

DON'T
WRITE
ANYTHING
HERE!!!



Name:

(as it would appear on official course roster)

Umail address:

@umail.ucsb.edu

1

e01

CS8 S18

EXAM: e01: Midterm Exam

ready?	date	points
true	Tue 04/24 03:30PM	100

You may not collaborate on this exam with anyone. If you need to use the restroom, you must leave your cell phone with the exam proctor before leaving the room.

- Write your name at the top of this page **AND EVERY ODD NUMBERED PAGE**.
- Double check that you turned in ALL pages; look for "End of Exam" on the last page.
- This exam is **closed book, closed notes, closed mouth, cell phone off**.
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes.
- This sheet will be collected with the exam, and might not be returned.
- Please write your name on your notes sheet.

NOTE: All references to **Python** on this exam mean **Python 3**, so you should answer accordingly.

1. (10 pts) Many products in the United States that are sold by weight are measured in ounces, while in most other countries, they are measured in grams. One ounce is approximately 28.3495 grams, and the abbreviation for ounce is **oz**.

Given this information, write the definition of a Python function called `oz2g` that takes a parameter called `ounces` and returns the corresponding value in grams.

2. (5 pts) Now write a function call to this function, as you might type it at the Python prompt, for example, that computes the number of grams equal to 3 ounces.

2

e01

CS8 S18

3. Testing a Python function can be done with the `pytest` module, as we did in lab02 and lab03 in this course. The [handout](#) you got with this exam has example code illustrating `pytest` test cases.

a. (4 pts) Why do we need the `pytest.approx` thing? That is, why can't we just write:
`assert cToF(100.0)==212.0?`

b. (6 pts) Now consider the `oz2g` function you write earlier. Please write two test cases for that function appropriate for use with `pytest`. For convenience, here is a table of two values you may use:

ounces	grams
2.5	70.8738
3	85.0486

c. (5 pts) When using test cases, we often write a “stub” before writing the correct function definition. Write a function definition for `oz2g` that illustrates this practice—that is, a function definition for `oz2g` with a “stub”.

DON'T
WRITE
ANYTHING
HERE!!!



DON'T
WRITE
ANYTHING
HERE!!!



Name:

(as it would appear on official course roster)

Umail address:

@umail.ucsb.edu

4. For the Python code in the left box, write the output in the right box

NOTE: The only difference between the two questions is that in one case, `item` has no quotation marks around it, while in the other case it does ("`item`"). That's on purpose. I want you to tell me what output Python will give in each of these two cases. If you are not sure, write your best guess.

(3 pts)	<pre>the_list = ["CS", "Math", "PStat"] for item in the_list: print(item, end=" ")</pre>	
(3 pts)	<pre>the_list = ["CS", "Math", "PStat"] for item in the_list: print("item", end=" ")</pre>	

3

e01

CS8 S18

4

e01

CS8 S18

5. Assuming the following assignment statements have been entered at the Python prompt:

```
school = "UC Santa Barbara"
abbrev = "UCSB"
course = "CS8"
qtr = "M17"
unis = ["UCSB", "UCLA", "Harvard", "Yale", "Stanford"]
cols = ["red", "green", "blue"]
cols.append("purple")
cols.append("blue")
```

DON'T
WRITE
ANYTHING
HERE!!!



Indicate the value of each of these expressions.

IMPORTANT: Use strict Python syntax. If it's a string, put it in quotes. Boolean values should be indicated as `False` and `True`, no quotes, with the capital letter.

	Expression	Result		Expression	Result
(3 pts)	<code>2 + 3 * 5</code>		(3 pts)	<code>10 % 3</code>	
(3 pts)	<code>abbrev * 2</code>		(3 pts)	<code>len(qtr)</code>	
(3 pts)	<code>qtr[0:2]</code>		(3 pts)	<code>unis[1]<unis[2]</code>	
(3 pts)	<code>course[-1]</code>		(3 pts)	<code>len(unis) <= 8</code>	
(3 pts)	<code>len(cols)</code>		(3 pts)	<code>cols[-1][-1]</code>	
(3 pts)	<code>cols[0]</code>		(3 pts)	<code>cols[2][0]</code>	
(3 pts)	<code>len(cols[1])</code>		(3 pts)	<code>"e" in cols[0]</code>	

DON'T
WRITE
ANYTHING
HERE!!!



