

CS 301 - Spring 2017
Instructor: Laura Hobbes LeGault

Midterm Exam 2 — 16.67%

(Last) Surname: _____ (First) Given name: _____

NetID (email): _____ @wisc.edu

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.

Fill in these fields (left to right) on the scantron form (use #2 pencil):

1. LAST NAME (surname) and FIRST NAME (given name), fill in bubbles
2. IDENTIFICATION NUMBER is your Campus ID number, fill in bubbles
3. Under ABC of SPECIAL CODES, write 002 (afternoon lecture), fill in bubbles
4. Under J of SPECIAL CODES, write D (exam version), fill in bubble 3

.....

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 in 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 44 points. You will have 50 minutes to complete the exam. **Be sure to read through every question completely.**

1. **Dual Choice** — 12 questions worth 1 point each. Choose the *best* answer.
2. **Multiple Choice** — 10 questions worth 2 points each. Choose the *best* answer.
3. **Fill-in-the-blank** — 4 blanks worth 3 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** any portable electronics now.

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]	indexing
	* / %	multiplicative
	+ -	additive
	< <= > >=	relational
lower	== !=	equality
	not	logical not
	and	logical and
	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.
`ord(c)` Return an integer representing the value of the single-character string `c`.
`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 methods from string and random modules:

`w.isalpha()` Return true if all characters in string `w` are letters, `w` not empty.
`w.isdigit()` Return true if all characters in string `w` are numbers, `w` not empty.
`w.upper()` Return the string `w` transformed to upper case.
`w.lower()` Return the string `w` transformed to lower case.
`random.randint(a,b)` Return a random integer N such that $a \leq N \leq b$.
`random.shuffle(x)` Shuffle the sequence `x` in place.

List and dictionary methods:

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

Dual Choice: Terminology

1. In list comprehension, an `if` statement would be placed _____ the `for` loop. (1)
 - A. before
 - B. after
2. The code `numbers[1:5]` _____ into the list `numbers`. (1)
 - A. indexes
 - B. slices
3. Given `numbers` is a list, the following function call gives `random.shuffle()` a _____ copy of the list: (1)

```
random.shuffle(numbers[:])
```

 - A. deep
 - B. shallow
4. Given `word` is a string, `word[0] = "A"` is **not** legal because strings are _____. (1)
 - A. immutable
 - B. mutable
5. A **key** in a Python dictionary must be both unique and _____. (1)
 - A. iterable
 - B. immutable
6. Repeated execution of a set of statements (usually with a loop) is called _____. (1)
 - A. incrementing
 - B. iterating
7. To store *mappings between values* where order is not important, use a _____. (1)
 - A. dictionary
 - B. list
8. **Appending** a value to a list places the value at the _____ of the list. (1)
 - A. beginning
 - B. end

True or False: Evaluating boolean expressions

9. `"apple" in { 1:"apple", 2:"banana", 3:"carrot" }` (1)
A. True
B. False
10. `"hello world".isalpha()` (1)
A. True
B. False
11. `ord("a") < ord("z")` (1)
A. True
B. False
12. `user_input = input("Yes or no?")` `# given this...` (1)
`user_input != "Y" or user_input != "y"` `# evaluate this`
A. True
B. False

Multiple Choice: Reading code

13. Which of the following best describes the *value* in `x` after this code executes? (2)
`x = [range(num) for num in range(10)]`
A. `x` has no value, because `range(num)` causes an error when `num = 0`.
B. A list containing 10 function calls.
C. A list containing 10 integers.
D. A list containing 10 lists.

14. What is the *type* of `x` after the following line of code is executed? (2)

```
x = len(str(input("Type something:")).isdigit())
```

- A. `str` (string)
- B. `bool` (boolean)
- C. `float`
- D. `int` (integer)

15. Given that `dict` is a dictionary initialized as (2)

```
dict = {"A":1, "B":2, "C":3}
```

which of the following lines of code changes the value associated with the key "B" to 0?

- A. `dict.insert("B", 0)`
- B. `dict.append("B", 0)`
- C. `dict["B"] -= 2`
- D. `dict[1] = 0`

16. What is the *value* in `x` after this (buggy) code executes? (2)

```
def flip_vowel_case(s):  
    for letter in s:  
        if letter in "aeiou":  
            return letter.upper()  
        else:  
            pass  
x = flip_vowel_case("hello world")
```

- A. This code causes a `TypeError` because `s` is not iterable.
- B. `"hE110 w0rld"`
- C. `"E00"`
- D. `"E"`

17. Given `my_list = ["a", 0, ["a", 0.5]]`, what is the *data type* of `x` after the following line of code is executed? (2)

```
x = my_list[2]
```

- A. `int` (integer)
- B. `float`
- C. `str` (string)
- D. `list`

18. What is the *value* in `x` after the following line of code is executed? (2)

```
x = range(3).append(4)
```

- A. `None`
- B. `[1, 2, 3, 4]`
- C. `[0, 1, 2, 4]`
- D. `4`

19. If the following code **does not** cause an error, what must the *data type* of `x` be? (2)

```
x["1"] == "a"
```

- A. `str` (string)
- B. `list`
- C. `dict` (dictionary)
- D. All of the above are legal data types for `x` with this syntax.

20. Which statement most accurately explains *why* the following function does *not* succeed in removing all elements from the list it receives as an argument? (2)

```
def clear(list_to_clear):  
    for index in range(len(list_to_clear)):  
        list_to_clear.pop(index)
```

- A. The `for` loop combined with `pop()` will skip elements.
- B. The function never returns the cleared list.
- C. The `pop()` function requires an element, not an index.
- D. `pop()` only returns a value, it does not remove it from the list.

21. Which of the following is *not* a legal dictionary **value**? (2)

- A. "apple"
- B. 5
- C. {"a":1, "b":2}
- D. All of the above are legal dictionary **values**.

22. **Challenge!** What is the *value* in **x** after the following code is executed? (2)

```
def recursive_fcn(num):  
    if num <= 0:  
        return num  
    else:  
        return num + recursive_fcn(num-3)  
  
x = recursive_fcn(5)
```

- A. -1
- B. 6
- C. 7
- D. 15

Fill-in-the-blank: Writing code

Fill in the blanks to complete the functions as their docstrings indicate. Each blank is worth **3 points**, and there are a total of 4 lines.

23. `def question_23 (transactions, holder_name):`
 `""" This function returns True if and only if the entry in the`
 `transactions dictionary for holder_name is a "buy" transaction.`
 `For example, given the transactions dictionary:`
 `transactions = {"Bob": ["sell", "AAPL", 5],`
 `"Jan": ["buy", "ANF", 10]}`
 `a holder_name argument of "Bob" would return False, whereas`
 `a holder_name argument of "Jan" would return True.`
 `"""`
 `result = False`

 `if _____ == "buy":` (3)
 `_____ # for full credit: modify result`

 `return result` (3)
24. `def question_24 (secret):`
 `""" This function should return the number of characters left to`
 `guess in the string secret, which are represented as a dash (-).`
 `For example:`
 `secret = "Wis--nsin"`
 `should return the integer value 2.`
 `"""`

 `count = 0`
 `for i in _____ :` (3)
 `if secret[i] == "-":`
 `_____ # for full credit: increment count` (3)
 `return count`

This page intentionally left blank.
Please leave it attached to your exam.