

# completed\_sets\_assignment\_c\_ANIL

September 28, 2025

## 1 Set Assignments – Solved

Solutions to all 15 set-based Python assignments.

### 1.1 Set Assignment

#### 1.1.1 Assignment 1: Creating and Accessing Sets

Create a set with the first 10 positive integers. Print the set.

```
[1]: s = set(range(1, 11))  
print(s)
```

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

#### 1.1.2 Assignment 2: Adding and Removing Elements

Add the number 11 to the set created in Assignment 1. Then remove the number 1 from the set. Print the modified set.

```
[2]: s = set(range(1, 11))  
s.add(11)  
s.remove(1)  
print(s)
```

{2, 3, 4, 5, 6, 7, 8, 9, 10, 11}

#### 1.1.3 Assignment 3: Set Operations

Create two sets: one with the first 5 positive integers and another with the first 5 even integers. Perform and print the results of union, intersection, difference, and symmetric difference operations on these sets.

```
[3]: A = {1, 2, 3, 4, 5}  
B = {2, 4, 6, 8, 10}  
  
print("Union:", A | B)  
print("Intersection:", A & B)  
print("Difference (A - B):", A - B)  
print("Symmetric Difference:", A ^ B)
```

Union: {1, 2, 3, 4, 5, 6, 8, 10}  
Intersection: {2, 4}  
Difference (A - B): {1, 3, 5}  
Symmetric Difference: {1, 3, 5, 6, 8, 10}

#### 1.1.4 Assignment 4: Set Comprehensions

Create a new set containing the squares of the first 10 positive integers using a set comprehension. Print the new set.

```
[4]: squares = {i**2 for i in range(1, 11)}  
print(squares)
```

{64, 1, 4, 36, 100, 9, 16, 49, 81, 25}

#### 1.1.5 Assignment 5: Filtering Sets

Create a new set containing only the even numbers from the set created in Assignment 1 using a set comprehension. Print the new set.

```
[5]: evens = {x for x in range(1, 11) if x % 2 == 0}  
print(evens)
```

{2, 4, 6, 8, 10}

#### 1.1.6 Assignment 6: Set Methods

Create a set with duplicate elements and remove the duplicates using set methods. Print the modified set.

```
[6]: # Sets cannot have duplicates by design  
# So we start with a list that has duplicates  
lst = [1, 2, 2, 3, 4, 4, 5]  
unique_set = set(lst)  
print(unique_set)
```

{1, 2, 3, 4, 5}

#### 1.1.7 Assignment 7: Subsets and Supersets

Create two sets: one with the first 5 positive integers and another with the first 3 positive integers. Check if the second set is a subset of the first set and if the first set is a superset of the second set. Print the results.

```
[7]: A = {1, 2, 3, 4, 5}  
B = {1, 2, 3}  
  
print("B is subset of A:", B.issubset(A))  
print("A is superset of B:", A.issuperset(B))
```

B is subset of A: True  
A is superset of B: True

### 1.1.8 Assignment 8: Frozenset

Create a frozenset with the first 5 positive integers. Print the frozenset.

```
[8]: fs = frozenset(range(1, 6))  
     print(fs)
```

frozenset({1, 2, 3, 4, 5})

### 1.1.9 Assignment 9: Set and List Conversion

Create a set with the first 5 positive integers. Convert it to a list, append the number 6, and convert it back to a set. Print the resulting set.

```
[9]: s = set(range(1, 6))  
     lst = list(s)  
     lst.append(6)  
     new_set = set(lst)  
     print(new_set)
```

{1, 2, 3, 4, 5, 6}

### 1.1.10 Assignment 10: Set and Dictionary

Create a dictionary with set keys and integer values. Print the dictionary.

```
[10]: # Regular sets are unhashable → cannot be dict keys  
      # Use frozenset instead  
      key1 = frozenset({1, 2})  
      key2 = frozenset({3, 4})  
      d = {key1: 10, key2: 20}  
      print(d)
```

{frozenset({1, 2}): 10, frozenset({3, 4}): 20}

### 1.1.11 Assignment 11: Iterating Over Sets

Create a set and iterate over the elements, printing each element.

```
[11]: s = {10, 20, 30, 40}  
      for item in s:  
          print(item)
```

40  
10  
20  
30

### 1.1.12 Assignment 12: Removing Elements from Sets

Create a set and remove elements from it until it is empty. Print the set after each removal.

```
[12]: s = {1, 2, 3}
print("Start:", s)
while s:
    removed = s.pop()
    print(f"Removed {removed}, Set now: {s}")
```

```
Start: {1, 2, 3}
Removed 1, Set now: {2, 3}
Removed 2, Set now: {3}
Removed 3, Set now: set()
```

### 1.1.13 Assignment 13: Set Symmetric Difference Update

Create two sets and update the first set with the symmetric difference of the two sets. Print the modified first set.

```
[13]: A = {1, 2, 3, 4}
B = {3, 4, 5, 6}
print("Before:", A)
A.symmetric_difference_update(B)
print("After:", A)
```

```
Before: {1, 2, 3, 4}
After: {1, 2, 5, 6}
```

### 1.1.14 Assignment 14: Set Membership Testing

Create a set and test if certain elements are present in the set. Print the results.

```
[14]: s = {2, 4, 6, 8, 10}
print("Is 4 in set?", 4 in s)
print("Is 5 in set?", 5 in s)
print("Is 10 in set?", 10 in s)
```

```
Is 4 in set? True
Is 5 in set? False
Is 10 in set? True
```

### 1.1.15 Assignment 15: Set of Tuples

Create a set containing tuples, where each tuple contains two elements. Print the set.

```
[15]: s = {(1, 'a'), (2, 'b'), (3, 'c')}
print(s)
```

```
{(1, 'a'), (3, 'c'), (2, 'b')}
```