

## CSC148 Summer 2016 Quiz 08 (15 minutes)

First\_name ..... Last\_name ..... Lab\_room# .....

Recall the declaration of class BinaryTree below.

```
class BinaryTree:
    """ A Binary Tree, i.e. arity 2.
    """

    def __init__(self, data, left=None, right=None):
        """
        Create BinaryTree self with data and children left and right.
        :param object data: data of this node
        :param BinaryTree|None left: left child
        :param BinaryTree|None right: right child
        """

        self.data, self.left, self.right = data, left, right
```

Read the docstring and examples below. Then, implement `max_value2`, WITHOUT using recursion. If recursion is used, there will be a 50% mark deduction.

```
from csc148_queue import Queue
def max_value2(t):
    """
    Return the max value in BinaryTree t

    :param t: a not None binary tree
    :type t: BinaryTree
    :rtype: object

    >>> t1 = BinaryTree(8)
    >>> max_value2(t1)
    8
    >>> t2 = BinaryTree(8, BinaryTree(7, BinaryTree(12), BinaryTree(5)), BinaryTree(11))
    >>> max_value2(t2)
    12
    """

    my_max= t.data
    q = Queue()
    q.add(t)
    while not q.is_empty():
        t1 = q.remove()
        my_max = max(t1.data, my_max)
        if t1.left:
            q.add(t1.left)
        if t1.right:
            q.add(t1.right)
    return my_max
```