Name:

(as it would appear on official course roster)

Umail address:

@umail.ucsb.edu

6. (4 pts) A complete Python program that prints out the following message is a very short program indeed:

```
I can code!
```

Please write that entire program in the space below. You certainly have more space than you will need.

7. (2 pts) To start work on a Python program you need to type a command that brings up the Integrated Development Environment for Python (version 3). What is this command?

8. (2 pts) What does the Unix `mkdir` command do?

5

e01

CS8 S18

6

e01

CS8 S18

9. Indicate with a check what Python will return as the type of each expression.

Note that when Python reports a type, it reports it with expressions such as `<class 'int'>` and `<class 'str'>`.

I'm omitting the `<class '___' >` part to save space.



a. (2 pts) `type("UCSB")`

<input type="checkbox"/> str	<input type="checkbox"/> int	<input type="checkbox"/> float	<input type="checkbox"/> list	<input type="checkbox"/> tuple	<input type="checkbox"/> dict	<input type="checkbox"/> bool
------------------------------	------------------------------	--------------------------------	-------------------------------	--------------------------------	-------------------------------	-------------------------------

b. (2 pts) `type(12)`

<input type="checkbox"/> str	<input type="checkbox"/> int	<input type="checkbox"/> float	<input type="checkbox"/> list	<input type="checkbox"/> tuple	<input type="checkbox"/> dict	<input type="checkbox"/> bool
------------------------------	------------------------------	--------------------------------	-------------------------------	--------------------------------	-------------------------------	-------------------------------

c. (2 pts) `type(1 + 2)`

<input type="checkbox"/> str	<input type="checkbox"/> int	<input type="checkbox"/> float	<input type="checkbox"/> list	<input type="checkbox"/> tuple	<input type="checkbox"/> dict	<input type="checkbox"/> bool
------------------------------	------------------------------	--------------------------------	-------------------------------	--------------------------------	-------------------------------	-------------------------------

d. (2 pts) `type("24")`

<input type="checkbox"/> str	<input type="checkbox"/> int	<input type="checkbox"/> float	<input type="checkbox"/> list	<input type="checkbox"/> tuple	<input type="checkbox"/> dict	<input type="checkbox"/> bool
------------------------------	------------------------------	--------------------------------	-------------------------------	--------------------------------	-------------------------------	-------------------------------

e. (2 pts) `type([3] + [4])`

<input type="checkbox"/> str	<input type="checkbox"/> int	<input type="checkbox"/> float	<input type="checkbox"/> list	<input type="checkbox"/> tuple	<input type="checkbox"/> dict	<input type="checkbox"/> bool
------------------------------	------------------------------	--------------------------------	-------------------------------	--------------------------------	-------------------------------	-------------------------------

f. (2 pts) `type(3 < 4)`

<input type="checkbox"/> str	<input type="checkbox"/> int	<input type="checkbox"/> float	<input type="checkbox"/> list	<input type="checkbox"/> tuple	<input type="checkbox"/> dict	<input type="checkbox"/> bool
------------------------------	------------------------------	--------------------------------	-------------------------------	--------------------------------	-------------------------------	-------------------------------

g. (2 pts) `type({"green": "verde"})`

<input type="checkbox"/> str	<input type="checkbox"/> int	<input type="checkbox"/> float	<input type="checkbox"/> list	<input type="checkbox"/> tuple	<input type="checkbox"/> dict	<input type="checkbox"/> bool
------------------------------	------------------------------	--------------------------------	-------------------------------	--------------------------------	-------------------------------	-------------------------------

End of Exam