

# Class\_Act28

## Abstraction Activity: Smart Cleaning Equipment (with a Twist!)

### Objective:

Reinforce the concept of abstraction while:

- Adding constructor parameters
  - Introducing a new method per subclass
  - Encouraging use of a loop to interact with multiple objects
- 

### Updated Instructions:

1. Import `ABC` and `abstractmethod` from `abc`.
2. Create an abstract class `Cleaner`:
  - Attribute: `brand`
  - Abstract method: `start_cleaning()`
3. Create 3 subclasses:
  - `Vacuum`
    - Add attribute: `power_level`
    - `start_cleaning()` prints:  
`"Vacuum [brand] at power level [power_level] is cleaning."`
  - `Mop`
    - Add attribute: `water_tank` (in liters)
    - `start_cleaning()` prints:  
`"Mop [brand] with [water_tank]L water is cleaning."`
  - `Sweeper`
    - Add attribute: `speed`
    - `start_cleaning()` prints:  
`"Sweeper [brand] running at [speed] RPM is cleaning."`
4. Create a list of all 3 cleaners and use a loop to call `start_cleaning()` on each.