APS106

UNIVERSITY OF TORONTO

Exam Jam: Course Review

We'll be starting at the 10 minute mark

Nested Containers and Indexing

```
smartphones = {
    'brands': ['Apple', 'Samsung', 'Google', 'OnePlus'],
    'specs': {
        'Apple': ('iPhone 12', 2020),
        'Samsung': ('Galaxy S20', 2020),
        'Google': ('Pixel 5', 2020),
        'OnePlus': ('8T', 2020)},
    'features': [
        ('Apple', 'iOS'),
        ('Samsung', 'Android'),
        ('Google', 'Stock Android'),
        ('OnePlus', 'Fast Charging')] }

print(smartphones['features'][-4:-2][0])
```

What is the output?

A. ('Apple','iOS')
B. ('Samsung', 'Android')
C. ['Apple', 'Samsung']
D. Error
E. None of the above

Nested Containers and Indexing

```
smartphones = {
    'brands': ['Apple', 'Samsung', 'Google', 'OnePlus'],
    'specs': {
        'Apple': ('iPhone 12', 2020),
        'Samsung': ('Galaxy S20', 2020),
        'Google': ('Pixel 5', 2020),
        'OnePlus': ('8T', 2020)},
    'features': [
        ('Apple', 'iOS'),
        ('Samsung', 'Android'),
        ('Google', 'Stock Android'),
        ('OnePlus', 'Fast Charging')] }

print(smartphones['features'][-4:-2][0])
```

What is the output?

[✓ A. ('Apple','iOS')]
B. ('Samsung', 'Android')
C. ['Apple', 'Samsung']
D. Error
E. None of the above

Containers: Sets

What is the output from this code?

```
values1 = set(list('heyheyhey'))
values2 = {'h','e','l','l','o','t','h','e','r','e'}
values3 = values1.intersection(values2)

print(len(values3))
print(values3[0])
```

Containers: Sets

What is the output from this code?

```
values1 = set(list('heyheyhey'))
values2 = {'h','e','l','l','o','t','h','e','r','e'}
values3 = values1.intersection(values2)

print(len(values3))
print(values3[0])
```

Answer:

```
>>2
>>Error - sets not
subscriptable
```

Dictionaries, Aliasing and Functions

What would be the output?

```
def my_func(param = {'key1': 0, 'key2': 0}):
    for key in param:
        param[key] += 40

    param['key4'] = 40
    param = dict(param)
    param['key5'] = 70

arg = {'key1':10, 'key2':20, 'key3': 30}
my_func(arg)

print(arg)
```

Dictionaries, Aliasing and Functions

What would be the output?

```
def my_func(param = {'key1': 0, 'key2': 0}):
    for key in param:
        param[key] += 40

    param['key4'] = 40
    param = dict(param)
    param['key5'] = 70

arg = {'key1':10, 'key2':20, 'key3': 30}
my_func(arg)

print(arg)
```

Solution:

```
{'key1': 50, 'key2': 60, 'key3': 70,
'key4': 40}
```

For loops and range()

What would be the output from this code?

```
my_result = ""
for i in range(221, 209, -4):
    my_result += str(i)[-2]
print(my_result)
```

What is the output?

A. 221
B. 211
C. 221 217 213
D. 111
E. None of the above

For loops and range()

What would be the output from this code?

```
my_result = ""
for i in range(221, 209, -4):
    my_result += str(i)[-2]
print(my_result)
```

What is the output?

A. 221
[✓ B. 211]
C. 221 217 213
D. 111
E. None of the above

Classes, Objects and Attributes

```
class Digit:
    def __init__(self, digit):
        self.digit = digit

class Number:
    def __init__(self, digit):
        self.number = int(digit.digit)

    def printer(self):
        return self.number

# Creating instances of Digit and Number
digit1 = Digit("5")
Number1 = Number(digit1)
Number2 = Number(Digit("3"))

print(Number1.printer() > Number2.printer())
```

What is the output?

a) True
b) False
c) Error
d) None of the above

Classes, Objects and Attributes

```
class Digit:
    def __init__(self, digit):
        self.digit = digit

class Number:
    def __init__(self, digit):
        self.number = int(digit.digit)

    def printer(self):
        return self.number

# Creating instances of Digit and Number
digit1 = Digit("5")
Number1 = Number(digit1)
Number2 = Number(Digit("3"))

print(Number1.printer() > Number2.printer())
```

What is the output?

