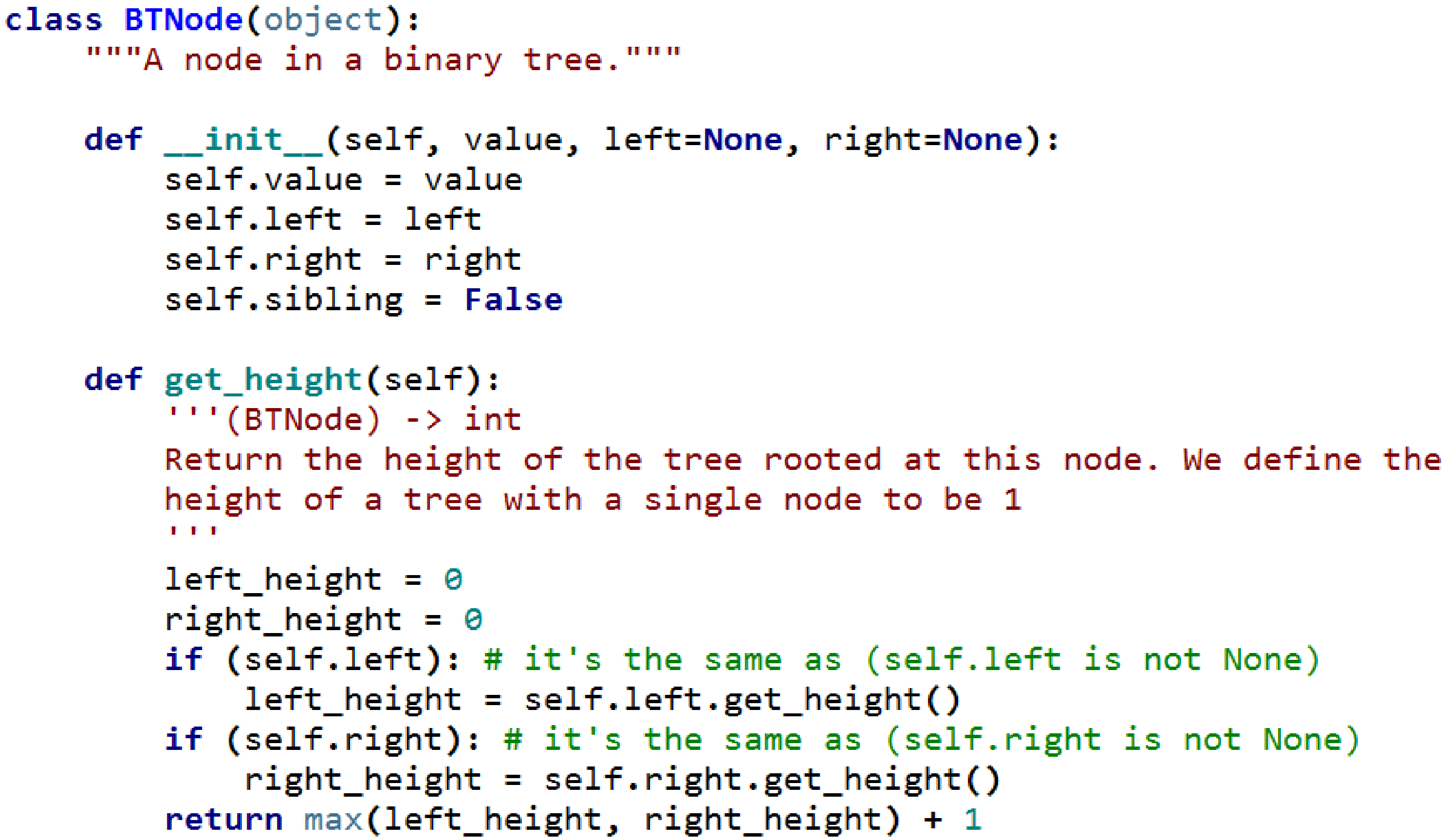


```
class BTNode(object):

    def __init__(self, value, left=None, right=None):
        '''(BTNode, int, BTNode, BTNode) -> NoneType
        Initialize this node to store value and have
        children left and right.
        '''
        self.value = value
        self.left = left
        self.right = right
        self.sibling = False

    def get_height(self):
        '''(BTNode) -> int
        Return the height of the tree rooted at this node.
        We define the height of a tree with a single node
        to be 1
        '''
```

