

Create Rubric

100 points

List View

Grid View

1 Create your rubric now or come back to it later. You can also make edits to your rubric while grading.

Q1 WWPD

10 points

Q1.1 a

2 points

Rubric

```
>>> tricky(f, 5)
```

1 +2.0

Correct: Function

2 +0.0

Incorrect

+ Add Rubric Item

Create Group

Import...

Q1.2 b

2 points

Rubric

```
>>> tricky(f, 5)(3)
```

1 +2.0

Correct: Error

2 +0.0

Incorrect

+ Add Rubric Item

Create Group

Import...

Q1.3 c

2 points

Rubric

```
>>> tricky(f, 5)(3, 20)
```

1 +2.0

Correct: 17

2 +0.0

Incorrect

+ Add Rubric Item

Create Group

Import...

Q1.4 d

1 point

Rubric

(d) (1.0 pt)

```
def outer_function():
    a = 5
    def inner_function():
        a = 10
        return a
    b = inner_function()
    print(f"Inner: {b}, Outer: {a}")

outer_function()
```

- Inner: 5, Outer: 10
- Inner: 10, Outer: 5
- Inner: 10, Outer: 10
- The code will result in an error.

1 +1.0

Correct: Inner: 10, Outer: 5

2 +0.0

Incorrect

+ Add Rubric Item

Create Group

Import...

Q1.5 e

1 point

Rubric

(e) (1.0 pt) Given a list of numbers, which code snippet uses filter to return a list with all neg removed? (Note: filter in Python returns an iterator that needs to be converted to a list)

```
numbers = [4, -1, -3, 2, 0, -5, 8]
 filter(lambda x: x < 0, numbers)
 filter(lambda x: x > 0, numbers)
 list(filter(lambda x: x < 0, numbers))
 list(filter(lambda x: x >= 0, numbers))
```

1 +1.0

Correct: list(filter(lambda x: x >= 0, numbers))

2 +0.0

Incorrect

+ Add Rubric Item

Create Group

Import...

Q1.6 f

1 point

Rubric

(f) (1.0 pt)

```
def square(number):
    try:
        return number ** 2
    except TypeError:
        return "hello"

print(square("two"))
 It will print "two"
 It will print 4
 It will print "hello"
 It will display a TypeError
```

1 +1.0

Correct: It will print "hello"

2 +0.0

Incorrect

+ Add Rubric Item

Create Group

Import...

Q1.7 g

1 point

Rubric

(g) (1.0 pt)

```
def check_even(number):
    return "Even" if number % 2 == 0 else "Odd"

numbers = [1, 2, 3, 4]
result = map(check_even, numbers)
print(list(result))
```

- [False, True, False, True]
- [Odd, Even, Odd, Even]
- [1, 2, 3, 4]
- [None, None, None, None]

1 +1.0

Correct: [Odd, Even, Odd, Even]

2 +0.0

Incorrect

Add Rubric Item

Create Group

Import...

Q2 Let's Explore

10 points

Q2.1 a

2 points

Rubric

(a) (2.0 pt) What is value of box (a)?

- func add_location(new_location) [parent=f1]
- func add_location(new_location) [parent=Global]
- func explore(places) [parent=f1]
- func explore(places) [parent=Global]
- ["moffitt", "mlk"]
- ["soda", "moffitt", "mlk"]

1 +2.0

func add_location(new_location) [parent=f1]

2 +0.0

Incorrect

Add Rubric Item

Create Group

Import...

Q2.2 b

2 points

Rubric

(b) (2.0 pt) What is the second element in the list loc, item (b)?

1 +2.0

["moffitt", "mlk"]

2 +1.0

answer includes moffitt and mlk but not in the correct format

3 +0.0

Incorrect

Add Rubric Item

Create Group

Import...

Q2.3 c

2 points

Rubric

(c) (2.0 pt) What is the parent of `lambda` function in frame 3, item (c)?

- Global
- f1
- f2
- f3

1 +2.0

f2

2 +0.0

Incorrect

Add Rubric Item

Create Group

Import...

Q2.4 d

2 points

Rubric

(d) (2.0 pt) What is the return value of the lambda function in f3, item (d)?

- An arrow pointing to the list `loc`
- `["soda", ["moffitt", "mlk"]]`
- `["soda", "moffitt", "mlk"]`
- `["soda"]`
- None

1 +2.0

None

2 +0.0

Incorrect

Add Rubric Item

Create Group

Import...

