Q1 Honor Code and Preliminaries
0 Points

Directions and Notes:

During this exam you will have 3 hours. Note that there won't be TA support available, so do your best if you're not sure about a question.

Along with this exam, you may reference your *handwritten* cheatsheets, but no other outside materials. (A digital version of handwritten notes would be OK.)

You may have a digital version of the CS88 Reference Sheet and your handwritten notes, but no other files.

Reminder: Please turn on your camera and share you *entire* screen. Do not share only your browser window.

Good luck!

> As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others.

— *UC Berkeley Honor Code*

By typing my name below, I am affirming that all work in this final is my own. I have not (and will not) use any resources outside of the allowed materials, nor will I collaborate with anyone else during the exam.

Q2 Conceptual Questions
20 Points

Select the correct answer for each of these questions.

Q2.1
2 Points

True or False: A *generator* allows you to represent a finite or infinite stream of data.

True

False

Q2.2
2 Points

What Python *keyword* turns a function into a generator?

next

return

generate

yield

Q2.3
2 Points

True or False: The `reduce` function in Python *can* return a sequence.

True

False

Q2.4
2 Points

True or False: The `filter` function in Python *always* returns a sequence.

True

False

Q2.5
2 Points

True or False: When an *exception* is raised by some function, our program *always* stops executing.

True

False

Q2.6
2 Points

In the "Maps" project, you primarily practiced which programming paradigm?

Imperative Programming

Array-Based Programming

Functional Programming

Object-Oriented Programming

Declarative Programing

Q2.7
2 Points

In the "Ants" project, you primarily practiced which programming paradigm?

Imperative Programming

Array-Based Programming

Functional Programming

Object-Oriented Programming

Declarative Programing

Q2.8
2 Points

Of these items, which1 does every recursive function need?

return statement

yield statement

if statement

base case

Q2.9
2 Points

Given a color Abstract Data Type, answer some questions based on this definition:

```
def color(r, g, b):
    return (r, g, b)

r = lambda color: color[0]
g = lambda color: color[1]
b = lambda color: color[2]
```

Does the g function violate the abstraction barrier?

Yes

No

Q2.10
2 Points

Consider a new function combine_colors with the following definition:

```
def combine_colors(color1, color2):
    return color(
        r(color1) + r(color2),
        g(color1) + g(color2),
        b(color1) + b(color2)
    )
```

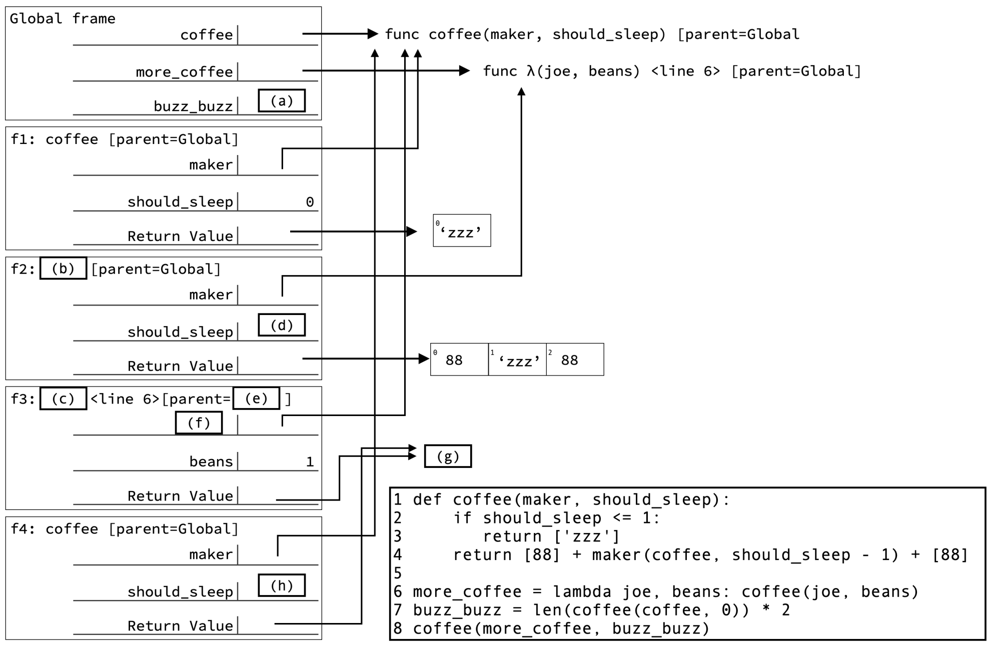
Does the combine_colors function violate the abstraction barrier?

