Do not sign this page.
It will be removed.

CS1114
Fall 2014

SECOND Test

NOTE: There are LONGER problems at the end of the test and a couple of functions to define before that.
A good strategy would be to skip any questions you cannot do quickly
to get to the LONGER problems at the end of the test; then go back through the shorter ones.

Closed book, closed notes, NO electronic devices of any type

You will have ONE HOUR AND 40 MINUTES to complete this test
You may not leave until 50 minutes have passed.
If you leave before the end of the test and there are corrections announced after you leave…

Do NOT tear any pages out of your Blue Book.
Do NOT tear any pages from this document.
Be sure to hand in all nine pages of this test.
Do not remove this page.

Place your answers for questions 1 through 13 in this document.
If you write any of these answers in your Blue Book you will get
NO CREDIT for them.
Place your answers for questions 14 and 15 in your Blue Book.

Put your name and ID and **circle your last name** at the top of each page
of this document and **circle your section letter**.

Place your name and ID on the cover of the Blue Book and **circle your last name**.

WRITE YOUR SECTION LETTER ON THE BLUE BOOK COVER

Kletenik **A** (10:00) Kletenik **C** (3:00) Gallagher **B** (3:00) Gallagher **D/E** (4:30)

If you need "scratch" paper, use your Blue Book
but cross out anything you do not want graded
and do not remove any page from your BlueBook..

There are 15 questions totaling ***144*** points
There are no optional questions. You must earn ***144*** points to score a 100%

Do not begin until you are instructed to do so
If you continue writing after time has been called
you will have earned a ZERO.

If your cell phone rings during this test
or if you answer your cell phone, you will receive a ZERO.
All cell phones must be inside bags or pockets during the entire test.
No cell phones are allowed on your desk at any time during the test.
Turn off all cell phones before the test starts
(If you are expecting a call, BEFORE the test starts leave your phone with the test monitor)

CS1114 Test TWO Fall 14 Print your name legibly:____

circle your last name

Circle your SECTION:

Kletenik A (10:00) Kletenik [✓ C] (3:00) Sign:____________________________________

Gallagher B (3:00) Gallagher D/E (4:30) Poly-ID:____________________________________

CIRCLE YOUR SECTION LETTER
ON EVERY PAGE

CIRCLE YOUR SECTION LETTER

The rest of the document's headers will not show this
but you must CIRCLE YOUR SECTION ANYWAY

CIRCLE YOUR SECTION LETTER

CIRCLE YOUR SECTION LETTER

The rest of the document's headers will not show this
but you must CIRCLE YOUR SECTION ANYWAY

CIRCLE YOUR SECTION LETTER

CS1114 Test TWO Spring 14 Print your name legibly:______________________________________
circle your last name

Sign:__________________________________

Poly-ID:____________________

Throughout the test, assume that from __future__ import print_function is there if you want to use the print function instead of the print statement – you are allowed to use either

1. (3 pts) The list type is an iterable object.

(circle your answer) TRUE FALSE

WHAT IS THE OUTPUT questions – write your answer to the right of the question

If the code won't compile, write "won't compile"
If it will raise a runtime exception, state the exception.
if you cannot remember the name of the exception, just write 'exception'

2. (3 pts)

```
one = "two"
two = "one"
two += one + "two"
print( one, two )
```

3. (3 pts)

```
lissy = { "testing" : "lissy" }
print( lissy [ "testing" ] )
```

4. (3 pts)

```
s = [ ['a'], ['b'], 'c' ]
t = s.pop( s. remove ( 'a' ) )
print( t )
```

5. (3 pts)

```
one = [ 2, 3, 4, 5, 6, 7 ]
one[ 2:4 ] = [ 55, 11 ]
print( one )
```

Print your name legibly:______________________________
circle your last name

Sign:______________________________

Poly-ID:____________________

WHAT IS THE OUTPUT questions – write your answer to the right of the question

If the code won't compile, write "won't compile"
If it will raise a runtime exception, state the exception.
if you cannot remember the name of the exception, just write 'exception'

