Fall 2024
Course Name: Data 6
Instructor: Tsang
Quiz 1

Print Your Name: _____

Print Your Student ID: _____

You have 50 minutes. There are 5 questions of varying credit (100 points total).

Question:	1	2	3	4	5	Total
Points:	10	10	40	20	20	100

◯ For questions with **circular bubbles**, you may select only one choice.

☐ For questions with **square checkboxes**, you may select one or more choices.

Anything you write outside the answer boxes or anything you *cross out* will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation.

You are disallowed from using the following: for loops, while loops, if statements, and/or operators, ternary operators, lambdas.

You may only write one statement per line (i.e. you may not use semicolons).

Question 1: foo (10 Points)

Suppose we have the following function definition:

```python
def foo(x):
        return x + 3
```

What do the following expressions evaluate to? Assume that each expression is run independently from one another, not sequentially.

Question 1.1 (3 Points)

```python
foo(4 * 3)
```

Question 1.2 (3 Points)

```python
foo(4) * 3
```

Question 1.3 (4 Points)

```python
foo(foo(10) // foo(-1)) % 3
```

Question 2: Let's draw triangles! (10 points)

Kenneth loves triangles (they're his favorite shape). He writes some code to draw triangles and wants to track how many times he ends up drawing one. He ideates three different workflows to track the number of times `draw_triangle` is called:

Workflow 1:
Run `CELL A` once, then run `CELL B` every time `draw_triangle` is called.

```
#CELL A
num_tries = 0
```

```
#CELL B
def draw_triangle():
    print("  *  ")
    print(" *** ")
    print("*****")

num_tries = num_tries + 1
```

Workflow 2:
Calling `draw_triangle` is sufficient to increase `num_tries` every time it is called.

```
def draw_triangle():
    num_tries = 0
    print("  *  ")
    print(" *** ")
    print("*****")
    return num_tries + 1
```

Workflow 3:
Run `CELL A` once, then re-assign the return value to `num_tries` every time `draw_triangle` is called.

```
#CELL A
num_tries = 0
def draw_triangle():
    print("  *  ")
    print(" *** ")
    print("*****")
    return num_tries + 1
```

```
#Do this every time draw_triangle is called
num_tries = draw_triangle()
num_tries
```

Given the following three workflows, which of them will correctly update the variable `num_tries` to reflect the number of times the function is called? Select all that apply.

- ☐ Workflow 1
- ☐ Workflow 2
- ☐ Workflow 3

Question 3: Birthdays! (40 points)

Suppose we have a Table `birthdays`, which contains a person's name and birthday (the birthday is stored in YYYY-MM-DD format). All values are stored as Strings. You can disregard the `Magic Year` for now.

Name	Birthday	Magic Year
Arthur	1987-04-01	1980
Beth	2003-05-23	2005
Chand	2008-01-01	2068

In the example table above, Arthur's birthday is on April 1st, 1987.

Problem continued on the next page…

Part A: `age_in_given_year` (10 points)

Implement a function `age_in_given_year`, which takes in a `year` and a `birthday`, and returns the age that person will turn on that `year`. You may return any negative number if the `birthday` is after the `year`.

For example,

`age_in_given_year("2010", "2000-01-01")` is equal to 10

`age_in_given_year("2010", "2010-01-01")` is equal to 0

`age_in_given_year("2010", "2012-01-01")` is equal to any negative number.

You may assume that all arguments are well-formatted.

Python

```
def age_in_given_year(year, birthday):
        """
        Inputs:
        year: a String representing the year we care about
        birthday: a String in YYYY-MM-DD format

        Returns:
        an integer representing the age a person with the given birthday will be
        in the given year. Returns any negative number if the birthday is after
        the year.
        """

        str_year, str_month, str_day = ________________________________________


        return ____________________________________________________________
```

Part B: `data_magic` (20 points)

Assume `age_in_given_year` has been implemented correctly. Let's create a new Table called `data_magic` which contains the same information as `birthdays`, but contains an extra column called "Age in Magic Year" signifying what age everyone in `birthdays` would be in their magic year. For example, Chand's magic year is 2068, and since their birthday is in 2008, their age in the magic year should be 60.

However, not everyone in our birthday table was born before their magic year! If anyone in our table was born after their magic year, they should not be a part of our resulting table.

