

Quiz #2

CISC 121, Winter 2011

*****Please do NOT turn over this page and begin the quiz until you are given permission*****

STUDENT NUMBER: -

Please do not write your name on the quiz, but write your student number clearly above and on every page that contains an answer.

Please remain quiet in the classroom. If you finish early, give your paper to the instructor or put it in the space provided and leave quietly to avoid disturbing those who are still working. Your eyes should always be on your paper, the instructor, or the blackboard. Please do not incur suspicion by looking all around the room, especially not in the direction of other students' papers. **Academic dishonesty of any kind will not be tolerated.**

You may use **one page of notes** (your "summary sheet") as a memory aid during the quiz. You may not refer to any other notes or books you may have brought with you. You may not use a calculator or computer.

Please make sure that you hand in every page of the quiz that contains an answer and that these pages are securely stapled together. There will be a stapler available.

You have **45 minutes** to complete this quiz. The quiz is marked out of **30 points**. Budget your time accordingly. The weight for each question is given with the question.

Please try to answer each question in the space provided. If you write an answer in a different place, please write a note indicating where I should look for it. If you need extra space for answers or rough work, you may use the backs of the pages.

Your answers will be marked for correctness, not for style. Comments are not required. However, I can't mark anything that isn't legible, and clear style or comments may help determine partial credit.

If you write more than one answer to a question, please make it very clear which answer should be marked. Circle your final answer and write "mark this" or cross out your other answers. If it's not obvious what you intended, I will arbitrarily pick an answer to mark.

This area for marking purposes only:

1. _____ / 10

2. _____ / 10

3. _____ / 10

total: _____ / 30

Question 1 (10 points):

This question shows part of an interactive Python session. In the box after each `print`, show the output that would result.

```
>>> listA = [1,3,5,7]
>>> listB = listA
>>> listA[3] = 9
>>> print(listB)
```

```
>>> listC = [2,4,6,8]
>>> listD = listC
>>> listC.append(10)
>>> print(listD)
```

```
>>> listE = [10,20,30]
>>> listG = listE
>>> listE = listE + [40]
>>> print(listG)
```

```
>>> listK = [1.1,2.2,3.3,4.4]
>>> listL = listK[1:]
>>> listK[1] = 5.5
>>> print(listL)
```

(continued on next page)

Question 1, continued:

The session on this page uses the following class definition:

```
class TwoNums:
    def __init__(self, val1, val2):
        self.one = val1
        self.two = val2
    def __str__(self):
        return str(self.one) + "," + str(self.two)
    def change(self):
        self.one += 1
        self.two += 2
    def copy1(self):
        return self
    def copy2(self):
        return TwoNums(self.one, self.two)
```

```
>>> a = TwoNums(1,1)
>>> b = a
>>> c = TwoNums(2,2)
>>> a = c
>>> a.one += 10
>>> b.one += 20
>>> c.one += 50
>>> print(a)
```

```
>>> print(b)
```

```
>>> print(c)
```

```
>>> d = TwoNums(3,5)
>>> e = d.copy1()
>>> f = d.copy2()
>>> d.change()
>>> print(d)
```

```
>>> print(e)
```

```
>>> print(f)
```

Question 2 (10 points):

In the Week 4 lab, you worked with lists of lists of integers to represent a greyscale image. One of the transformations that an image processing program might do with an image is to rotate it.

Write a function called `rotate` which takes a list of lists of integers as a parameter. The function should return a new list of lists of integers which is the same as the parameter rotated 90 degrees clockwise. The function should not change its parameter.

If the numbers in `image` represent this picture:



the numbers in `rotate(image)` should represent this picture:



Here are two small examples:

```
>>> image1 = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]
>>> image2 = rotate(image1)
>>> image2
[[9,5,1],[10,6,2],[11,7,3],[12,8,4]]
>>> image3 = [[1,2],[3,4],[5,6],[7,8]]
>>> image4 = rotate(image3)
>>> image4
[[7,5,3,1],[8,6,4,2]]
```

Your function does not need to do any error checking. It may assume that its parameter will be a list of lists of integers, and that every list inside the list (every row of the image) has the same size. It may not make any other assumptions about the size of the list.

Please write your function in the space provided on the next page.

Question 2, continued:

```
def rotate(image):  
    """  
    Assumes image is a "rectangular" list of lists: each row has the  
    same length. Returns a new image which is the parameter rotated  
    clockwise 90 degrees. Does not change the parameter.  
    """  
    # Write the body of the function below:
```

Question 3 (10 points):

Write a new method called `delAfter` for the `LinkedList` class. Below is part of the `LinkedList` module, showing where the new method would go:

```
class ListNode:
    def __init__(self, value):
        self.data = value
        self.next = None # special value, means no next node

class LinkedList:
    def __init__(self):
        self.first = None

    def delAfter(self, value):
        """
        Deletes the node *after* the first node containing value.
        If value doesn't occur in the list or occurs in the last
        node of the list, the list is not changed
        """
```

The body of your method would go here, but write it in the box on the next page

Important: Your `delAfter` method may not call any of the other methods from the original `LinkedList` class (`set`, `get`, `delete`, etc.)

Here is a transcript of part of a Python session using my solution to this problem.

```
>>> print(nums)
[2, 3, 8, 2, 6, 3]
>>> nums.delAfter(2)
>>> print(nums)
[2, 8, 2, 6, 3]
>>> nums.delAfter(6)
>>> print(nums)
[2, 8, 2, 6]
>>> nums.delAfter(6)
>>> print(nums)
[2, 8, 2, 6]
>>> nums.delAfter(3)
>>> print(nums)
[2, 8, 2, 6]
```

Question 3, continued:

Please write the body of your `delAfter` method in the box below.

```
def delAfter(self, value):  
    """  
    Deletes the node *after* the first node containing value.  
    If value doesn't occur in the list or occurs in the last  
    node of the list, the list is not changed  
    """
```