Implement the method get_height()

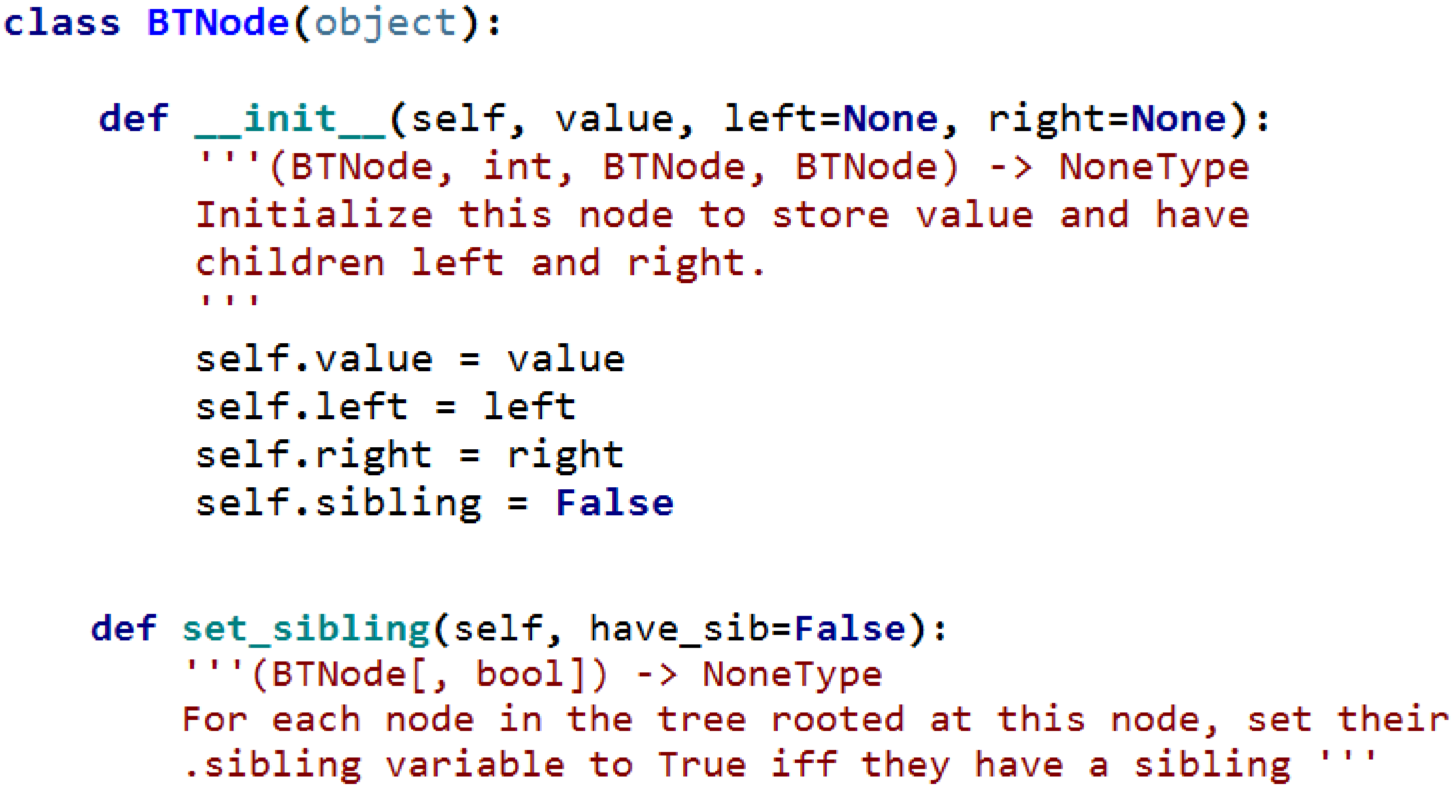


```python
class BTNode(object):
    """A node in a binary tree."""

    def __init__(self, value, left=None, right=None):
        self.value = value
        self.left = left
        self.right = right
        self.sibling = False

    def get_height(self):
        '''(BTNode) -> int
        Return the height of the tree rooted at this node. We define the
        height of a tree with a single node to be 1
        '''
        left_height = 0
        right_height = 0
        if (self.left): # it's the same as (self.left is not None)
            left_height = self.left.get_height()
        if (self.right): # it's the same as (self.right is not None)
            right_height = self.right.get_height()
        return max(left_height, right_height) + 1
```

