

Closed book, closed notes, log out of computer! Please write neatly!

4.1 For the following, indicate whether the expression will evaluate to True or False. You can assume the code does not generate a type error but cannot assume anything about the value of any variables. A satisfactory answer requires 3/4 correct.

	True	False
<code>1 == 1.0</code>	☐	☐
<code>not True or False</code>	☐	☐
<code>"hello"[:2] == "he"</code>	☐	☐
<code>(value == 0) and (value != 0)</code>	☐	☐

4.2. For each of the following code snippets, enter the final value for `x` on the line:

a. _____ <pre>x = 6 if x >= 2: x = 2 else: x = 4 if x < 4: x = 8</pre>	b. _____ <pre>x = 2 if x > 6: x = 6 elif x > 4: x = 4 elif x > 2: x = 2</pre>
---	---

4.3 For each of the following while loops indicate whether the loop is guaranteed to terminate or not execute (indicate with “T”), is guaranteed to be an infinite loop (indicate with “I”) or depends (indicate with “D”). A satisfactory answer requires 3/4 correct.

a. _____ <pre>i = 0 while i < 10: i = i + 1</pre>	b. _____ <pre>i = 10 while i < 5: i = i - 1</pre>
c. _____ <pre>i = "a" while len(i) > 0: i = i + "a"</pre>	d. _____ <pre>from random import randint i = 5 while i < 10: i = i - randint(1, 5)</pre>

CS 146 Quiz 1 Retest 10/4/24

1.1. Which of the following instructions in a recipe show why recipes can be an imperfect analogy for an algorithm in a Computer Science context? *Select all that apply.*

- ☐ Bake until the internal temperature is 165 °F.
- ☐ Cut into bite-sized pieces.
- ☐ Add 10 grams of all-purpose flour.
- ☐ Preheat the oven to 325 °F.
- ☐ Mix thoroughly.

1.2. Write (in the box) the value of `x` after the code below executes?

```
x = 4
y = 2 * x
y = 3
y = 3
x = y + x
# x = 2 * x
```

1.3. Evaluate the following expressions and indicate if an error would occur, or if it is a valid Python expression, indicate what the value would be. Make the type of any values clear by showing quotes or decimal portions as relevant.

(a) `"3" + "2"`

(b) `"3" * "2"`

(c) `1 + 3 / 4`

2.2. Write an appropriate docstring for the following function:

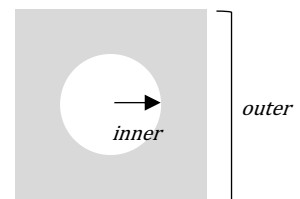
```
import turtle as t
def mystery(n, l):
    """
```

```
    """
    for i in range(n):
        t.forward(l)
        t.left(360/n)
```

2.3. Below each snippet report the value you would assign to the variable `n` to so that `total` has the specified value after the code executes. Each snippet should be considered independently. The value for `n` must be an integer that results in valid Python code (i.e., not cause an error).

<pre>total = n for i in range(2): for j in range(3): total = total + 1</pre>	<pre>total = 0 for i in range(3): total = total + 1 for j in range(n): total = total + 1</pre>
a) total is 12	b) total is 8

2.1. Write a function named `outset_area` that has two parameters `inner`, `outer`, representing the inner hole radius and outer side length of a square with a hole, and returns the area of region shaded grey below. Recall that the area surrounding the hole is $outer^2 - \pi \cdot inner^2$. You can assume $outer > 2 \cdot inner$ and there exists a constant `PI` set to the value of π . You do not need to include comments or docstrings.



```
>>> outset_area(1, 4)
12.858
```

3.1 Enter in the box (right) what each code block would print when run via the green arrow in Thonny. Each block should be considered independently. A satisfactory answer requires 2/3 correct.

<pre>l = ["a", "b", 3] l.reverse() print(l[2])</pre>	
<pre>s = "Midd" print(s.lower().replace("M", "dd"))</pre>	
<pre>s = "a" + "b".upper() s.lower() print(s)</pre>	

3.2. Evaluate the following expressions, writing the resulting string into the boxes, one character per box (recall a space is a character), and shading in unused boxes to indicate the end of the string (the empty string would have all boxes shaded). The value of `astring` is `"cs 4ever!"`. A satisfactory answer requires 2/3 correct.

For example: `astring`

c	s		4	e	v	e	r	!		
---	---	--	---	---	---	---	---	---	--	--

`astring[:4]`

--	--	--	--	--	--	--	--	--	--	--

`astring[2::2]`

--	--	--	--	--	--	--	--	--	--	--

`astring[-6]`

--	--	--	--	--	--	--	--	--	--	--

3.3 For the following, indicate whether each block is valid Python if `value` is both a string and a list, only a string, only a list or valid for neither a list nor a string. You can assume `value` is already assigned and not empty. A satisfactory answer requires 2/3 correct.

	Both	Only string	Only list	Neither
<code>value[0]</code>	☐	☐	☐	☐
<code>value[0] + "a"</code>	☐	☐	☐	☐
<code>value[0] = "a"</code>	☐	☐	☐	☐