

CS 301 - Spring 2016
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. 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 001 (morning lecture), fill in bubbles
4. Under J 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 21 questions and is worth a total of 84 points. You will have 50 minutes to complete the exam. **Be sure to read through every question completely.**

1. **Dual Choice** — 8 questions worth 2 points each.
2. **Multiple Choice** — 11 questions worth 4 points each. Choose the *best* answer.
3. **Fill-in-the-blank** — 2 questions worth 12 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:index]	slicing
	x[index]	indexing
	* / %	multiplicative
	+ -	additive
	< <= > >=	relational
	== !=	equality
	not	logical not
lower	and	logical and
	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)` Return the length (the number of items) of an object.
`range(n)` Returns a list of `n` consecutive integers beginning at 0.
`min(x)` Returns the smallest item in the iterable `x`.
`max(x)` Returns the largest item in the iterable `x`.

Constants and functions from the math module:

`math.pow(x,y)` Returns `x` raised to the power `y`. Converts both arguments to floats.
`math.pi` The mathematical constant $\pi = 3.141592\dots$

Functions from the os module:

`os.path.exists(p)` Returns `True` if the file at path `p` exists, `False` otherwise.

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

Files:

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

Functions from the numpy module (as np):

<code>np.array(L,t)</code>	Returns the list <code>L</code> as an array containing elements of type <code>t</code> .
<code>np.arange(n)</code>	Returns an array of <code>n</code> integers from 0 to <code>n-1</code> .
<code>np.mean(a)</code>	Returns the mean (average) of the elements of array <code>a</code> .
<code>np.std(a)</code>	Returns the standard deviation of the elements of array <code>a</code> .
<code>np.random.rand(n)</code>	Returns an array of <code>n</code> uniformly distributed floats between 0 and 1.

Functions from matplotlib.pyplot (as plt):

<code>plt.plot(x,y)</code>	Plots <code>(x,y)</code> coordinates using default line style and color.
<code>plt.xlabel(s)</code>	Sets the x-axis label of the current plot to <code>s</code> .
<code>plt.ylabel(s)</code>	Sets the y-axis label of the current plot to <code>s</code> .
<code>plt.title(s)</code>	Sets the title of the current plot to <code>s</code> .
<code>plt.hist(a,n)</code>	Divides the values in <code>a</code> into <code>n</code> bins and plots a histogram.
<code>plt.pie(a)</code>	Creates a pie chart with wedges proportional to the values in <code>a</code> .
<code>plt.show()</code>	Display a figure and pause until the figure has been closed.

A or B: Terminology

1. The code
`total += a.vol`
adds _____ to the variable `total`.
A. the value of `a`'s `vol` property
B. the return value from `a`'s `vol` method (2)
2. A `try` block is always followed by a(n) _____ block.
A. `except`
B. `catch` (2)
3. When creating an array from a list, all elements must be _____.
A. able to be cast to the same type
B. of the same type (2)
4. The command
`pip install numpy`
will download and install NumPy when run _____.
A. in the Python interpreter
B. on the command line (2)
5. `random.shuffle(deck)` is an example of a _____.
A. method
B. function (2)
6. The `__repr__()` method should _____ an object's string representation.
A. `print`
B. `return` (2)
7. `plt.show()` should be called _____ any other calls to modify a plot.
A. after
B. before (2)
8. `my_list.append(4)` is an example of a _____.
A. method
B. function (2)

Multiple Choice: Reading code

9. A program you are running displays the following output in the console: (4)

```
<__main__.Animal instance at 0x02855328>
```

Which of the following statements about this program is necessarily **true**?

- A. The program crashed.
- B. The program stores Animal objects in a dictionary.
- C. The Animal class does not contain a `__repr__()` method.**
- D. The Animal class `__init__()` method expects a numeric argument.

10. Choose the correct mode so that the following code will add the current date and time to the end of a file *without erasing any previous entries*: (4)

```
import datetime
with open("datefile.txt", MODE) as f:
    f.write(str(datetime.datetime.now())) # gets current time
```

- A. MODE: "a"
- B. MODE: "rb"
- C. MODE: "r"
- D. MODE: "w"

11. After the following code has run, what is the *type* of the variable `x`? Assume numpy has been imported as `np`. (4)

```
x = np.array(["1","2","3","4","5"], float)[0]
```

- A. array
- B. string
- C. float**
- D. This code will produce a ValueError.

12. What sort of graph is displayed by the following complete program? (4)

```
import numpy as np
import matplotlib.pyplot as plt

plt.plot(np.array([0,1,2]),np.array([1,2,3]))
plt.plot(np.arange(12))
plt.hist(np.random.rand(1000))
```

- A. A line graph
- B. A pie graph
- C. A histogram
- D. No output is displayed**

13. NumPy's documentation includes the following entry for `numpy.random.randint()`: (4)

`numpy.random.randint(low, high=None, size=None)`

Returns a *size*-shaped array of random integers (or a single such random int if *size* not provided) from the “discrete uniform” distribution in the “half-open” interval `[low, high)`. If *high* is `None` (the default), then results are from `[0, low)`.

