CS 301 - Spring 2016

Instructor: Laura Hobbes LeGault

Midterm Exam 2 — 16.67%

Full name: ______________________________

Student ID #: ______________________________

I certify that I will keep my answers covered and do my best to not allow my exam paper to be viewed by another student during the exam or prior to completion of their exam. I also certify that I have not viewed or any way used another's work in completing my answers. I understand that being caught allowing another to view my work or being caught viewing another's work are both violations of this agreement and either will result in automatic failure of the course and an academic misconduct letter to the Deans Office for myself and any other individuals involved.

Signature: ______________________________

The following exam has 26 questions and is worth a total of 88 points. You will have 50 minutes to complete the exam. **Be sure to read through every question completely.**

The questions on the exam are as follows:

1. **Dual Choice** — 12 questions worth 2 points each.
2. **Multiple Choice** — 10 questions worth 4 points each. Choose the *best* answer.
3. **Fill-in-the-blank** — 4 blanks worth 6 points each. Be complete.

You may not use notes or books, your neighbors, or calculators or any other electronic devices on this exam. **Turn off and put away** your cell phone, pager, Inspector Gadget Watch, etc. now.

IMPORTANT: Answers for Dual and Multiple Choice questions *must* be marked on a scantron. The answer marked on the scantron will be the only answer graded.

Disclaimer: the following are provided for your reference only, and the inclusion of information here does not guarantee it will be used on the exam.

Operator Precedence Table:

level	operator	description
higher	`( <expression> )`	grouping with parentheses
	`x[index:index]`	slicing
	`x[index]`	indexing
	`* / %`	multiplicative
	`+ -`	additive
$\updownarrow$	`< <= > >=`	relational
	`== !=`	equality
	`not`	logical not
	`and`	logical and
lower	`or`	logical or
	`= += *=`	(compound) assignment

Built-in functions:

`raw_input(p)`	Prompts the user for input using `p` and returns the user's input as a string.
`len(s)`	Return the length (the number of items) of an object.
`range(n)`	Returns a list of `n` consecutive integers beginning at `0`.
`range(a,b)`	Returns a list of consecutive integers beginning at `a` and ending before `b`.
`type(x)`	Returns the *data type* of the value stored in `x`

Constants and functions from math, string, and random modules:

`math.pi`	The mathematical constant $\pi = 3.141592...$
`w.isdigit()`	Return true if all characters in the string `w` are digits and `w` is not empty.
`random.randint(a,b)`	Return a random integer N such that `a <= N <= b`.

List and dictionary methods:

`list.append(x)`	Add the value `x` to the end of `list`.
`list.insert(i,x)`	Insert the value `x` at the `i`th index of `list`.
`list.remove(x)`	Remove the first instance of the value `x` from `list`.
`list.pop(i)`	Remove the value at index `i` from `list`.
`dict.keys()`	Return a copy of `dict`'s list of keys.
`dict.values()`	Return a copy of `dict`'s list of values.

A or B: Terminology

1. If a function does not explicitly return a particular value, it returns the value __________. (2)

    A. `""` (empty string)

    B. None

2. In the expression `d["a"] = 5`, the literal `"a"` serves as a ______________. (2)

    A. key

    B. index

3. Dictionary keys *must* be ______________. (2)

    A. mutable

    B. immutable

4. Adding one to an integer variable is called ______________. (2)

    A. incrementing

    B. iterating

5. To use a value elsewhere in a program, a function must ______________ that value. (2)

    A. `print`

    B. `return`

6. In this expression: (2)

    ```
    x = [random.randint(1,10) for i in range(5)]
    ```

    `x` is created using a technique called ______________.

