CS 301 - Spring 2018

Instructor: Laura Hobbes LeGault

Midterm Exam 3 — 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. 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 9:55a (Hobbes morning)
   - **002** - MWF 1:20p (Hobbes afternoon)
   - **003** - TR 1:00p (Paul Barford)
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 22 questions and is worth a total of 42 points. You will have 50 minutes to complete the exam. **Be sure to read through every question completely.**

1. **Dual Choice** — 10 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** — 2 blanks worth 3 points each, 1 worth 6 points. 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:index]`	slicing
	`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)`	Displays prompt `p` and returns the user's input as a string.
`len(s)`	Returns the length (the number of items) of an object.
`range(n)`	Returns a list of `n` consecutive integers beginning at `0`.

The random module:

`random.randint(a,b)`	Returns a random integer in `[a, b]`.
`random.shuffle(s)`	Randomly changes the order of the sequence `s` in place.

The os and sys modules:

`os.path.exists(p)`	Returns `True` if the file at path `p` exists, `False` otherwise.
`os.listdir(p)`	Returns a list containing the names of the entries in directory `p`.
`sys.argv`	The list of command line arguments passed to a Python script.

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`.
`dict.keys()`	Returns a copy of `dict`'s list of keys.
`dict.values()`	Returns a copy of `dict`'s list of values.

Files:

`open(p,m)`	Opens the file at path `p` in mode `m`, returning an object of type `file`.
`f.read()`	Returns the entire contents of the file object `f` as a string.
`f.readline()`	Returns the next line of the file object `f` as a string.
`f.readlines()`	Returns the entire contents of the file object `f` as a list of strings.
`f.write(s)`	Writes the string `s` to the file object `f`.
`f.close()`	Closes the file object `f`.

The numpy module (as np):

`np.append(a,x)`	Returns a copy of array `a` with value `x` appended to it.
`np.array(L,t)`	Returns the list `L` as an array containing elements of type `t`.
`np.arange(n)`	Returns an array of `n` integers from 0 to `n-1`.
`np.mean(a)`	Returns the mean (average) of the elements of array `a`.
`np.random.rand(n)`	Returns an array of `n` uniformly distributed floats in `[0,1)`.
`np.random.rand(a,b)`	Returns an `a`×`b` array of uniformly distributed floats in `[0,1)`.

The matplotlib.pyplot module (as plt):

`plt.plot(x,y)`	Plots `(x,y)` coordinates using default line style and color.
`plt.xlabel(s)`	Sets the x-axis label of the current plot to `s`.
`plt.ylabel(s)`	Sets the y-axis label of the current plot to `s`.
`plt.title(s)`	Sets the title of the current plot to `s`.
`plt.legend()`	Places a legend on the axes for all labelled lines.
`plt.show()`	Display a figure and pause until the figure has been closed.

The pandas module (as pd):

`pd.DataFrame(d)`	Returns a 2-dimensional data structure from data `d`. Optional argument: `columns` with labels for each column
`pd.read_csv(p)`	Reads CSV file at path `p` into DataFrame.
`df.max()`	Returns the maximum value in `df`. Optional argument: `axis` where 0 = cols, 1 = rows
`df.idxmax()`	Returns index of first occurrence of maximum. Optional argument: `axis` where 0 = cols, 1 = rows
`df.plot()`	Plots the values in `df` as line plots.
`df.plot(x,y)`	Plots `(x,y)` coordinates as a line plot.
`df.plot.scatter(x,y)`	Plots `(x,y)` coordinates as a scatter plot.
`df.plot.hist()`	Plots the values in `df` as a histogram.

Dual Choice: Terminology

1. Appending to a **NumPy array** adds the new element ________________ . (1)
   - A. in place
   - B. to a deep copy

2. The keyword for creating a **new error** in Python is ________________ . (1)
   - A. `return`
   - B. `raise`

3. Adding the option `"ro"` to the `plt.plot()` function creates ________________ . (1)
   - A. red dots
   - B. a red-orange line

4. If the current working directory is **empty**, `os.path.exists("./")` returns ________ . (1)
   - A. True
   - B. False

5. Code that runs **after an error occurs** should be placed in a ____________ block. (1)
   - A. `try`
   - B. `except`

6. `df` is a DataFrame with two numeric columns. `df.plot()` with **no arguments** produces a graph with ________________ . (1)
   - A. one line
   - B. two lines

7. The function `pd.read_csv(filename)` reads the contents of a CSV and returns a value of type ________________ . (1)
   - A. pandas `DataFrame`
   - B. numpy `array`

8. Looping through a file object `f`, opened in read mode, is equivalent to looping through the value returned by ________________ . (1)
   - A. `f.readlines()`
   - B. `f.read()`

9. I'm studying whether your lecture section influences your grades on exams. A student's **section** is an example of their ____________________ in this study. (1)

   A. treatment

   B. outcome

10. If the function call `open("./e3.txt","r")` does **not** cause an error, the file `e3.txt` must exist and is located in the ____________________ . (1)