Q2.5 e

2 points

Rubric

(e) (2.0 pt) What is value of `result` and the Return Value of `add_location` in f2, item (e)?

- None
- Error
- An arrow pointing to the box (a)
- An arrow pointing to the list `loc`
- "soda"

1 +2.0

None

2 +0.0

Incorrect

Add Rubric Item

Create Group

Import...

Q3 A Broken Phone Book

4 points

Q3.1 a**2 points**

Rubric

- Select the option which describes the result of this code.
- The code errors and will return a key error
 - The code errors and will return a syntax error
 - The code is incorrect and will return a list of valid contacts
 - The code is correct and will return a list of compromised contacts

1 +2.0

Correct - will result in a key error

2 +0.0

incorrect

[+ Add Rubric Item](#)[Create Group](#)[Import...](#)**Q3.2 b****2 points**

Rubric

How will this code behave?

- The code block runs as expected, no changes needed.
- The code block errors and does not run.
- The code block runs but does not run as expected.

1 +2.0

Correct - code block runs but not as expected.

2 +0.0

Incorrect

[+ Add Rubric Item](#)[Create Group](#)[Import...](#)**Q4 Your Mileage May Vary****6 points**

Rubric

Q4.1 a**1 point**

```
def avg(lst):  
    return _____
```

1 +1.0

Correct: `sum(lst) / len(lst)`

2 +0.5

Correct, but with syntax error

3 +0.0

Incorrect

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q4.2 b

2 points

[Rubric](#)

```
mpgs = list(filter(_____  
_____  
_____))
```

1 +2.0

Correct: `mpgs = list(filter(lambda x: x >= 20, mpgs))`

2 +1.0

Minor Error

3 +0.0

Incorrect

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q4.3 c

2 points

[Rubric](#)

```
squared_difference = list(map(_____  
_____  
_____,))
```

1 +2.0

Correct:

`squared_difference = list(map(lambda x: (x-mean)**2,`

2 +1.0

Minor Error

3 +0.0

Incorrect

[+ Add Rubric Item](#)[Create Group](#)[Import...](#)**Q4.4 d****1 point**[Rubric](#)

(1.0 pt) Finally, find the mean (average) of the squared_difference list, assigning it to variance.

variance = _____

1 +1.0Correct: `avg(squared_difference)`**2 +1.0**

Correct: An equivalent answer

3 +0.5

Small Error

4 +0.0

Incorrect: Anything Else

[+ Add Rubric Item](#)[Create Group](#)[Import...](#)**Q5 CineDict****10 points**[Rubric](#)**Q5.1 a****2 points**

```
def create_movie(title, director, year):
    return _____
```

1 +2.0

Correct

`return {"title": title, "director": director, "releas`**2 -0.5**Syntax error `ie 'year' instead of 'release_year'`**3 -0.5**

Did not include quotes

4 -1.0

Missed one of the three options

5 +0.0

Incorrect/Blank

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q5.2 b

5 points

[Rubric](#)

```
def add_movie_to_catalog(movie_catalog, movie):
    director = movie[-----]
    if ----- not in -----:
        movie_catalog[-----] = [movie]
    else:
        movie_catalog[director].append(-----)
    return movie_catalog
```

1 **+5.0**

Fully Correct

2 **+1.0**

```
director = movie["director"]
```

3 **+1.0**

```
if director
```

4 **+1.0**

```
not in movie_catalog:
```

5 **+1.0**

```
movie_catalog[director] = [movie]
```

6 **+1.0**

```
movie_catalog[director].append(movie)
```

7 **-0.5**

Did not include quotes/incorrect use of quotes or appended a list

8 **+0.0**

Incorrect/Blank

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q5.3 c

3 points

[Rubric](#)

```
def get_movies_by_director(movie_catalog, director):
    return ----- .get(
        -----, -----)
```

1 **+3.0**

Fully Correct

2 +1.0

First blank is `movie_catalog`

3 +1.0

Second blank is `director`

4 +1.0

Third blank is `[]`

5 +0.0

Incorrect

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q6 Composing Trees with Trees

10 points

Q6.1 a

2 points

[Rubric](#)

(2.0 pt) Fill in blank (a).

1 +2.0

Correct: `tree.is_leaf()`

2 +1.0

Minor Error

3 +0.0

Incorrect

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q6.2 b

2 points

[Rubric](#)

(2.0 pt) Fill in blank (b).