Implement the method set_sidbling()

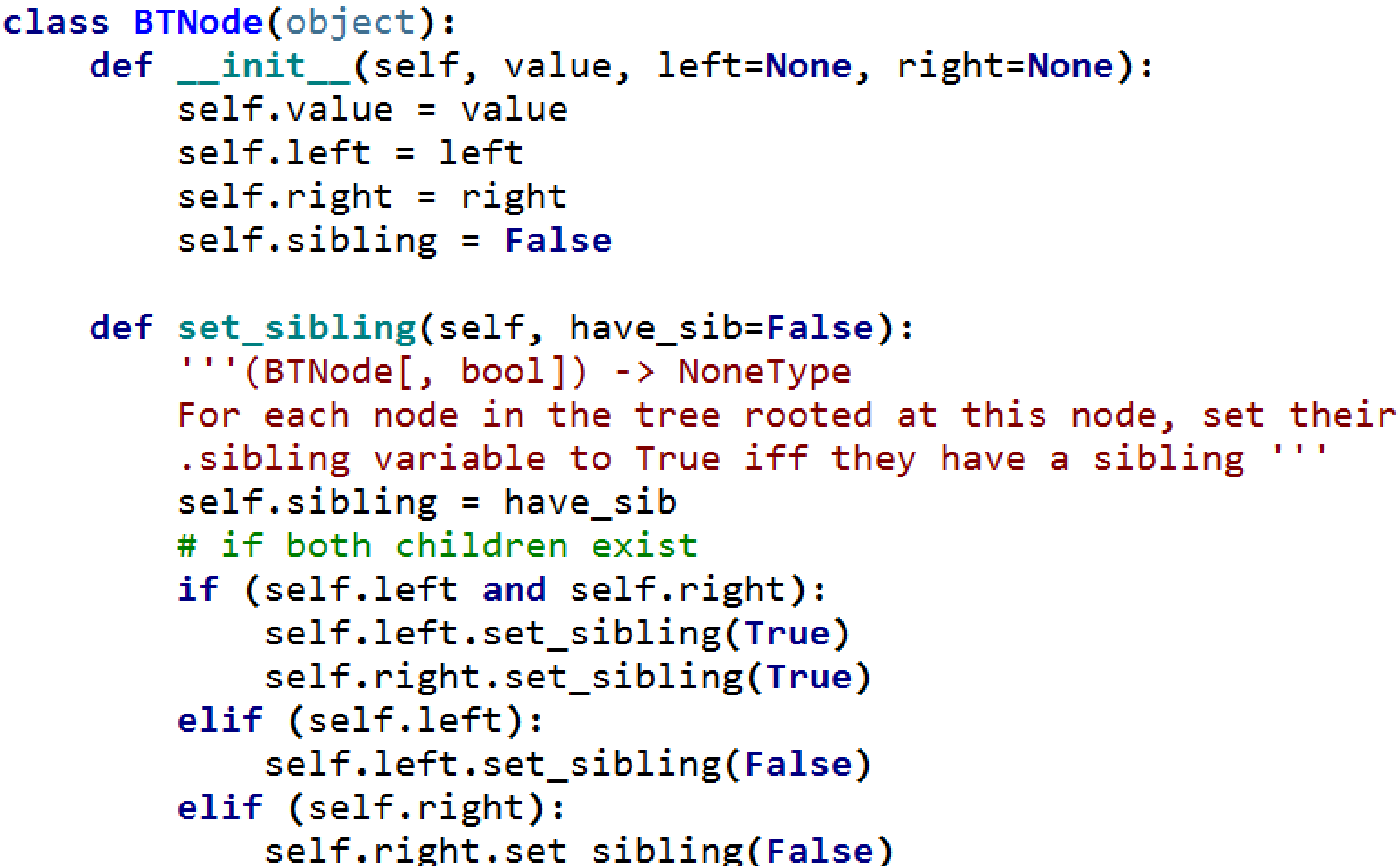


```
class BTNode(object):

    def __init__(self, value, left=None, right=None):
        '''(BTNode, int, BTNode, BTNode) -> NoneType
        Initialize this node to store value and have
        children left and right.
        '''
        self.value = value
        self.left = left
        self.right = right
        self.sibling = False


    def set_sibling(self, have_sib=False):
        '''(BTNode[, bool]) -> NoneType
        For each node in the tree rooted at this node, set their
        .sibling variable to True iff they have a sibling '''
```

