Activity 2: Data Structures

Set Operations Explained:

1. Union: Combines all elements from two sets, removing duplicates.
    - Use Case: Union can be used in the context of an autonomous vehicle to merge data from several sensors to provide a comprehensive picture of the surroundings. Combining items that are identified by the front and back cameras, for instance.
2. Intersection: Returns elements that are common to both sets.
    - Use Case: A common obstacle observed by several sensors may be found using intersection, which helps in validating the presence of such obstacle.
3. Difference: Returns elements that are in one set but not in the other.
    - Use Case: Difference can help in the detection of moving things by allowing the sensors to distinguish new items that were not there during a prior scan.
4. Symmetric Difference: Returns elements that are in either set, but not in both.
    - Use Case: By comparing the present collection of observed items with a prior set, symmetric difference may be utilized to identify changes in the surrounding environment.

Python Code for Example of Set Operations in a Driverless Car Context:

```
front_camera_detections = {"car", "pedestrian", "bicycle"}

rear_camera_detections = {"car", "truck", "bicycle"}


# Union
```

```
all_detections = front_camera_detections | rear_camera_detections
print("All Detections (Union):", all_detections)

# Intersection
common_detections = front_camera_detections & rear_camera_detections
print("Common Detections (Intersection):", common_detections)

# Difference
new_detections = front_camera_detections - rear_camera_detections
print("New Detections (Difference):", new_detections)

# Symmetric Difference
changed_detections = front_camera_detections ^ rear_camera_detections
print("Changed Detections (Symmetric Difference):",
changed_detections)
```