For example, the table below

Name	Birthday	Magic Year
Arthur	1987-04-01	1980
Beth	2003-05-23	2003
Chand	2008-01-01	2068

turns into this table:

Name	Birthday	Magic Year	Age in Magic Year
Beth	2003-05-23	2003	0
Chand	2008-01-01	2068	60

Python

```
data_magic = birthdays._____(blank a)_____(
    "Age in Magic Year",
    birthdays._____(blank b)_____(
        _____(blank c)_____,
        _____(blank d)_____,
        _____(blank e)_____
    )
).where(
    "Age in Magic Year",
    _____(blank f)_____
)
```

1. What goes in _____(blank a)_____? (3 points)
   - ◯ column
   - ◯ relabeled
   - ◯ sort
   - ◯ apply
   - ◯ join
   - ◯ with_column
   - ◯ with_row

2. What goes in _____(blank b)_____? (3 points)
   - ◯ column
   - ◯ relabeled
   - ◯ sort
   - ◯ apply
   - ◯ join
   - ◯ with_column
   - ◯ with_row

3. What goes in _____(blank c)_____? (3 points)
   - ◯ birthdays
   - ◯ data_magic
   - ◯ age_in_given_year
   - ◯ “Name”
   - ◯ “Birthday”
   - ◯ “Magic Year”

4. What goes in _____(blank d)_____? (3 points)
   - ◯ birthdays
   - ◯ data_magic
   - ◯ age_in_given_year
   - ◯ “Name”
   - ◯ “Birthday”
   - ◯ “Magic Year”

5. What goes in _____(blank e)_____? (3 points)
   - ◯ birthdays
   - ◯ data_magic
   - ◯ age_in_given_year
   - ◯ "Name"
   - ◯ "Birthday"
   - ◯ "Magic Year"

6. What goes in blank f? (5 points)

Part C: Methods (10 points)

Now, assume that `data_magic` has been implemented correctly. Su Min gives us another table, `data_methods`, which notes how each person picked their Magic Year. The options are "Random", "Lucky Number", or "Other".

Below is what `data_methods` looks like:

Name	Method of Selection
Beth	Random
Chand	Random
Arthur	Lucky Number

As a reminder, this is what `data_magic` looks like:

Name	Birthday	Magic Year	Age in Magic Year
Beth	2003-05-23	2003	0
Chand	2008-01-01	2068	60

1. What type of variable is the Method of Selection? (5 points)
   - ◯ Categorical
   - ◯ Numerical
2. Select which call expression would result in the following table: (5 points)

Name	Method of Selection	Birthday	Magic Year	Age in Magic Year
Beth	Random	2003-05-23	2003	0
Chand	Random	2008-01-01	2068	60

   - ◯ `data_methods.join(data_magic)`
   - ◯ `data_magic.join(data_methods)`
   - ◯ `data_methods.join("Name", data_magic)`
   - ◯ `data_magic.join("Name", data_methods)`

Question 4: You knew this was coming (20 points)

Given the following Python code, answer the following questions:

Recommendation: Read through the questions first, then analyze the code.

```
1  dont = "trust"
2
3  def trust(what):
4      dont = "dont" + " " + what
5      return print(dont)
6
7  dont = trust(dont + " " + dont)
8
9  trust("you see!")
10
```

1. Besides the global frame, how many new local frames are opened? (5 points)
   Note: Do not count any local frames opened from calling `print`.
   - ○ 0
   - ○ 1
   - ○ 2
   - ○ 3
   - ○ 4

2. Fill in the local frame that's created in line 7.

trust	
what	BLANK ONE
dont	BLANK TWO
Return value	BLANK THREE

2a. BLANK ONE should be… (5 points)

- ◯ "dont dont"
- ◯ "trust trust"
- ◯ "dont trust"
- ◯ "dont what"
- ◯ None

2b. BLANK TWO should be…(5 points)

- ◯ "dont dont dont"
- ◯ "dont trust trust"
- ◯ "dont dont trust"
- ◯ "dont dont what"
- ◯ "trust dont dont"
- ◯ "trust trust trust"
- ◯ "trust dont trust"
- ◯ "trust dont what"
- ◯ None

2c. BLANK THREE should be… (5 points)

- ◯ "dont dont dont"
- ◯ "dont trust trust"
- ◯ "dont dont trust"
- ◯ "dont dont what"
- ◯ "trust dont dont"
- ◯ "trust trust trust"
- ◯ "trust dont trust"
- ◯ "trust dont what"
- ◯ print(dont)
- ◯ None

There is a question behind this page as well.

Question 5: Why is it SO HOT (20 points)

Jedi observes that less people were out and about on October 2nd, 2024. He theorizes that it must be the heat that led people to stay indoors. It was also quite humid on that day…

Write the relevant social sciences term next to each term/statement. You may only use each term once. Not all terms will be used. (2 points each)

Options: Confounding variable, theory, concept, causal hypothesis, associative hypothesis, exploratory research question, unit of analysis, scientific method, aggregation, disaggregation, internal validity, external validity, generalizability, categorical variable, numerical variable

1. Separating out the data we collected into different age groups to see if different groups reacted differently to the heat.
2. The number of people staying indoors.
3. How does temperature affect the behavior of people in urban settings?
4. Jedi asks his friends what temperature it feels like in order to measure the temperature. This is a threat to…
5. "Heat can impact human social behavior and social dynamics."
6. Social behavior and social dynamics
7. Jedi worries that this study only applies to students at UC Berkeley, but not students at Stanford. This is a threat to…
8. Humidity
9. "To avoid discomfort, more people stay indoors on hotter days."
10. "More people stay indoors on hotter days."