Which of the following is **not** a legal call to `numpy.random.randint()`?

- A. `numpy.random.randint(5, size=(2,4))`
- B. `numpy.random.randint(10)`
- C. `numpy.random.randint()`
- D. `numpy.random.randint(high=5, low=2)`

14. Which of the following correctly creates an instance of the `Animal` class with the species `Moose`? (4)

- A. `Animal('Moose')`**
- B. `new Animal('Moose')`
- C. `Animal.__init__(Animal, 'Moose')`
- D. `Animal.__repr__(Animal, 'Moose')`

15. If **no errors** occur in the following code, which line will **not** be executed?

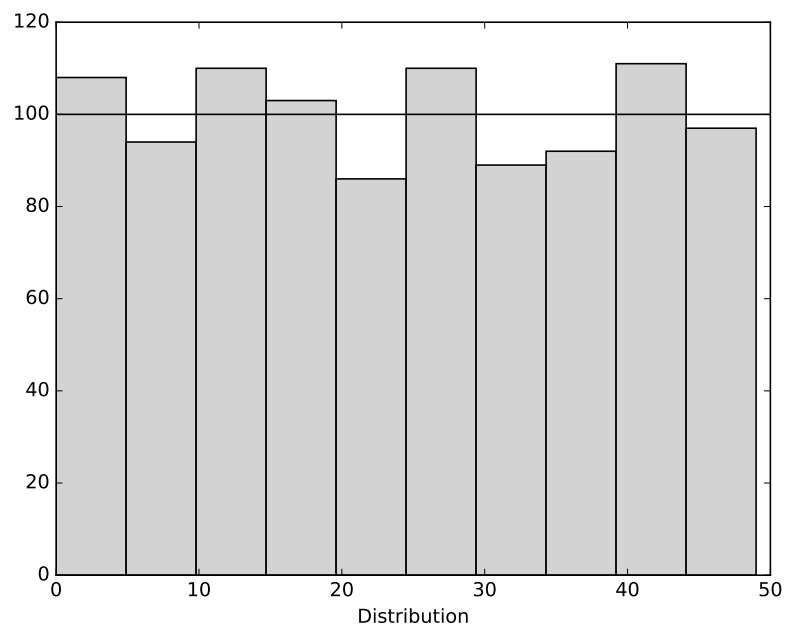
(4)

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

- A. LINE A
- B. LINE B**
- C. LINE C
- D. All lines of code will run.

16. Which of the following commands was **not** used to create the plot below?

(4)



- A. `plt.show()`
- B. `plt.plot([0,1], [100,100])`
- C. `plt.hist(numpy.random.randint(0,50,1000))`
- D. `plt.title("Distribution")`**

17. Which of the following is **not** an example of a *relative* path? (4)

- A. hamilton.mp3
- B. C:\Users\Documents\design.docx
- C. ./federalist.txt
- D. CS301/Lab1/lib/triangles.py

18. Why might the following code cause an error? (4)

```
f = open('doesnotexist.txt', 'w')
s = f.read()
```

- A. The file `doestnotexist.txt` does not exist.
- B. Bad syntax; the first line should be `f.open('doesnotexist.txt', 'w')`.
- C. If `doesnotexist.txt` is too large, `read()` will cause a memory overflow.
- D. **The file was opened in write mode and cannot be read.**

19. Given the following list of imported modules from the beginning of a file, which function call is *not* legal? (4)

```
import numpy
import matplotlib.pyplot as plt
import random
```

Pay attention to module names - all functions are otherwise called and used correctly.

- A. `num = [random.randint(1,10) for i in range(500)]`
- B. `a = np.array(num,int)`
- C. `a = numpy.random.randint(1,11,size=500)`
- D. `plt.plot(a)`

Fill-in-the-blank: Writing code

For each of the questions on this page, fill in the value, operator, or statement needed to produce the indicated output. Each line is worth **6 points**.

```
20.  def decipher( D, s ):
        """ D is a Caesar Cipher translation dictionary, where the encoded
            (uppercase) letter is the key and the decoded (uppercase) letter
            is the value.
            Complete the function so that it returns the decoded version of
            encoded (uppercase) string s.  """

        newstring = ""           # will contain the decoded string

        for letter in _____:           (6)
            if letter.isalpha():
                c = _____           (6)
            else:
                c = letter
            newstring += c
        return newstring
```

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

21. Write a function named `question_21` that expects a list of `Animal` objects and a domain (either `"land"` or `"water"`) and returns the number of animals with that domain in the list. (12)

Recall that instances of the `Animal` class have a `species`, a `domain`, and either an `area` or `vol` depending on whether they are land or water animals respectively. The `Animal` class also contains a `get_requirement()` method, which returns either `area` or `vol` depending on the animal's domain.

You do not *need* to comment your code, but recall that your grade is based on how well I can understand what you're trying to do.

This page intentionally left blank.