Solution 20: Python Set Basic Operations

1) Create a set with braces (write code)

One correct answer:

```
names = {"Alice", "Bob", "Chelsea"}

print(names)
print(type(names))
```

(Printing a set may show items in a different order.)

2) Empty set vs empty dictionary (what does it print?)

```
<class 'dict'>
<class 'set'>
```

3) Set from list (duplicates disappear)

```
6
3
```

4) Quick check (fill in the blanks)

- `{}` makes an empty **dictionary**.
- `set()` makes an empty **set**.

5) `in` and `not in` (what does it print?)

```
True
False
True
```

---

6) List vs set membership (fill in the blanks)

- `"x" in l` has **O(n)** average time complexity.
- `"x" in s` has **O(1)** average time complexity.

---

7) Membership check using a set (write code)

One correct answer:

```
items = ["pen", "pencil", "eraser", "pen"]
target = "eraser"

s = set(items)
print(target in s)
```

---

8) Fix the bug (syntax)

Corrected code:

```
s = {1, 2, 3}
print(2 not in s)   # should print False
```

---

9) Add and discard (what does it print?)

Step by step:

- Start: `{10, 20}`
- Add 30 → `{10, 20, 30}`
- Discard 40 → no change
- Discard 10 → `{20, 30}`

So:

```
2
False
True
```

---

10) `discard` facts (True/False)

1. **True**
2. **False**
3. **True**

---

11) Build and change a set (write code)

One correct answer:

```
s = set()

for x in range(1, 6):
    s.add(x)

s.discard(3)

print(len(s))
print(3 in s)
print(5 in s)
```

Output would be:

- `len(s)` is 4
- `3 in s` is False
- `5 in s` is True

---

12) Remove duplicates (what does it print?)

```
5
3
True
```

---

13) Convert set back to list (write code)

One correct answer:

```
s = {"red", "blue", "green"}

colors = list(s)
print(type(colors))
```

It prints:

```
<class 'list'>
```

14) Find common items (write code)

One correct answer:

```
a = ["apple", "banana", "pear", "apple"]
b = ["pear", "kiwi", "banana", "banana"]

s = set(a)
common = set()

for x in b:
    if x in s:
        common.add(x)

print(len(common))
```

`common` contains `{"pear", "banana"}`, so it prints:

```
2
```

15) Does the list have duplicates? (fill in the blanks)

A common way:

```
l = [5, 1, 5, 2]

s = set(l)

if len(s) < len(l):
    print("duplicates")
else:
    print("no duplicates")
```