Sets and Maps

Sets

- A **set** is a collection of elements where each element is unique.
- Common set operations:
  - **Union**: The union of two (2) sets (A ∪ B) is the set that contains all the elements in either set.
    Example: {1, 3, 5, 7} ∪ {2, 3, 4, 5} = **{1, 2, 3, 4, 5, 7}**
  - **Intersection**: The intersection of two (2) sets (A ∩ B) is the set that contains only the elements common to both sets.
    Example: {1, 3, 5, 7} ∩ {2, 3, 4, 5} = **{3, 5}**
  - **Difference**: The difference of sets *A* and *B* (A – B) is the set that contains the elements that are in *A* but not in *B*.
    Example: {1, 3, 5, 7} – { 2, 3, 4, 5} = **{1, 7}**
  - **Subset**: Set A is a subset of set B (A ⊂ B) if every element of set A is also an element of set B.
    Example: {1, 3, 5, 7} ⊂ {1, 2, 3, 4, 5, 7} = **true**
- Java contains three (3) general-purpose set implementations. All are included in the **`java.util`** package.
  - **`HashSet`** – This stores its elements in a hash table without a guaranteed order upon iteration. This is the best-performing implementation.
  - **`TreeSet`** – This stores its elements in a special type of tree where elements are sorted (natural or custom) during iteration.
  - **`LinkedHashSet`** – This stores its elements in a hash table with a linked list running through it. The order of the elements during the iteration is the same as the order they were inserted into the set.
- Sample codes for sets in Java:

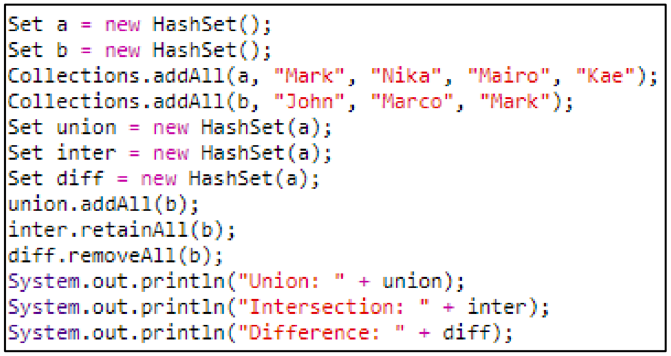
1. To create an empty set:

```
Set a = new HashSet();
Set b = new TreeSet ();
Set c = new LinkedHashSet();
```

2. To add items to the set:

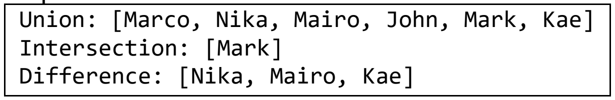
```
Collections.addAll(a, "Mark", "Nika", "Mairo", "Kae");
//equivalent to multiple use of add()
```

3. To determine the union, intersection, and difference:



```
Set a = new HashSet();
Set b = new HashSet();
Collections.addAll(a, "Mark", "Nika", "Mairo", "Kae");
Collections.addAll(b, "John", "Marco", "Mark");
Set union = new HashSet(a);
Set inter = new HashSet(a);
Set diff = new HashSet(a);
union.addAll(b);
inter.retainAll(b);
diff.removeAll(b);
System.out.println("Union: " + union);
System.out.println("Intersection: " + inter);
System.out.println("Difference: " + diff);
```

Output:



```
Union: [Marco, Nika, Mairo, John, Mark, Kae]
Intersection: [Mark]
Difference: [Nika, Mairo, Kae]
```

4. To determine whether a set is a subset of another set:

```
System.out.println(a.containsAll(b));
```

- Curly braces or the **`set()`** function can be used to implement sets in Python.
- Sample codes for sets in Python:

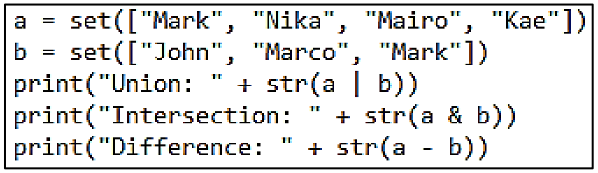
1. To create an empty set:

```
a = set()
```

2. To initialize a set:

```
a = set(["Mark", "Nika", "Mairo", "Kae"])
b = {"John", "Marco", "Mark"}
#Equivalent to multiple use of add()
```

3. To determine the union, intersection, and difference:



```
a = set(["Mark", "Nika", "Mairo", "Kae"])
b = set(["John", "Marco", "Mark"])
print("Union: " + str(a | b))
print("Intersection: " + str(a & b))
print("Difference: " + str(a - b))
```

4. To determine whether a set is a subset of another set:

```
print(a.issubset(b))
```