Implement the method set_sidbling()

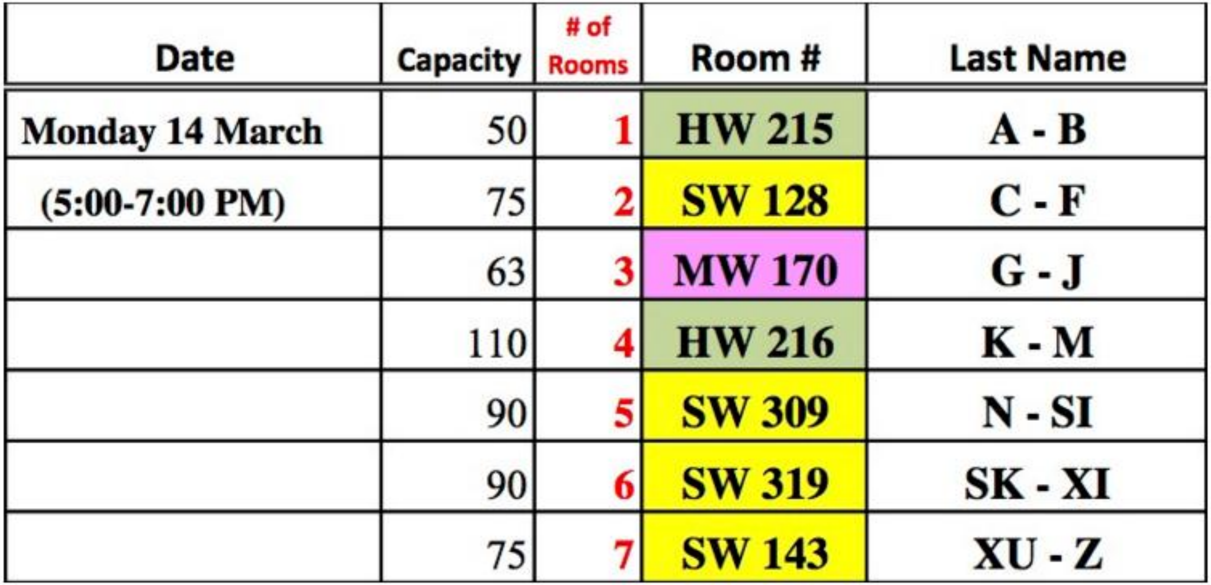


```
class BTNode(object):
    def __init__(self, value, left=None, right=None):
        self.value = value
        self.left = left
        self.right = right
        self.sibling = False

    def set_sibling(self, have_sib=False):
        '''(BTNode[, bool]) -> NoneType
        For each node in the tree rooted at this node, set their
        .sibling variable to True iff they have a sibling '''
        self.sibling = have_sib
        # if both children exist
        if (self.left and self.right):
            self.left.set_sibling(True)
            self.right.set_sibling(True)
        elif (self.left):
            self.left.set_sibling(False)
        elif (self.right):
            self.right.set_sibling(False)
```

Term Test #2 (date & location)

The second term test will be held on **Monday March 14th, 2016** from **5:00-7:00pm**. The test will be closed book, no aids allowed. You must bring your student card. Watch this space for room allocation. The room allocation is as follows:



Date	Capacity	# of Rooms	Room #	Last Name
Monday 14 March	50	1	HW 215	A - B
(5:00-7:00 PM)	75	2	SW 128	C - F
	63	3	MW 170	G - J
	110	4	HW 216	K - M
	90	5	SW 309	N - SI
	90	6	SW 319	SK - XI
	75	7	SW 143	XU - Z

http://www.utsc.utoronto.ca/~bharrington/csca48/tests.shtml

Term Test #2 (topics)

You're responsible for everything from week 1 of A08 up until the cutoff. But generally the focus will tend to be on the **material covered since the previous test**.

