Worksheet 23: Python Set Union

Name: _____ Date: _____

Instructions

- Answer in the blanks.
- For “write code” questions, write valid Python code (no functions needed).
- For “what does it print” questions, write the **exact** output (line by line).
- If a question says “order may vary”, any correct order is acceptable.

Part A — Union basics

1) Meaning + syntax (fill in the blanks)

- Union means items that are in _____ set. (Choose: `either`, `both`)
- Syntax: `either = set1 ____ set2` (Choose: `|`, `&`)

2) Simple union (what does it print?)

```
a = (1, 2, 3)
b = (3, 4)

print(a | b)
```

Output (order may vary):

```

```

3) Union vs intersection (fill in the blanks)

Fill in the blanks.

- `A | B` gives items in _____ `A` _____ `B`. (Choose: `either`, `both`, `or`, `and`)
- `A & B` gives items in _____ `A` _____ `B`. (Choose: `either`, `both`, `or`, `and`)

---

4) Check membership after union (what does it print?)

```
math_club = ('Amy', 'Ben', 'Chloe', 'Dylan')
swim_club = ('Ben', 'Dylan', 'Eva')

either = math_club | swim_club

print("Eva" in either)
print("Zoe" in either)
```

Output:

```

```

Part B — Union for “unique items”

5) Unique items from two lists (fill in the blanks)

Complete the code so `unique` is a set of all unique items from both lists.

Choose from: `set`, `list`, `|`, `&`.

```
list1 = ["chips", "apple", "apple", "cookie"]
list2 = ["cookie", "banana"]

unique = ______(list1) ____ ______(list2)

print(unique)
```

Output (order may vary):

Output (order may vary):

```
```

6) How many unique items? (what does it print?)

```
list1 = ["chips", "apple", "apple", "cookie"]
list2 = ["cookie", "banana"]

unique = set(list1) | set(list2)
print(len(unique))
```

Output:

```
```

7) Add first, then union (what does it print?)

(order may vary)

```
s = ('a', 'b')
t = ('b', 'c')

s.add("d")
print(s | t)
```

Output:

```
```

8) Unique items from many lists (write code)

We have many shopping lists.

Write code to put **all unique items** into `result` using a loop and **union**.

```
lists = [
    ["milk", "egg", "egg"],
    ["egg", "bread"],
    ["apple"],
]

result = ____________________

for l in lists:
    s = set(l)
    result = _________________

print(result)
```

Expected output (order may vary):

```
{'milk', 'egg', 'bread', 'apple'}
```

Part C — Union of many sets

9) Fill in the blanks

Find the union of all sets:

```
sets = [
    {'A', 'B'},
    {'B', 'C'},
    {'A', 'C'},
    {'D'}
]

result = ______

for s in sets:
    result = result ____ s  # choose from: &, |, +, -

print(result)
```

Output (order may vary):

```
```

10) Trace the loop (fill in the blanks)

```
sets = [
    {1, 2},
    {2, 3},
    {10},
]

result = set()
for s in sets:
    result = result | s
```

After the loop:

- After the **1st** set, `result` is _____
- After the **2nd** set, `result` is _____
- After the **3rd** set, `result` is _____

11) Fix the bug (union vs intersection)

This code is supposed to find the union, but it uses the wrong operator.

Fix **one line**.

```
sets = [
    {'red', 'blue'},
    {'blue', 'green'},
    {'yellow'},
]

result = set()
for s in sets:
    result = result & s   # BUG!

print(result)
```

Expected output (order may vary):

```
{'red', 'blue', 'green', 'yellow'}
```

Part D — Mini challenges

12) Weekend activities (what does it print?)

```
sat = ('tennis', 'lego', 'movie')
sun = ('movie', 'piano', 'lego')

either_day = sat | sun
print(either_day)
```

Output (order may vary):

```

```

13) Union of letters

What is the output?

```
word1 = "cat"
word2 = "hat"

s1 = set(word1)  # s1 = {'c', 'a', 't'}
s2 = set(word2)  # s2 = {'h', 'a', 't'}

print(s1 | s2)
```

Output (order may vary):

```

```

14) Numbers divisible by either 2 or 3 (set union)

Write code to build:

- `evens` : numbers from `1` to `20` that are divisible by `2`
- `threes` : numbers from `1` to `20` that are divisible by `3`
- `either` : numbers divisible by `2` **or** `3` (use union)

```
evens = set()
threes = set()

for i in range(1, 21):
    if _______________:
        evens.add(i)
    if _______________:
        threes.add(i)

either = ________________________  # set union

print(either)
```

Expected output (order may vary):

```
{2, 3, 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20}
```