1 +2.0

Correct: `tree.fn(tree.value)`

2 +1.0

Minor Error

3 +0.0

Incorrect

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q6.3 c

2 points

Rubric

(2.0 pt) Fill in blank (c).

1 +2.0

Correct: `tree.branches`

2 +1.0

Minor Error

3 +0.0

Incorrecdt

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q6.4 d

2 points

Rubric

(2.0 pt) Fill in blank (d).

1 +2.0

Correct: `branch.fn`

2 +1.0

Minor Error

3 +0.0

Incorrect

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q6.5 e

2 points

Rubric

(2.0 pt) Fill in blank (e).

1 +2.0

Correct: `f(g(x))`

2 +1.0

Minor Error

3 +0.0

Incorrect

+ Add Rubric Item

Create Group

Import...

Q7 Shopping List 2

8 points

Rubric

```
def lnk_aggregate(d):
    agg_lnk = None
    curr_lnk = None
    for item_name, total_price in d.items():
        node_lnk = -----
        if agg_lnk == None:
            agg_lnk = -----
            curr_lnk = -----
        else:
            -----
        return agg_lnk
```

1 +8.0

Fully correct

2 Creating `node_lnk`

3 Reassigning `agg_lnk`

4 Reassigning `curr_lnk`

5 `curr_lnk.rest = node_lnk`

6 `curr_lnk = curr_lnk.rest`

7 -0.5

Minor syntax errors, such as:

- Not capitalizing `Link`
- Not including the underscore in variable names
- Misspelling variable names

8 -1.0

Multiple minor syntax errors

9 +0.0

Incorrect/Blank

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q8 Mario Kart

14 points

Q8.1 a

2 points

[Rubric](#)

What should go in blank (a)?

- self.items.append(item_weight)
- self.items.append(self.weight)
- self.speed += item_weight
- self.acceleration += item_weight
- self.weight += item_weight

1 +0.0

self.items.append(item_weight)

2 +0.0

self.items.append(self.weight)

3 +0.0

self.speed += item_weight

4 +0.0

self.acceleration += item_weight

5 +2.0

self.weight += item_weight

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q8.2 b

2 points

[Rubric](#)

(b) (2.0 pt) What should go in blank (b)?

- self.items.append(item_name)
- self.items.append(item_weight)
- self.items += item_name
- self.weight += item_name

1 +2.0

self.items.append(item_name)

2 +0.0

self.items += [item_weight]

self.items.append(item_weight)

3 +0.0

self.items += item_name

4 +0.0

self.weight += item_name

5 +0.0

self.items += [item_weight]

6 +0.0

None

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q8.3 c

2 points

[Rubric](#)

What should fill in blank (c)?

1 +2.0

Any of the following:

- vehicles[i] == self
- vehicles[i] is self

2 +1.5

Minor syntax error, such as any of the following:

- Correct answer, but used parentheses instead of square brackets to index into vehicles
- vehicles[i] = self

3 +0.0

Incorrect

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q8.4 d

2 points

[Rubric](#)

) (2.0 pt) What should fill in blank (d)?

1

+2.0

Any of the following:

- vehicles[next_location]
- vehicles[curr_location + 1]

2

+1.5

Correct answer, but minor syntax error, such as any of the following:

- vehicles(next_location)

3

+0.0

Incorrect

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q8.5 e

2 points

[Rubric](#)

) (2.0 pt) What should fill in blank (e)?

1

+2.0

next_vehicle.speed -= 1 or equivalent

2

+1.0

Answer includes next_vehicle.speed - 1

3

+0.0

Incorrect

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q8.6 f

4 points

[Rubric](#)

```
^> b.items  
['blue shell']  
>> b.acceleration  
8.0  
'''
```

```
-----  
-----
```

▶ 1

Modifying the bike's acceleration

▶ 2

Picking up the item

[+ Add Rubric Item](#)[Create Group](#)[Import...](#)

Q9 Generate Factors

10 points

Q9.1 a

8 points

[Rubric](#)

```
'''  
i = _____  
-----:  
if i ..... k:  
    i = _____  
elif ..... % ..... == .....:  
    .....  
    i += _____
```

1 Partial credit

2 +8.0

Full credit

3 +0.0

Incorrect

4 -0.5

Syntax error (e.g. using parentheses with `yield`) or minor mistake, such as `i=0` instead of `i=1`, `i >= k` instead of `i > k`, or `i % k` instead of `k % i`

[+ Add Rubric Item](#)[Create Group](#)[Import...](#)

Q9.2 b

2 points

[Rubric](#)

) (2.0 pt) What would the output be if we called `list(generate_factors(10))`? Please note that we are `list` and not `next`.