6. (3 pts)

```
one = ( 2, 3, 4 )
print( one[ 2: ] )
```

7. (3 pts)

```
one = [ 'snake', 'piggy', 'tiger'  ]
for two in one:
    print( two[ 2 ] )
```

8. (3 pts)

```
alpha = { 1:"joe", 3:"sue", "joe":4 }
beta  = { alpha[ "joe" ] : 1 }
print( beta )
```

9. (**6 pts**) Given:

```
def alpha( one ):
    ''' ??? '''
    one.append( 3 )
    one = 3

def main():
    one = [ 4, 3, 2 ]
    print( "before" )
    print( one )
    alpha( one )
    print( "after" )
    print( one )
```

When this program is run, what is the output?

CS1114 Test TWO Spring 14 Print your name legibly:____________________________________
circle your last name

Sign:__________________________________

Poly-ID:____________________

WHAT IS THE OUTPUT questions – write your answer to the right of the question

If the code won't compile, write "won't compile"
If it will raise a runtime exception, state the exception.
if you cannot remember the name of the exception, just write 'exception'

10. (3 pts)

```
alpha = [ 4, '44', '4', [ '4' ], 'four' ]
alpha.remove( '4' )
print( alpha )
```

11. (3 pts)

```
alpha = { "S" : "snakey", "T" : "turkey", "D" : "doggie" }
print( alpha[ '''S''' ] )
```

12. (3 pts)

```
alpha = [ 5, 4, 3, 2, 1 ]
print( alpha.index(5) )
```

13. (3 pts) The dictionary type has the `.count()` method? (**circle your one answer**)

a) yes b) no

Print your name legibly:______________________________
circle your last name

Sign:______________________________

Poly-ID:____________________

14. (15 pts) Complete the definition of this function so that it does what the docstring states.

```
def avgDepth( dictOfLakeToDepth, listOfLakeNames ):
    ''' Takes a dictionary of lake names that map to the lake's depth.
        The depth is of float type.
        Also takes a list of lake names to process.
        Returns the average depth of those lakes in the list of names.
        You are guaranteed that there is at least one of the lake
        names in both the dictionary and the list thus preventing
        division by zero from happening.
    '''
```

Write your answer here, in this document.

CS1114 Test TWO Spring 14 Print your name legibly:_____________________________________
circle your last name

Sign:____________________________________

Poly-ID:______________________

15. (15 pts) Complete the of definition this function so that it does what the docstring states.

```
def removeAllButLast( listy, valueToRemove ):
    ''' Takes a list and a value to remove from the list.
        Returns a list with all but the last (rightmost) instances of
        value removed.
        Does not change the original that was passed in.
        For example if the passed in list is: [ 2, 4, 2, 6, 7, 2, 8 ]
        and 2 is to be removed,
                   the returned list will be: [ 4, 6, 7, 2, 8 ]
 '''
```

Write your answer here, in this document.

For the BlueBook question(s) :

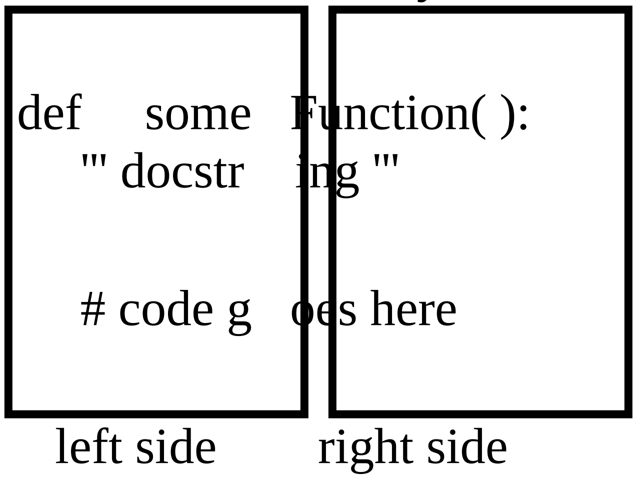
0. WRITE YOUR SECTION LETTER ON THE COVER

Kletenik **A** (10:00) Kletenik **C** (3:00) Gallagher **B** (3:00) Gallagher **D/E** (4:30)

1. Use blue or black INK or write in VERY DARK PENCIL.
   Do not write in tiny print.
   or**:** expect to get a zero for your work.

also

2. Start writing on the LEFT side of a pair of pages
   and write all the way across BOTH pages:



Do NOT start writing on any right hand page.

