Unit 5 Activity: Equivalence Testing in Python

Task:

Run equivalence.py in your chosen Jupyter Notebook workspace - Testing with Python - which is an implementation of equivalence partitioning. This test partitions integers [-3,5] into equivalence classes based on lambda x, y: (x-y)%4 == 0.

In the output, you should be able to see how a set of objects to be partitioned are considered, and a function evaluates if the two objects are equivalent before printing the result.

test_equivalence_partition() produces the following output:

set([1, -3]) set([2, -2]) set([3, -1]) set([0, 4]) 0 : set([0, 4]) 1 : set([1, -3]) 2 : set([2, -2]) 3 : set([3, -1]) 4 : set([0, 4]) -2 : set([2, -2]) -3 : set([1, -3]) -1 : set([3, -1])

You should carry out further investigations on the code and experiment with it.

Answer:

The script executed successfully and produced the following output:

C: > Users > pc > Documents > 00000_Essex > 03_Secure Software Development > 05_Unit > Activity-Evquivalence Testing in Python > equ

```python
# CODE SOURCE: https://stackoverflow.com/questions/38924421/is-there-a-standard-way-to-partition-an-

def equivalence_partition(iterable, relation):
    """Partitions a set of objects into equivalence classes

    Args:
        iterable: collection of objects to be partitioned
        relation: equivalence relation. I.e. relation(o1,o2) evaluates to True
            if and only if o1 and o2 are equivalent

    Returns: classes, partitions
        classes: A sequence of sets. Each one is an equivalence class
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\pc> & C:/Users/pc/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/pc/Documents/000
tivity-Evquivalence Testing in Python/equivalence.py"
{1, -3}
{2, -2}
{3, -1}
{0, 4}
-3 : {1, -3}
-2 : {2, -2}
-1 : {3, -1}
0 : {0, 4}
1 : {1, -3}
2 : {2, -2}
3 : {3, -1}
4 : {0, 4}
PS C:\Users\pc>
```

Equivalence classes:

```
{1, -3}
{2, -2}
{3, -1}
{0, 4}
```

Mapping of each integer to its equivalence class:

```
-3 : {1, -3}
-2 : {2, -2}
-1 : {3, -1}
0  : {0, 4}
1  : {1, -3}
2  : {2, -2}
3  : {3, -1}
4  : {0, 4}
```

This matches the partitioning based on the equivalence relation $(x-y) \bmod 4 = 0$. Each set contains numbers that are equivalent modulo 4.