    A. list construction

    B. list comprehension

7. `L` is a copy of `M`. If I can change `M[0]` *without* affecting `L`, `L` must be a __________ of `M`. (2)

    A. deep copy

    B. shallow copy

8. Function arguments are given values when the function is ______________. (2)

    A. called

    B. defined

True or False: Evaluating boolean expressions

9. `True and (False or not False)` (2)

   A. True

   B. False

10. `int(43.1) == 43.0` (2)

   A. True

   B. False

11. `"$14.50".isdigit() and math.pi < 5` (2)

   A. True

   B. False

12. `len( range(4,5) ) == 1` (2)

   A. True

   B. False

Multiple Choice: Reading code

13. Given this while loop condition (and assuming `x` has already been initialized to `"A"`), which of the following loop contents will not necessarily result in an infinite loop? (4)

```
while not x.isdigit():
    CODE
```

   A. `print (x)`

   B. `x += "1"`

   C. `x = x[:]`

   D. `x = input()`

14. Complete the following for loop header (assume `my_list` is a non-empty list of strings): (4)

```
for i in ITER:
    my_list[i] += my_list[i+1]
```

A. `ITER: len( range(my_list) )`

B. `ITER: len( my_list )`

C. `ITER: range( len(my_list)-1 )`

D. `ITER: my_list`

15. Which function call **cannot** have happened between the first and last line of this code fragment, given the output? (4)

```
print (L)  # prints [5, 6, 3, 1, 2, 2]
CODE
print (L)  # prints [5, 6, 3, 1, 2]
```

A. `L.remove(2)`

B. `L.remove(5)`

C. `L.pop()`

D. `L.pop(5)`

16. Given the following valid code, what is the *type* of the variable `a`? (4)

```
a["O"] = ["X", " ", "X"]
```

A. string

B. list

C. dictionary

D. array

17. All of the following statements about this *valid* line of code are True **except**: (4)

```
sample[5] += 1
```

A. `type(sample)` cannot be string.

B. The value at `sample[5]` may be a float.

C. If `sample` is a list, `len(sample) > 5`.

D. If `sample` is a dictionary, this code adds the key `6`.

18. What is the output when the following code is run? (4)

```
def my_fcn(x):
    for i in x:
        if i % 2 == 0:
            print (i),
        else:
            return

my_fcn([2, 3, 4])
```

A. 2

B. 2 4

C. 0 2

D. There is no output.

19. What is the output of the following line of code? (4)

```
print (str( len( range(5)[:2] ) ).isdigit())
```

A. True

B. [0, 1, 2]

C. [0, 1, 2, 3, 4]

D. 2

20. What is the value of the counter after the following code is run? Be careful and trace through ALL of the code. (4)

```
counter = 0

for i in range(5):
    for j in range(i):
        counter += 1
    counter -= 1
```

A. 0

B. 10

C. 5

D. -5

21. Which of the following types is *not* a legal dictionary **value**? (4)

A. string

B. int

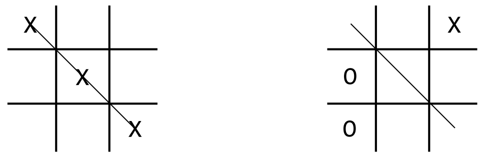
C. dictionary

D. All of the above are legal dictionary **value** types.

Fill-in-the-blank: Writing code

Fill in the blanks to make the functions behave as the comments indicate. Each line is worth **6 points**, and there are a total of 4 lines.

22. For *full* credit, your function must return `True` for the left configuration and `False` for the one on the right:



```
[['X', ' ', ' '],      [[' ', ' ', 'X'],
 [' ', 'X', ' '],       ['O', ' ', ' '],
 [' ', ' ', 'X']]       ['O', ' ', ' ']]
```

```
def diagonal_win_l2r(board):
    """ board is a 3x3 list-of-lists.  Finish the condition so the
       function returns True if there is a line of 'X' or a line of 'O'
       diagonally from top left to bottom right, and False otherwise.
    """

     if ________________________________________________:        (6)
         return True
     return False
```

23.
```
def mean(L):
    """ L is a list of int values.  Fill in the blanks so the
        function returns the mean (average) of all list values.
        Recall the mean of N values is their sum divided by N.
    """
    total = 0
    for num in L:

        ________________________________________________        (6)

    return total / ____________________                         (6)
```

24.
```
def ltr_to_num(s, ltr_dict):
    """ s is a string, ltr_dict is a dictionary which contains
        lowercase letters as keys and integers as values.
        Complete the function so it prints the numeric equivalent
        of each character in the string.
    """
    i = 0
    while i < len(s):

        print ________________________________________ ,        (6)
        i = i + 1
```