Do not remove any page from the Blue Book.

Print your name legibly:______________________________________
circle your last name

Sign:__________________________________

Poly-ID:____________________

Write your answer to this question in your BLUEBOOK

If you need "scratch" paper, use the Blue Book but cross out anything you do not want graded.
Start on the left hand page of each pair of pages.

16. (40 pts) The University has sent out its students to get donations to the U.
Each student goes around and collects donation amounts from donors.

After a student has gone around for a while, they report back to the computer running your program.
They wait together and then one by one goes up to the computer running your code.

Each student enters their student ID (a str) and how many donations they have to enter.
Then, for each donation, they will enter the name (just a str) of the donor and the amount that donor gave.
A donor might have donated to this student more than once. These will be added to the donor's previous amount.
Each student reports only once.

There are 25 students who will go out taking donations and all have different IDs.
For the example run shown below, we only show the first three students.

The purpose of your program is to build a dictionary that will contain the information about which donors and how much they gave to each student.

It's completely fine if a donor gives to more than one student.

Your job is build a dictionary of dictionaries so that some other programmer, later on, can add the ability to find out, for any specific student, by using their ID, the names and donation amounts of all the donors that donated to that student.

Do NOT write the code that does the lookup for a student.
Only write `main` and the one function described below.

---- REQUIREMENTS:

Define a function named `getOneStudentData` that handles ONE student's collected donations.
It will be passed the dictionary of dictionaries.
It gets student's ID (no validation is done). It then ask for how many donations they will enter (again without validation).
It then it loops, asking for the name and donation amount of each donor.
If this is a first time donation to this student, it enters the donor's name and donation amount.
If the donor has previously donated to this student, it updates that donor's amount.
You must implement the donor and their donation amount as a dictionary.
This dictionary will be entered as the value for this student's ID in the passed in dictionary which will hold data for all studuents.

Define `main` so that the whole problem of building the dictionary of dictionaries to hold all the students' data is solved there.
`main` is where the `for` statement will be that calls `getOneStudentData` twenty-five times.
Your student data dictionary must be defined as an empty dictionary in `main` and passed into `getOneStudentData` in the `for` loop.
That is **ALL** that will be in your main. No calculating, no printing.
In `main`, just define an empty dictionary and pass it into `getOneStudentData` in the `for` loop.
And you must define `getOneStudentData`, of course.

CS1114   Test TWO Spring 14   Print your name legibly:______________________________________
circle your last name

Sign:____________________________________

Poly-ID:_____________________

You do not have to write any docstrings if you choose good names.
You do not have to make your program executable or double-clickable.
Define no modules.

Here is a sample run showing only the first three students.
The dictionary of dictionaries your code must produce for this sample run is shown to the right at the bottom.
User input is shown in **bold**.

```
Enter your ID 10293
How many donations? 3
Name and $ for each:
Bob Blizzard
10.00
Mary Snow
8.99
Cold Mann
15.01

Enter your ID 20222
How many donations? 2
Name and $ for each:
Percy Winter
8.44
Bob Summer
1002.33

Enter your ID 30021
How many donations? 2
Name and $ for each:
John Sterling
0.01
John Sterling
0.01

Enter your ID
...
```

Showing only the first three students.

Here is the dictionary of dictionaries that your code will have produced after these first three students have entered their data:

```
{ 10293 : { 'Bob Blizzard' : 10.00,
            'Mary Snow' : 8.99,
            'Cold Mann' : 15.00 },
  20222 : { 'Percy Winter' : 8.44,
            'Bob Summer' : 1002.33 },
  30021 : { 'John Sterling' : 0.02 }  }
```

Notice that 0.02 is the sum of Sterling's donations

That's all you code has to do.

There is no calculating or printing.

Just put the donor data into the dictionary as shown by calling `getOneStudentData`. in main's for loop.

You do NOT write the lookup for this problem – ONLY write the adding of the data for each students' donations