Solution 23: Python Set Union

1) Meaning + syntax (fill in the blanks)

- Union means items that are in **either** set.
- Syntax: `either = set1 | set2`

2) Simple union (what does it print?)

One possible output (order may vary):

```
{1, 2, 3, 4}
```

3) Union vs intersection (fill in the blanks)

- `A | B` gives items in **either** `A` **or** `B`.
- `A & B` gives items in **both** `A` **and** `B`.

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

```
True
False
```

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

One correct answer (order may vary):

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

unique = set(list1) | set(list2)

print(unique)
```

One possible output (order may vary):

```
{'chips', 'apple', 'cookie', 'banana'}
```

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

```
4
```

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

One possible output (order may vary):

```
{'a', 'b', 'c', 'd'}
```

8) Unique items from many lists (write code)

One correct answer:

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

result = set()

for l in lists:
    result = result | set(l)

print(result)
```

Expected output (order may vary):

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

9) Fill in the blanks

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

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

print(result)
```

One possible output (order may vary):

```
{'A', 'B', 'C', 'D'}
```

10) Trace the loop (fill in the blanks)

(order may vary)

- After the **1st** set, `result` is `{1, 2}`
- After the **2nd** set, `result` is `{1, 2, 3}`
- After the **3rd** set, `result` is `{1, 2, 3, 10}`

---

11) Fix the bug (union vs intersection)

Fix the operator:

```
result = result | s
```

Expected output (order may vary):

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

---

12) Weekend activities (what does it print?)

One possible output (order may vary):

```
{'tennis', 'lego', 'movie', 'piano'}
```

---

13) Union of letters (what does it print?)

One possible output (order may vary):

```
{'c', 'a', 't', 'h'}
```

---

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

One correct answer:

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

for i in range(1, 21):
    if i % 2 == 0:
        evens.add(i)
    if i % 3 == 0:
        threes.add(i)

either = evens | threes

print(either)
```

Expected output (order may vary):

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