[✓ a) True]
b) False
c) Error
d) None of the above

Pandas, DataFrames and iloc/loc

What is the output from this code?

```
import pandas as pd

data = [
    ["Alice", "APS106", 92, "MY150"],
    ["Bob", "CIV185", 95, "BA1150"],
    ["Charlie", "APS112", 73, "MY150"],
    ["Diana", "MAT187", 88, "MC252"]
]
columns = ["Name", "Subject", "Grade", "Location"]
students_df = pd.DataFrame(data, columns=columns)
students_df.sort_values(by = "Name")

students_df.loc[0:2, "Name":"Grade"].iloc[-2]
```

Pandas, DataFrames and iloc/loc

What is the output from this code?

```
import pandas as pd

data = [
    ["Alice", "APS106", 92, "MY150"],
    ["Bob", "CIV185", 95, "BA1150"],
    ["Charlie", "APS112", 73, "MY150"],
    ["Diana", "MAT187", 88, "MC252"]
]
columns = ["Name", "Subject", "Grade", "Location"]
students_df = pd.DataFrame(data, columns=columns)
students_df.sort_values(by = "Name")

students_df.loc[0:2, "Name":"Grade"].iloc[-2]
```

Solution:

```
Name            Bob
Subject      CIV185
Grade            95
Name: 1, dtype: object
```

String Methods

What is the output from this code?

```
input_string = "  Hello World.   "
input_string = input_string.strip().replace(" ", "_").upper()
input_string = input_string.replace('World','APS106')
input_string.lower()
print(input_string)
```

UNIVERSITY OF TORONTO

String Methods

What is the output from this code?

```
input_string = "  Hello World.  "
input_string = input_string.strip().replace(" ", "_").upper()
input_string = input_string.replace('World','APS106')
print(input_string)
```

Solution: **`HELLO_WORLD.`**

While and For Loops, range() and conditions

```
i = 0
results = []
while i < 2:
    for num in range(3, 5):
        if i % 2 == 0:
            results.append((i, num, "Even"))
        else:
            results.append((i, num, "Odd"))
    i += 1

print(results)
```

While and For Loops, range() and conditions

```
i = 0
results = []
while i < 2:
    for num in range(3, 5):
        if i % 2 == 0:
            results.append((i, num, "Even"))
        else:
            results.append((i, num, "Odd"))
    i += 1

print(results)
```

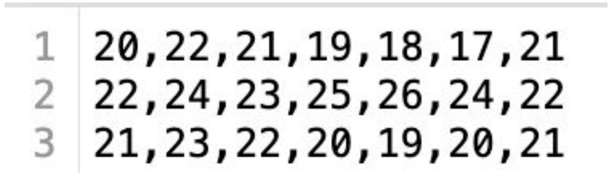
Solution:

```
[(0, 3, 'Even'), (0, 4, 'Even'),
(1, 3, 'Odd'), (1, 4, 'Odd')]
```

Text Files – Reading and Writing

Write a program that iterates through the comma-separated file `weekly temperatures.txt`, and writes each week's temperature (represented by an individual row) to a nested list `weekly_temperatures`. The inputs and outputs look like so:

weekly_temperatures.txt



```
20,22,21,19,18,17,21
22,24,23,25,26,24,22
21,23,22,20,19,20,21
```

```
weekly_temperatures = [
    [20, 22, 21, 19, 18, 17, 21],
    [22, 24, 23, 25, 26, 24, 22],
    [21, 23, 22, 20, 19, 20, 21]]
```

Write the Code: User-Defined Classes

Write a `PetDatabase` class that manages a collection of pets as a Python dictionary with the **owner_name** as the key and their **pet_name** as the value. Note that an individual owner may have more than one pet (hint: use a list for the values). The class should allow owners and pets to be **added to** and **removed from** the database, and it should provide a way to **search for all pets** from a particular owner. If you try **print your database object** instance, it should return the dictionary with the owners and pets.

Your class should work with the following code:

```
pet_db = PetDatabase()

pet_db.add_pet("Tisha", "Katia")
pet_db.add_pet("Catonio Banderas", "Ben")
pet_db.add_pet("Nugget", "Katia")
pet_db.add_pet("Moody", "Katia")

pet_db.remove_pet("Moody", "Katia") # remove Moody
print(pet_db.find_pets_by_owner("Ben")) # Expected output: ['Catonio Banderas']
print(pet_db) # Expected output: {"Ben":['Catonio Banderas'], "Katia": ["Tisha","Nugget"]}
```

