CS 301 - Fall 2017

Instructor: Laura Hobbes LeGault and Alexi Brooks

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 for those portions. Answers for Fill-in-the-blank questions must be marked on this exam.

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 your **lecture number**, fill in bubbles:
   001 - MWF 8:50a (Hobbes morning)
   002 - MWF 1:20p (Hobbes afternoon)
   003 - TR 9:30a (Alexi)
4. Under *F* of SPECIAL CODES, write **A** (exam version), fill in bubble **0**

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 23 questions and is worth a total of 43 points. You will have 50 minutes to complete the exam. **Be sure to read through every question completely.**

1. **Dual Choice** — 11 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
	`( <expression> )`	grouping with parentheses
higher	`x[index]`	indexing
	`* / %`	multiplicative
	`+ -`	additive
↕	`< <= > >=`	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 input as a string.
`len(s)`	Return the length (the number of items) of an object.
`sum(x)`	
`range(n)`	Returns a list of `n` consecutive integers beginning at `0`.
`range(a,b)`	Returns a list of consecutive integers beginning at `a`, ending before `b`.
`type(x)`	Returns the *data type* of the value stored in `x`

Constants, methods/functions from string and random modules:

`w.isalpha()`	Return True if all characters in `w` are letters, `w` not empty.
`w.isdigit()`	Return True if all characters in `w` are numerals, `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 <= N <= 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 and return 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. Given `x = [7, 2, -5, 4]`, slicing `x` using the code `x[1:3]` produces the list _____ . (1)

   A. `[2, -5]`

   B. `[2, -5, 4]`

2. **List comprehension** syntax uses a _____ loop. (1)

   A. `while`

   B. `for`

3. In the loop header `for A in B`, the value in _____ *must* be **iterable**. (1)

   A. `A`

   B. `B`

4. A key/value pair in a dictionary should be written as _____ . (1)

   A. `key:value`

   B. `key,value`

5. Given the following `for` loop **never** produces an error, `collection` must be a _____ . (1)

```
for x in collection:
    print (collection[x])
```

   A. list

   B. dictionary

6. To place a value at the **beginning** of an existing list, use the _____ method. (1)

   A. append

   B. insert

7. Given two list variables `A` and `B`, if I add a value to `A` and it changes the length of `B`, we say that `B` is a _____ copy of `A`. (1)

   A. deep

   B. shallow

8. A value that *can* be changed **in place** is called _____ . (1)

   A. mutable

   B. immutable

True or False: Evaluating boolean expressions

9. `"apple".isdigit()` (1)

   A. True

   B. False

10. `1 in {1:2, 3:4}` (1)

    A. True

    B. False

11. `x == "Y" or "N"` (1)

    A. True

    B. False

Multiple Choice: Reading code

12. Which of the following conditions correctly checks if the `massage_list` contains any massages to cancel, displaying the error message when there are none? (2)

```
def cancel(massage_list, index):
    if CONDITION:
        print ("Unable to cancel a massage, no massages scheduled.")
    else:
        # remove massage from list
```

    A. `index > 0`

    B. `index < range(len(massage_list))`

    C. `index in range(len(massage_list))`

    D. `len(massage_list) == 0`

13. Which of the following lines of code correctly adds a new power to the `ditto` dictionary's list of absorbed powers, stored under the key `"power"`? (2)

    A. `ditto["power"] = ditto.append("growl")`

    B. `ditto["power"].append("pulverizing pancake")`

    C. `ditto = ditto.append("growl")`

    D. `ditto.insert("power", "splash")`

14. Given a list, `level`, as follows: (2)

```
level = [{"favored":  True, "score":  50},
         {"favored":  False, "score":  74},
         {"favored":  True, "score":  32}]
```

which of the following options *best* describes the value in `x`?

```
x = sum( [e["score"] for e in level if e["favored"]==True] )
```

A. The score of the **last** favored employee

B. A **list** of all favored employees' scores

C. The **total** score of all favored employees

D. A **count** of all favored employees in the level

15. Which line of code produces the **same** value of `x`? (2)

```
x = []
for i in range(10):
   x.append(2**i)
```

A. `x = [for i in range(10) 2**i]`

B. `for 2**i in range(10):  x = [i]`

C. `x = [2**i for i in range(10)]`

D. `for i in range(10):  x = [2**i]`

16. What is the *value* in `x` after the following code is executed? **Be careful**, trace the code! (2)

```
def rounds_down(x):
   if x <= 1:
      return x
   return rounds_down(x/2) + rounds_down(x/2)

x = rounds_down(7)
```

A. 2

B. 3

C. 4

D. 7

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

```
nested = [[1, "this is", "only"], {"1":"test"}]
x = nested[1]
```

A. `int` (integer)

B. `str` (string)

C. `list`

D. `dict` (dictionary)

18. What is the *value* in `x` after this code executes? **Be careful**, trace the code! (2)

```
numbers = [2, 4, 4, 4, 7, 9]
i = 0
while i < len(numbers):
   if i % 2 == 0:
      numbers.pop(i)
   i = i +1
x = len(numbers)
```

A. This code produces an IndexError.

B. `2`

C. `3`

D. `4`

19. Which of the following is a valid dictionary **key**? (2)

A. `5.9`

B. `["cut", "splash", "growl"]`

C. `{"a":1, "b":2}`

D. None of these are valid dictionary keys.

20. What is the *type* of `x` after the following line of code is executed? Assume `random` has been imported. (2)

```
x = str(range(random.randint(1, 10))).isdigit()
```

A. `bool` (boolean)

B. `str` (string)

C. `list`

D. `int` (integer)

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

```
x = [10, 4, 1, 3].pop(1)
```

A. `None`, `list.pop()` does not return a value.

B. `10`

C. `4`

D. `1`

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.

22.
```
def total_points (word):
    """ This simplified function returns a word's score for a game.
        All words have a base score equal to their number of letters.
        Words which are palindromes receive 5x their base score.
            total_points("cat") => 3
            total_points("level") => 25
        ⇒ Assume the is_palindrome(word) function exists and returns
        True if and only if a word is a palindrome, False otherwise.
    """

    score = _____   # number of characters in word        (3)

    if _____:                                             (3)
        score = score*5
    return score
```

23.
```
def exam_avg (students, exams):
    """ Exams is a list of exam names (e.g.  ["e1","e2","e3"]).
        Students is a list of dictionaries, containing exam scores
        as floats between 0 and 1.  For example, one might be:
            {"e1":0.76, "e2":0.91, "e3":0.42}
        Code your for loops to print each EXAM average.
    """

    for _____ :                                           (3)
        total = 0

        for _____ :                                       (3)
            total += student[exam]
        print total / len(students)
```


This page intentionally left blank, as a back cover.

Please leave it attached to your exam.

