Worksheet 20: Python Set Basic Operations

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).

Part A — Create sets

1) Create a set with braces (write code)

Write Python code to create a set named `names` that contains these strings:

- `"Alice"`
- `"Bob"`
- `"Chelsea"`

Then print `names` and `type(names)`.

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

```
a = {}
b = set()

print(type(a))
print(type(b))
```

Output:

---

3) Set from list (duplicates disappear)

What does it print?

```
l = [1, 2, 2, 3, 3, 3]
s = set(l)

print(len(l))
print(len(s))
```

Output:

---

4) Quick check (fill in the blanks)

Fill in the blanks by choosing from `list`, `set`, `dict`.

- `{}` makes an empty ______________________.
- `set()` makes an empty ______________________.

Part B — Membership check (`in` / `not in`)

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

```
s = {"A", "B", "C"}

print("B" in s)
print("D" in s)
print("D" not in s)
```

Output:

```

```

---

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

`l` is a list, and `s` is a set.

Fill in the blanks (use `1` or `n`):

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

---

7) Membership check using a set (write code)

Write code that:

1. converts `items` to a set named `s`
2. prints whether `target` is in `s` (`True` or `False`)

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

# complete the code:

s = ___________________

print(_________________)
```

---

8) Fix the bug (syntax)

This code has a syntax bug. Fix it so it runs.

```
s = {1, 2, 3}
print(2 no in s)   # should print False
```

Write the corrected code:

Part C — Add and discard

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

```
s = {10, 20}

s.add(30)
s.discard(40)   # 40 is not in the set
s.discard(10)

print(len(s))
print(10 in s)
print(30 in s)
```

Output:

10) `discard` facts (True/False)

Circle **True** or **False**.

1. `s.discard(x)` removes `x` if `x` is in the set. (True / False)
2. `s.discard(x)` crashes if `x` is not in the set. (True / False)
3. After `s.add(x)`, `x in s` is always True. (True / False)

11) Build and change a set (write code)

Write code that does the following:

- Start with an empty set `s`.
- Add numbers `1`, `2`, `3`, `4`, `5` into `s` (use a loop).
- Discard `3`.
- Print:
    - `len(s)`
    - `3 in s`
    - `5 in s`

```
s = __________________  # create an empty set

for x in range(_______, _______):
    s.________________  # add or append?

s.____________________  # discard 3

print(________________)
print(________________)
print(________________)
```

Part D — Convert between list and set

12) Remove duplicates (what does it print?)

```
words = ["hi", "bye", "hi", "yes", "bye"]
unique = set(words)

print(len(words))
print(len(unique))
print("yes" in unique)
```

Output:

13) Convert set back to list (write code)

You are given:

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

Write code that converts `s` to a list named `colors`.

Then print `type(colors)`.

```
colors = _______________________
print(__________________________)
```

Part E — Mini challenges (sets + loops)

14) Find common items (write code)

You are given two lists: `a` and `b`. Write code to build a set named `common` that contains items that appear in **both** lists.

Hint:

- Convert one list to a set for fast membership check.
- Use a loop and `add`.

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

s = ________________________  # convert list a to set s
common = ___________________  # create an empty set

for x in b:
    if x _________ s:         # in or not in?
        common.__________(x)  # add or append?

# Now, common should contain "pear" and "banana"

print(len(common))
```

Output:

```

```

15) Does the list have duplicates?

You are given a list `l`.

Complete the condition so it prints `"duplicates"` when the list has duplicates:

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

s = ___________________   # convert l to a set

if len(s) ______ len(l):  # compare the lengths
    print("duplicates")
else:
    print("no duplicates")
```