From 2019 exam

Write the Code: Functions, String Indexing

Write a function `scramble_items` that takes as input a list of strings, ints, floats or a combination of the three types and returns a list of strings with their characters scrambled. The scrambling process will be performed on each item in the list based on indices: characters with even indices are all placed after the characters with odd indices. For example, if the string is `"Engineers!"` then the scrambled string will he `"nier! Egnes"`

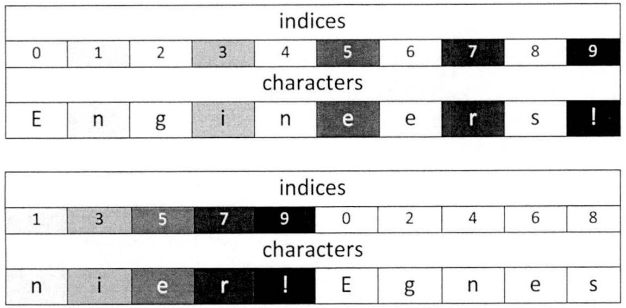
```
def scramble_items(sample_list):
    """[item, item,... ,item] -> [str, str,... ,str]
    Input: list of items that could be int, float, or string.
    Output: list of scrambled strings. """
```

Write the Code: Functions, String Indexing cont.

Sample function use:

```
sample_list = ['Elon Tusk', 420, 'Engineers']
new_list = scramble_list(sample_list)
print(new_list)

>> ['lnTsEo uk','240','nier!Egnes']
```



indices									
0	1	2	3	4	5	6	7	8	9
characters									
E	n	g	i	n	e	e	r	s	!

indices									
1	3	5	7	9	0	2	4	6	8
characters									
n	i	e	r	!	E	g	n	e	s

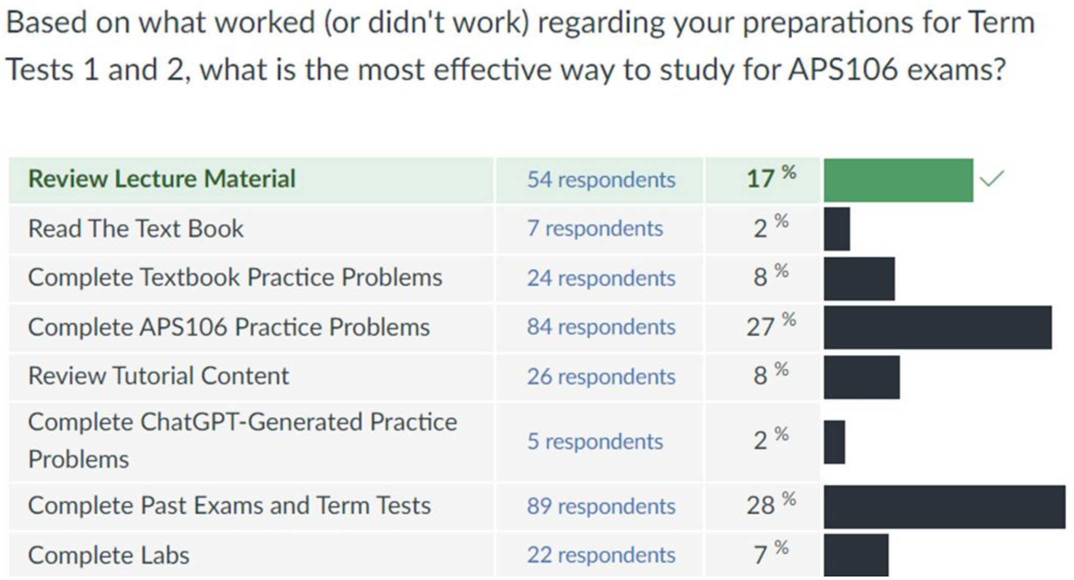
From 2019 exam

Complete the code:

```
def scramble_items(sample_list):
    """[item, item,... ,item] -> [str, str,... ,str]
    Input: list of items that could be int, float, or string.
    Output: list of scrambled strings. """
```

Study Tips

- Practice, practice, practice
- You told us:



Based on what worked (or didn't work) regarding your preparations for Term Tests 1 and 2, what is the most effective way to study for APS106 exams?

Review Lecture Material	54 respondents	17 %	✓
Read The Text Book	7 respondents	2 %	
Complete Textbook Practice Problems	24 respondents	8 %	
Complete APS106 Practice Problems	84 respondents	27 %	
Review Tutorial Content	26 respondents	8 %	
Complete ChatGPT-Generated Practice Problems	5 respondents	2 %	
Complete Past Exams and Term Tests	89 respondents	28 %	
Complete Labs	22 respondents	7 %	

Best of luck during this exam season!