Worksheet 22: Python Set Intersection

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 — Intersection basics

1) Meaning + syntax (fill in the blanks)

Fill in the blanks.

- Intersection means items that are in _____ sets. (both, either, or neither?)
- Syntax: `common = set1 _____ set2`. (Choose from: `+`, `-`, `*`, `/`, `&`, `|`)

2) Simple intersection (what does it print?)

```
a = {1, 2, 3, 4}
b = {3, 4, 5}

print(a & b)
```

Output (order may vary):

```

```

3) Empty intersection (what does it print?)

```
x = {"A", "B"}
y = {"C", "D"}

print(x & y)
print(len(x & y))
```

Output:

```

```

4) Duplicates are removed (what does it print?)

(order may vary)

```
s1 = {2, 2, 2, 3, 4}
s2 = {1, 2, 2, 5}

common = s1 & s2
print(s1)
print(s2)
print(common)
print(len(common))
```

Output:

```

```

Part B — Overlap and conflict

5) Piano schedule conflict (fill in the blanks)

Complete the code so it prints `True` if there is a conflict, otherwise `False`.

```
ada = {"Mon", "Wed"}
ben = {"Tue", "Thu"}

common = ada ______ ben              # items in both sets
is_conflict = len(common) ______ 0

print(is_conflict)
```

Output:

```

```

6) Overlapping dishes (fix + answer)

The comment is wrong. Fix it, then write the output.

```
lunch = {"steak", "salmon", "pasta"}
dinner = {"chicken", "potato", "salmon"}

overlap = lunch & dinner  # ________________________
print(len(overlap))
```

Output:

```

```

7) Two lists → common items (write code)

You are given two **lists** (they may have duplicates).

Write code that:

- 1) converts both lists to sets

- 2) finds the intersection
- 3) prints the intersection set

```
list1 = [1, 2, 2, 3, 5]
list2 = [2, 4, 4, 5]

s1 = ____________________  # convert list1 to set
s2 = ____________________  # convert list2 to set
common = ________________  # find the intersection

print(common)
```

Output (order may vary):

```
{2, 5}
```

8) Print each common item (write code)

Write code that prints each item in the intersection (one per line).

```
kid1 = {"lego", "puzzle", "ball", "drone"}
kid2 = {"puzzle", "book", "ball"}

common = ____________________________

for x in common:
    ____________
```

Expected output (order may vary):

```
ball
puzzle
```

9) No overlap?

What is the output?

```
team_red = {"Amy", "Ben"}
team_blue = {"Chloe", "Dylan"}

common = team_red & team_blue

if len(common) == 0:
    print("no overlap")
else:
    print("overlap")
```

Output:

```

```

Part C — Build intersection with a loop

10) Build the common set (fill in the blanks)

This code finds the common items **without** using `&`.

```
a = [4, 7, 2, 9, 5]
b = {2, 5, 8}

common = _____________   # create an empty set

for x in a:
    if x ______ b:
        common._________________

print(common)
```

What is the output? (order may vary)

```

```

11) Trace it (fill in the table)

11) Trace it (fill in the table)

We run this code:

```
a = [1, 2, 2, 3]
b = {2, 3}

common = set()
for x in a:
    if x in b:
        common.add(x)
```

Fill in the table.

Step	x	common (set)
0 (start)	—	`set()`
1	1	_____
2	2	_____
3	2	_____
4	3	_____

12) Fix the bug (intersection)

This code is supposed to print the common items, but it has **two** bugs. Fix the code.

```
a = {1, 2, 3}
b = {2, 3, 4}

common = a & b

for x in b:
print(x)
```

Write the corrected code:

13) Intersection facts (True/False)

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

1. If `x` is in `A & B`, then `x` is in `A` and `x` is in `B`. (True / False)
2. `A & A` is always the same as `A`. (True / False)
3. `A & B` is always the same as `B & A`. (True / False)
4. If `len(A & B) == 0`, then the two sets have no overlap. (True / False)

14) Common letters (what does it print?)

(order may vary)

```
word1 = ["c", "a", "t"]
word2 = ["h", "a", "t"]

s1 = set(word1)
s2 = set(word2)

print(s1 & s2)
```

Output:

15) Three sets (what does it print?)

(order may vary)

(order may vary)

```
a = {1, 2, 3, 4}
b = {2, 4, 6}
c = {0, 2, 4}

common = (a & b) & c
print(common)
```

Output:

```

```

16) Common factors (write code)

Write code that:

- 1) builds a set `f1` of all factors of `12`
- 2) builds a set `f2` of all factors of `18`
- 3) prints the common factors using set intersection (order may vary)

Hint: a factor `i` means `n % i == 0`.

```
n1 = 12
n2 = 18

# find all the factors of n1:
f1 = _____________  # create an empty set
for i in range(_________, _________):
    if ___________________________:
        f1.add(i)

# find all the factors of n2:
f2 = _____________  # create an empty set
for i in range(_________, _________):
    if ___________________________:
        f2.add(i)

common = _________________  # intersection of f1 and f2
```

```
print(common)
```