Yes

No

Q3 Coffee, Coffee, ☕
19 Points

Fill in the blanks to complete the environment diagram. All the code used is in the box to the right, and the code runs to completion with no errors.



```
1 def coffee(maker, should_sleep):
2     if should_sleep <= 1:
3        return ['zzz']
4     return [88] + maker(coffee, should_sleep - 1) + [88]
5
6 more_coffee = lambda joe, beans: coffee(joe, beans)
7 buzz_buzz = len(coffee(coffee, 0)) * 2
8 coffee(more_coffee, buzz_buzz)
```

Q3.1
3 Points

What is the value of **box (a)**?

0

1

2

['zzz']

Q3.2
2 Points

What is the value of **box (b)**?

λ

coffee

more_coffee

maker

Q3.3
2 Points

What is the value of **box (c)**?

λ

coffee

more_coffee

maker

Q3.4
2 Points

What is the value of **box (d)**?

0

1

2

['zzz']

Q3.5
2 Points

What is the value of **box (e)**?

Global

coffee

f1

f2

Q3.6
2 Points

What is the value of **box (f)**?

joe

coffee

maker

Q3.7
2 Points

What is the value of **box (g)**?

0

[]

['zzz']

[88, 88]

Q3.8
2 Points

What is the value of **box (h)**?

0

1

2

['zzz']

Q3.9
2 Points

What is the final result of line 8, coffee(more_coffee, buzz_buzz) ?

['zzz']

[88, 88]

[88, 'zzz', 88]

[88, 88, 'zzz', 88, 88]

Q4
10 Points

Which of the following HOFs should we use to solve this problem? Assume we can't use sort() , or len() or any other methods that work on a list directly, but any valid map function, filter function, etc. could be used.

If we say " reduce first" then map , we mean that the reduce call happens before we do a map. e.g.

```
value = reduce(some_function, sequence)
result = map(..., ...)
```

We can use value as a part of our mapper OR as our sequence.

Q4.1
2 Points

Problem:
Input: A list of words
Output: A sequence with words whose first letter is a vowel.

map

filter

reduce

map first then reduce

reduce first then map

reduce first then filter

filter first then map

filter first then reduce

Not Possible

Q4.2
2 Points

Problem:

Input: A list of numbers

Output: The product of all numbers which are prime.

map

filter

reduce

map first then reduce

reduce first then map

reduce first then filter

filter first then map

filter first then reduce

Not Possible

Q4.3
2 Points

Problem:

Input: A list of numbers

Output: All the elements in a list whose value is within 5 of the largest value in the list. (i.e. if the largest item were 10, find all the items > 5. You do not know the largest item.)

map

filter

reduce

map first then reduce

reduce first then map

reduce first then filter

filter first then map

filter first then reduce

Not Possible

Q4.4
2 Points

Problem:
Input: A list of words, e.g. ['once', 'upon', 'a', 'time']
Output: A list of all the letters used, without duplicates, e.g.
['e', 'm', 't', 'n', 'c', 'i', 'p', 'a', 'o', 'u']

map

filter

reduce

map first then reduce

reduce first then map

reduce first then filter

filter first then map

filter first then reduce

Not Possible

Q4.5
2 Points

Problem: Word Count

Input: A list of words, e.g. ['once', 'upon', 'a', 'time']

Output: A dictionary of the number of times each word is used, e.g. {'once': 1, 'upon': 1, 'a': 1, 'time': 1}

map

filter

reduce

map first then reduce

reduce first then map

reduce first then filter

filter first then map

filter first then reduce

Not Possible

Q5
17 Points

Consider the following Tree class:

```
class Tree:
    name = 'Tree'

    def __init__(self, value, branches=()):
        self.value = value
        for branch in branches:
            assert isinstance(branch, Tree), "Branches must be trees"
        self.branches = list(branches)

    def __repr__(self):
        """e.g. Tree(1, [Tree(2)])"""
        if self.branches:
            branches_str = ', ' + repr(self.branches)
        else:
            branches_str = ''
        return f'{self.name}({self.value}{branches_str})'

    def is_leaf(self):
        return not self.branches
    def add_branch(self, tree):
        assert isinstance(tree, Tree)
        self.branches.append(tree)
```

Q5.1
3 Points

Let's say we wanted to make a new class `LinkTree` that enforced that we have only *one* branch. Which of the following class definitions should we use?

