

# Seminar Errors and Data Structures

## Activity 1: Errors

Incorporate the following code into a Python program to handle exceptions.

```
try:
    # do something
    pass

except ValueError:
    # handle ValueError exception
    pass

except (TypeError, ZeroDivisionError):
    # handle multiple exceptions
    # TypeError and ZeroDivisionError
    pass

except:
    # handle all other exceptions
    Pass
```

Look at: [https://github.com/gicanon/handle\\_exceptions](https://github.com/gicanon/handle_exceptions)

## Activity 2: Data Structures

1. Set operations include:
  - Union
  - Intersection
  - Difference
  - Symmetric difference

Explain a use for each of these set operations within the context of your summative assessment.

2. Write a Python program to carry out a linear search on a list data structure.

### 1. Set Operation

**Union:** The set method 'union()' joins all elements from the original set and the specified set. It can also be any iterable object, not just a set. Also, duplicate elements are not displayed. Especially the last aspect is a powerful attribute to display only data once to save data storage (W3schools, 2022a)

Input:

Output:

```
x = {"a", "b", "c"}
y = {"f", "d", "a"}
z = {"c", "d", "e"}
{'b', 'e', 'a', 'f', 'd', 'c'}
```

```
result = x.union(y, z)
```

```
print(result)
```

**Intersection:** The set method 'intersection()' returns a set with only the elements that have a similarity in both sets. Another powerful method for only recognising elements contained multiple times in different sets (W3schools, 2022b).

Input:

Output:

```
x = {"a", "b", "c"}
y = {"c", "d", "e"}
z = {"f", "g", "c"}
{'c'}
```

```
result = x.intersection(y, z)
```

```
print(result)
```

**Difference:** The set method 'difference()' returns a set and only displays the elements from one set with no similarity (W3schools, 2022c).

Input:

```
x = {"apple", "banana", "cherry"}  
y = {"google", "microsoft", "apple"}
```

```
z = x.difference(y)
```

```
print(z)
```

Output:

```
{'banana', 'cherry'}
```

**Symmetric difference:** In contrast to difference, the method 'symmetric\_difference()' returns a set with elements which have no similarity from both sets. This method can be used for recognising those elements which are new and do not have any similarities (W3schools, 2022d).

Input:

```
x = {"apple", "banana", "cherry"}  
y = {"google", "microsoft", "apple"}
```

```
z = x.symmetric_difference(y)
```

```
print(z)
```

Output:

```
{'banana', 'cherry', 'google',  
'microsoft'}
```

## References:

W3schools (2022a) Python Set union() Method. Available from: [https://www.w3schools.com/python/ref\\_set\\_union.asp](https://www.w3schools.com/python/ref_set_union.asp) [Accessed 07 August 2022].

W3schools (2022b) Python Set intersection() Method. Available from: [https://www.w3schools.com/python/ref\\_set\\_intersection.asp](https://www.w3schools.com/python/ref_set_intersection.asp) [Accessed 07 August 2022].

W3schools (2022c) Python Set difference() Method. Available from: [https://www.w3schools.com/python/ref\\_set\\_difference.asp](https://www.w3schools.com/python/ref_set_difference.asp) [Accessed 07 August 2022].

W3schools (2022d) Python Set symmetric\_difference() Method. Available from: [https://www.w3schools.com/python/ref\\_set\\_symmetric\\_difference.asp](https://www.w3schools.com/python/ref_set_symmetric_difference.asp) [Accessed 07 August 2022].

## 2. Linear Search

Look at: [https://github.com/gicanon/simple\\_linear\\_search](https://github.com/gicanon/simple_linear_search)