    A. current working directory

    B. root directory

Multiple Choice: Reading code

11. Given a file formatted as follows: (2)

```
LEN_PETAL,WID_PETAL,LEN_SEPAL,WID_SEPAL
5.1,3.5,1.4,0.2
4.9,3,1.4,0.2
4.7,3.2,1.3,0.2
```

what is the *data type* of `x` after the following code executes?

```python
import csv
with open(filename, "r") as f:
   reader = csv.DictReader(f)
   for row in reader:
      x = row
```

   A. `list`

   B. `dict` (dictionary)

   C. `str` (string)

   D. `float`

12. Which of the following `if` statements could be used in place of the `try` statement to prevent an `IndexError` from occurring? Assume `index` is a positive integer (e.g. 10). (2)

```
x = raw_input("Enter a word: ")
try:
  print x[index]
except IndexError:
  print "Not enough letters!"
```

A. No `if` necessary, `print x[index]` will never cause an IndexError.

B. `if len(x) < index:`

C. `if len(x) == index:`

D. `if index < len(x):`

13. Given the following pandas DataFrame `df`, containing five (5) students' (randomly-generated) exam scores: (2)

```
      e1     e2
0  32.68  25.20
1  40.21  41.11
2  39.62  36.47
3  24.91  36.94
4  30.02  30.16
```

In how many rows does the variable `x`, defined as vollows, contain `True`?

```
x = df["e2"] > df["e1"]
```

A. 5

B. 3

C. 2

D. `x` contains only a single boolean value, `True`.

14. In studying the association between coffee and cancer, which of the following statements (if true) would **not** be a potential *confounding factor*? (2)

A. People of all genders drink coffee.

B. People tend to smoke on their coffee breaks.

C. Only old people drink coffee.

D. 80% of non-coffee drinkers and 10% of coffee drinkers drink green tea.

15. Which of the following lines will **not** run if `ERROR LINE` **does not** cause any error? (2)

```
try:
   ERROR LINE
   LINE A
except TypeError:
   LINE B
LINE C
```

A. `LINE A`
B. `LINE B`
C. `LINE C`
D. The program will crash without running any of these lines.

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

```
import numpy as np
x = np.array([1,2]) + np.array([5,10])
```

A. `None`, this code causes a TypeError.
B. `[1,2,5,10]`
C. `[[1,2], [5,10]]`
D. `[6,12]`

17. What is the *value* in `x` after the following code executes? Assume pandas has been imported as `pd` and numpy as `np`. (2)

```
df = pd.DataFrame(np.random.rand(10,2), columns=["x","y"])
df["y"] = df["y"]+2
x = df.idxmax(axis=1)
```

A. A pandas `Series` containing 10 rows of `"y"`
B. A deep copy of `df["y"]`
C. `"y"`
D. Impossible to tell without knowing the random values generated.

18. What is the *data type* of `x` after the following code executes? (2)

```
import numpy as np
x = np.array([1,2,3], float)[1]
```

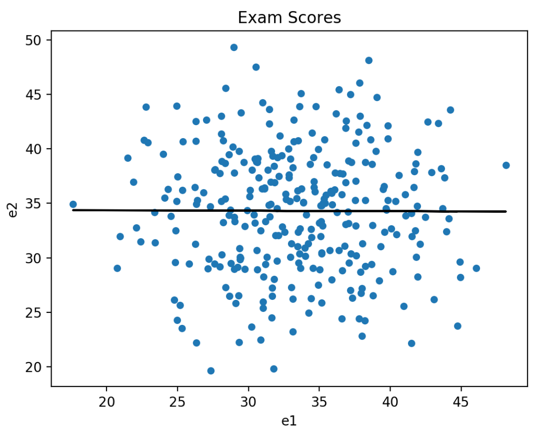
A. `array`

B. `int`

C. `float`

D. `None`, this code causes a TypeError.

19. Which of the following statements was **not** present in the code which created the image file `figure.png`, shown here? Assume `matplotlib.pyplot` and `pandas` were imported as `plt` and `pd`, respectively, and that the data shown is stored in the DataFrame `df`. (2)



A. `plt.show()`

B. `plt.title("Exam Scores")`

C. `df.plot("e1", "e2")`

D. `plt.plot(df["e1"], m*df["e1"]+b)`

20. Which of the following import statements will cause this line to work without error? (2)

```
nums = rand(5)
```

A. `from numpy.random import rand`

B. `import numpy.random as rand`

C. `import numpy as np`

D. No `import` syntax allows NumPy's `rand()` function to be called like this.

Fill-in-the-blank: Writing code

For each of the blanks in question 20, fill in the statement needed to produce the indicated output. Each line is worth **3 points**.

21.
```
import random

with ______________________________:     # write to q20.txt          (3)
    for i in range(10):
        # write a random int 1-100 on each line of the file

        outfile.write(______________________________)                (3)
```

For the last question, you will write a complete function.

22. Write a function called `filter` which takes a `color` as a string and a list of CSV `lines` as dictionaries. Your function should create a new, empty list, add every dictionary from the `lines` which contains the value in `color` associated with the key `"HUE"`, and then return this new list of dictionaries. (6)

You may assume every dictionary in `lines` has the following format:

```
{ "HUE":"red", "U":9.2, "G":0.07, "R":10.72, "I":8.3, "Z":8.0 }
```

Draco dormiens nunquam titillandus.