This class should inherit from a `Tree` and have all the methods of a `Tree`, but ensure that there's ever only 1 branch. Invalid `LinkTree`s should raise a `LinkTreeError`.

```
class LinkTreeError(Exception):
    pass


class LinkTree( ___(a)___ ):
    name = 'LinkTree'
    def __init__(self, value, branches=()):
        self.value = value
        for branch in branches:
            assert isinstance(branch, ___(b)___)
        if len(branches) > 1:
            ___(c)__ LinkTreeError('LinkTree can only have one branch')
        self.branches = list(branches)

    def add_branch(self, item):
        ___(some changes omitted)___
        assert isinstance(item, LinkTree)
        self.branches.append(item)
```

What should go in the space ___(a)___ ?
If nothing should go in that space, write 'nothing'.

Q5.2
2 Points

What should go in the space ___(b)___?
If nothing should go in that space, write 'nothing'.

Q5.3
2 Points

What should go in the space ___(c)___?
If nothing should go in that space, write 'nothing'.

Q5.4
3 Points

We want to adapt `reduce` to work on `Tree`s. We're going to add a new method to our class.

```
from functools import reduce
def reduce(self, func):
    """Reduce the values of a tree to a single value.
    >>> tree = Tree(1, [Tree(2), Tree(3)])
    >>> tree.reduce(lambda x, y: x + y)
    6
    >>> tree = Tree(1, [Tree(2, [Tree(3)]), Tree(4)])
    >>> tree.reduce(lambda x, y: x * y)
    24
    """
    if ___(a)___.is_leaf():
        return self.value
    else:
        results = [___(b)___] + [___(c)___ for branch in self.branches]
        return reduce(___(d)___, results)
```

What should go in the space ___(a)___?
If nothing should go in that space, write 'nothing'.

Q5.5
2 Points

What should go in the space ___(b)___?

self.value

value

self

self.branches

Q5.6
3 Points

What should go in the space ___(c)___?

self.reduce(func, branch)

self.reduce(func)

branch.reduce(func)

self.reduce(branch)

Q5.7
2 Points

What should go in the space ___(d)___?

self.value

self.func

self

func

Q6 Efficiency
6 Points

For each of the following questions, select the time complexity of the code provided. Assume that the input is nonnegative.

Q6.1
2 Points

```
def five_n_plus_two(n):
    if n == 0:
        return 2
    else:
        return 5 + five_n_plus_two(n - 1)
```

$O(1)$

$O(n)$

$O(n^2)$

$O(2^n)$

Q6.2
2 Points

```
def five_n_plus_two(n):
    s = 2
    for _ in range(5):
        s += n
    return s
```

$O(1)$

$O(n)$

$O(n^2)$

$O(2^n)$

Q6.3
2 Points

```
def add_numbers(n):
    s = 0
    for i in range(1, n):
        for z in range(1, i):
            s += z
    return s
```

$O(1)$

$O(n)$

$O(n^2)$

$O(2^n)$

Q7 Generating Generators Generates Generators
4 Points

We're going to write a function that turns another function into a generator that can be called max_times .

```
def generator_generator(func, max_times):
    """
    >>> numbers = generator_generator(lambda x: x + 1, 4)
    >>> list(numbers)
    [1, 2, 3, 4]
    >>> sum(generator_generator(lambda x: x + 1, 10))
    55
    """
```

Put the following lines of code in the correct order:

```
(a) while counter < max_times:
(b) yield func(counter)
(c) counter += 1
(d) counter = 0
```

Indentation does not matter.