- RecursionError: maximum recursion depth exceeded error.
- StopIteration error.
- [1, 2, 5, 10]
- [1, 2, 5, 10, 1, 2, ...]
- An infinite loop

1 +2.0

Correct: infinite loop

2 +0.0

Incorrect

[+ Add Rubric Item](#)[Create Group](#)[Import...](#)

Q10 May I take your order?

4 points

Q10.1 a

1 point

Rubric

(a) (1.0 pt) What is the order of growth of `sum_nums_A`?

- $O(1)$
- $O(\log(n))$
- $O(n)$
- $O(n^2)$
- $O(2^n)$

1 +1.0

Correct $O(n^2)$

2 +0.0

Incorrect $O(n)$

3 +0.0

Incorrect $O(2^n)$

4 +0.0

Incorrect $O(1)$

5 +0.0

Incorrect $O(\log n)$

+ Add Rubric Item

Create Group

Import...

Q10.2 b

1 point

Rubric

(b) (1.0 pt) `sum_nums_A` will perform faster than `sum_nums_B` for large inputs

- True
- False

1 +1.0

Correct False

2 +0.0

Incorrect/Both bubbled

+ Add Rubric Item

Create Group

Import...

Q10.3 c

1 point

Rubric

(c) (1.0 pt) `sum_nums_B` will perform faster than `sum_nums_C` for large inputs

- True
- False

1 +1.0

Correct True

2 +0.0

Incorrect/Bubbled both

+ Add Rubric Item

Create Group

Import...

Q10.4 d

1 point

Rubric

(d) (1.0 pt) sum_nums_C will perform faster than sum_nums_A for large inputs

- True
- False

1 +1.0

Correct:

2 +0.0

Incorrect/Bubbled both

+ Add Rubric Item

Create Group

Import...

Q11 NBA Networking

14 points

Rubric

Q11.1 a

4 points

```
SELECT _____  
FROM _____  
WHERE _____  
GROUP BY _____  
ORDER BY _____;
```

1 +0.5

Correct:

2 +0.5

Correct:

3 +1.0

Correct:

4 +1.0

Correct:

5 +1.0

Correct:

6 +0.0

Incorrect/Blank

7 -0.25

Minor syntax errors (extra or missing characters/words, incorrect aliasing, spelling mistakes)

+ Add Rubric Item

Create Group

Import...

Q11.2 b

3 points

Rubric

```
SELECT -----  
FROM -----  
-----;  
ORDER BY -----;
```

1 +0.5

Correct: `SELECT staff.name, staff.fav_player`

2 +0.5

Correct: `FROM staff`

3 +1.0

Correct:

`JOIN players ON staff.food_order = players.food_order`

4 +1.0

Correct: `ORDER BY staff.name ASC` OR
`ORDER BY staff.name`

5 +0.0

Incorrect/Blank

6 -0.25

Minor syntax errors (extra or missing characters/words, incorrect aliasing, spelling mistakes)

+ Add Rubric Item

Create Group

Import...

Q11.3 c

3 points

Rubric

```
SELECT -----  
FROM -----  
WHERE -----;  
-----;
```

1 +1.0

Correct: `SELECT name, food_order` OR
`SELECT s.name, s.food_order`

2

+0.5

Partial Credit: `SELECT name, food_order` or
`SELECT s.name, s.food_order` but includes additional unnecessary column(s)

3

+1.0

Correct: `FROM staff, players` or
`FROM staff AS s, players AS p`

4

+0.5

Partial Credit: `FROM staff` or `FROM players`

5

+1.0

Correct:

`WHERE staff.fav_player = players.name AND players.ju.`

6

+0.5

Partial Credit: `WHERE staff.fav_player = players.name`
OR `WHERE players.juice = 'apple'`

7

+0.0

Incorrect/Blank

8

-0.25

Minor syntax errors (extra or missing characters/words, incorrect aliasing, spelling mistakes, forgetting `AND`, extra `;`, using `==` instead of `=`, forgetting `''` around `apple`)