—Brian

The term test #2 covers everything before and including heaps.

i.e. no complexity

Term Test #2 (preparation)

Go over all lecture notes/tutorial code.

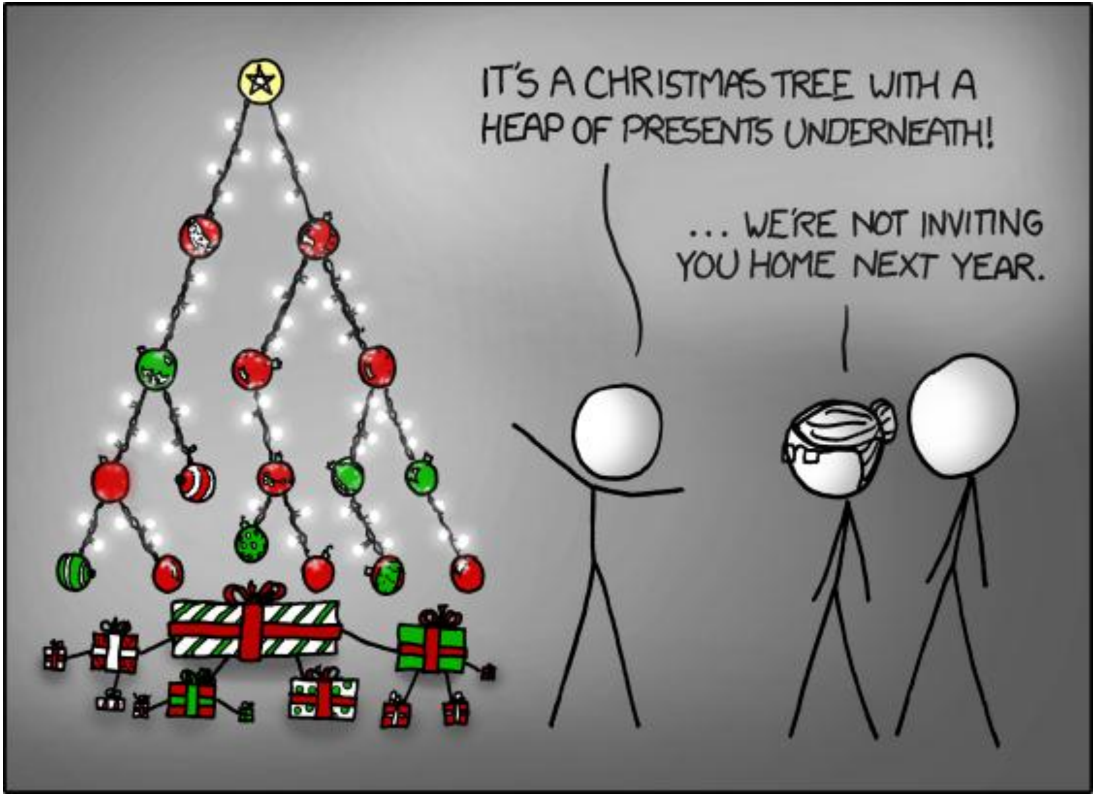
Redo exercises/quizzes if you didn’t get perfect.

Try some past tests. Time yourself and don’t use Python.

Need help?
Practicals/Brian and Nick’s office hours/Piazza/Anti-lecture

http://www.utsc.utoronto.ca/~bharrington/csca48/practicals.shtml

Quiz


IT'S A CHRISTMAS TREE WITH A HEAP OF PRESENTS UNDERNEATH!
... WE'RE NOT INVITING YOU HOME NEXT YEAR.

https://xkcd.com/835/