Solution:

```
def generator_generator(func, max_times):
    """
    >>> numbers = generator_generator(lambda x: x + 1, 4)
    >>> list(numbers)
    [1, 2, 3, 4]
    >>> sum(generator_generator(lambda x: x + 1, 10))
    55
    """
    (d) counter = 0
    (a) while counter < max_times:
        (b) yield func(counter)
        (c) counter += 1
```

Q7.1
1 Point

Line 1:

A

B

C

D

Q7.2
1 Point

Line 2:

A

B

C

D

Q7.3
1 Point

Line 3:

A

B

C

D

Q7.4
1 Point

Line 4:

A

B

C

D

Q8 SQL
8 Points

Q8.1
2 Points

The programming language SQL best models which paradigm?

Imperative Programming

Array-Based Programming

Functional Programming

Object-Oriented Programming

Declarative Programing

Q8.2
2 Points

Put the SQL keywords in the right order they must be used in order for a query to work. Not all of these parts of a query are necessary, but if they are all there, there is only one correct order.

ORDER
FROM
GROUP BY
LIMIT
JOIN
WHERE
SELECT

SELECT FROM GROUP BY LIMIT JOIN WHERE ORDER

SELECT JOIN FROM WHERE GROUP BY LIMIT ORDER

SELECT FROM ORDER WHERE GROUP BY JOIN LIMIT

SELECT FROM JOIN WHERE GROUP BY ORDER LIMIT

SELECT JOIN FROM GROUP BY WHERE ORDER LIMIT

Q8.3
2 Points

For the following two questions consider these two tables:

drinks

id	name	syrup_id
1	pumpkin spice latte	1
2	strawberry latte	4
3	mocha	2
4	peppermint hot chocolate	3
5	vanilla latte	5
6	strawberry frappuccino	4

id	name	syrup_id

syrups

id	name	price
1	pumpkin spice	0.25
2	chocolate	0.50
3	pettermint	0.75
4	strawberry	1.00
5	vanilla	1.25

How many rows are in the cartesian product of the two tables drinks and syrups?

5

6

11

30

25

36

Q8.4
2 Points

How many rows would the returned table have if we JOIN the tables with the condition drinks.syrup_id = syrups.id?

5

6

11

30

25

36

Q9
8 Points

Please fill out the post-exam proctoring form.

https://go.c88c.org/proctoring-form

You'll get points for this question as long as you fill out the proctoring form.

Optionally, tell us how you're feeling!

Final Exam (Gradescope) ● Ungraded

134 Days, 11 Hours Late

Student
Anjali Gurajapu

Total Points
- / 92 pts

Question 1
Honor Code and Preliminaries 0 pts

Question 2 Conceptual Questions		20 pts
2.1	(no title)	2 pts
2.2	(no title)	2 pts
2.3	(no title)	2 pts
2.4	(no title)	2 pts
2.5	(no title)	2 pts
2.6	(no title)	2 pts
2.7	(no title)	2 pts
2.8	(no title)	2 pts
2.9	(no title)	2 pts
2.10	(no title)	2 pts
Question 3 Coffee, Coffee, ☕		19 pts
3.1	(no title)	3 pts
3.2	(no title)	2 pts
3.3	(no title)	2 pts
3.4	(no title)	2 pts
3.5	(no title)	2 pts
3.6	(no title)	2 pts
3.7	(no title)	2 pts
3.8	(no title)	2 pts
3.9	(no title)	2 pts
Question 4 (no title)		10 pts
4.1	(no title)	2 pts
4.2	(no title)	2 pts
4.3	(no title)	2 pts
4.4	(no title)	2 pts
4.5	(no title)	2 pts
Question 5 (no title)		17 pts
5.1	(no title)	3 pts
5.2	(no title)	2 pts
5.3	(no title)	2 pts
5.4	(no title)	3 pts

5.5	(no title)	2 pts
5.6	(no title)	3 pts
5.7	(no title)	2 pts

Question 6
Efficiency 6 pts

6.1	(no title)	2 pts
6.2	(no title)	2 pts
6.3	(no title)	2 pts

Question 7
Generating Generators Generates Generators 4 pts

7.1	(no title)	1 pt
7.2	(no title)	1 pt
7.3	(no title)	1 pt
7.4	(no title)	1 pt

Question 8
SQL 8 pts

8.1	(no title)	2 pts
8.2	(no title)	2 pts
8.3	(no title)	2 pts
8.4	(no title)	2 pts

Question 9
(no title) 8 pts