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q11.4 d

4 points

[Rubric](#)

```
SELECT _____  
FROM _____  
WHERE _____  
GROUP BY _____  
ORDER BY _____;
```

1

+1.0

Correct: `SELECT team, COUNT(*) AS total_staff`

2

+0.5

Partial Credit: `SELECT team` or
`SELECT COUNT(*) AS total_staff` and includes additional unnecessary column(s)

3 **+0.5**

Correct: `FROM staff, players` or with correct aliasing

4 **+0.25**

Partial Credit: `FROM staff` or `FROM players`

5 **+1.0**

Correct: `WHERE staff.fav_player = players.name`

6 **+0.5**

Partial Credit: Includes `fav_player` or `name`

7 **+1.0**

Correct: `GROUP BY team`

8 **+0.5**

Partial Credit: Includes `GROUP BY team` but with extra words/characters

9 **+0.5**

Correct: `ORDER BY total_staff DESC`

0 **+0.25**

Partial Credit: `ORDER BY total_staff` or `ORDER BY total_staff ASC` or includes `DESC`

+0.0

Incorrect/Blank

-0.25

Minor syntax errors (extra or missing characters/words, incorrect aliasing, spelling mistakes, extra `()` around `DESC`, extra `;`, `,` in `ORDER BY` statement, doesn't rename to `total_staff`)

-0.5

Major syntax errors

[+ Add Rubric Item](#)

[Create Group](#)

[Import...](#)

Q12 Bonus Questions

0 points

Q12.1 a

0 points

 Rubric 

(Any reasonable name counts for credit. :))

 1  +1.0

Answered something :)

Antony / Antonio / Anthony /Andy /Anty were the most common group, followed by names from the Ants project.

 2  +0.0

Blank

 Add Rubric Item

 Create Group

 Import...

Q12.2 b

0 points

 Rubric 

This question is extra credit. Do not attempt it until you are done! What is the order of growth of fib?

 1  +1.0

Correct - Linear $O(n)$

In this case, the first time a number like fib(3) needs to be calculated, it must make two recursive calls, but on the 2nd (or Nth) times a number needs to be calculated, the result is looked up from the dictionary of results, so the overall order of growth is linear

 2  +0.0

Incorrect - Constant $O(1)$

 3  +0.0

Incorrect - Quadratic / N^2

 4  +0.0

Incorrect - Logarithmic / $\log(n)$

 5  +0.0

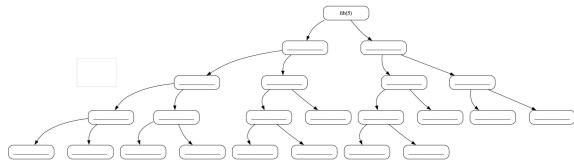
Incorrect - Exponential

 6  +0.0

[+ Add Rubric Item](#)[Create Group](#)[Import...](#)

Q12.3 c

0 points

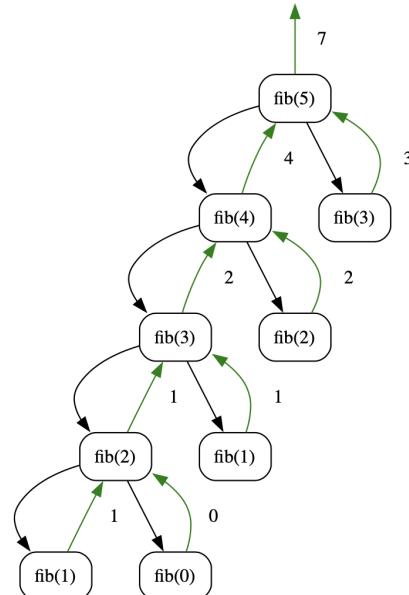
[Rubric](#)

1 +2.0

Correct

The function calls `fib(n-1)` first and calculates that result then stores it. The next time it tries to calculate a value, it can look up the result from the dictionary and doesn't make any additional function calls.

When `fib(3)` is called a second time (the right hand side) it makes no function calls, because the result is already in the dictionary.



(note this is just the output from recursionvisualiser.com - there were no expectations to write the return values.)

2 +1.0

Pretty close, but misses something like the exact order of calls.

(A full call tree for a "regular" `fib(5)` function does not count for this option, sorry!)

3 +0.0

Incorrect / Blank

[Add Rubric Item](#) [Create Group](#) [Import...](#)