S19 - Classes Part I

Start Assignment

- Due Thursday by 23:59
- Points 500
- Submitting a text entry box or a website url
- Available after 14 Nov at 8:30

Session 19 - Classes Part - I

Topics Covered in the class are:

- Objects and Classes
- Class Attributes
- Callable Class Attributes
- Classes are Callable
- Data Attributes
- Function Attributes
- Initializing Class Attributes
- Creating Attributes at Run-Time

Part II (next session) will cover:

- Properties
- Property Decorators
- Read-Only and COmputed Properties
- Deleting Properties
- Class and Static MEthods
- Class Body Scope

The class code can be **found here (https://canvas.instructure.com/courses/9532398/files/276924229?wrap=1)** (https://canvas.instructure.com/courses/9532398/files/276924229/download?download_frd=1) .

Assignment

Your assignment Problem Statement is:

Create a class named *SmartDevice* that mimics the behavior of a smart device with the following functionalities:

1. **Initialization:**
    - The class should have an initializer (__init__) that accepts:
        - device_name (string)
        - model_number (string)
        - is_online (boolean, default is False)
    - The class should also have a class attribute device_count which keeps track the total number of devices created
2. **Attributes:**
    - SmartDevice should have the following attributes:
        - device_name: The name of the device
        - model_number: The model number of the device
        - is_online: A boolean indicating if the device is currently online
        - status: A dictionary that stores the current status of various device features (e.g., battery level, temperature)
3. **Methods:**
    - update_status(attribute, value): Adds or updates a status attribute in the status dictionary
    - get_status(attribute): Returns the value of a specific status attribute. If the attribute does not exist, it should return 'Attribute not found'.
    - toggle_online(): Changes the device's online status (is_online) to its opposite value
    - reset():L Resets all status attributes to their default values (i.e. clears the status dictionary)
4. **Callable Class:**

- Make the SmartDevice class callable, such that calling an instance returns its device_name and model_number in a formatting string

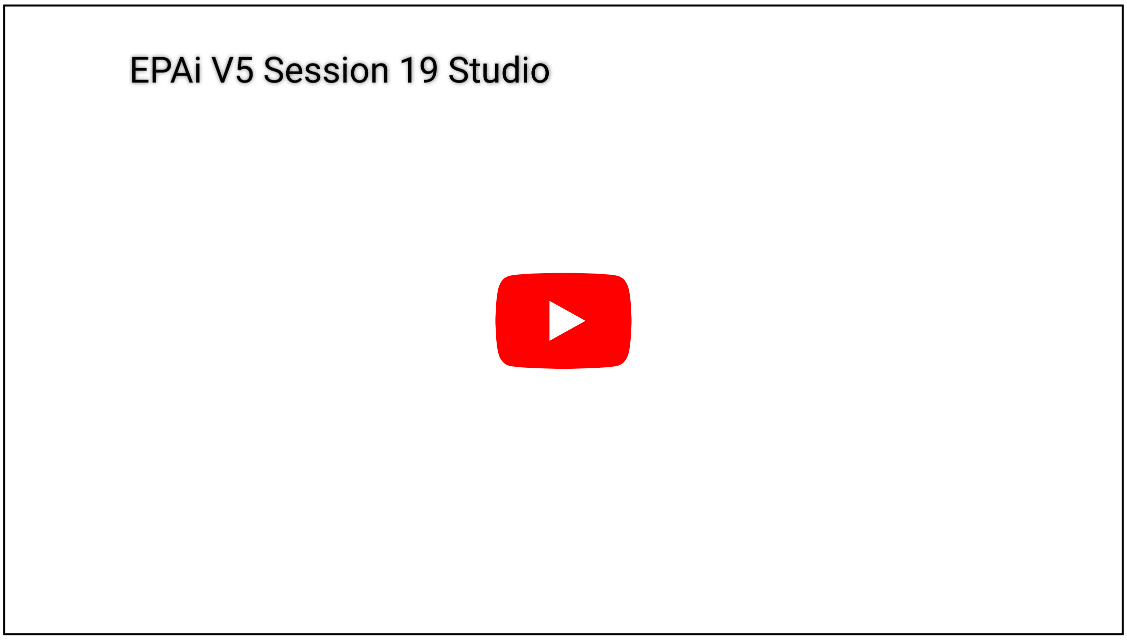
5. **Function Attributes:**
   - Add a callable function attribute to the class named device_info which returns the current state of the device as a dictionary

Submission Requirements:

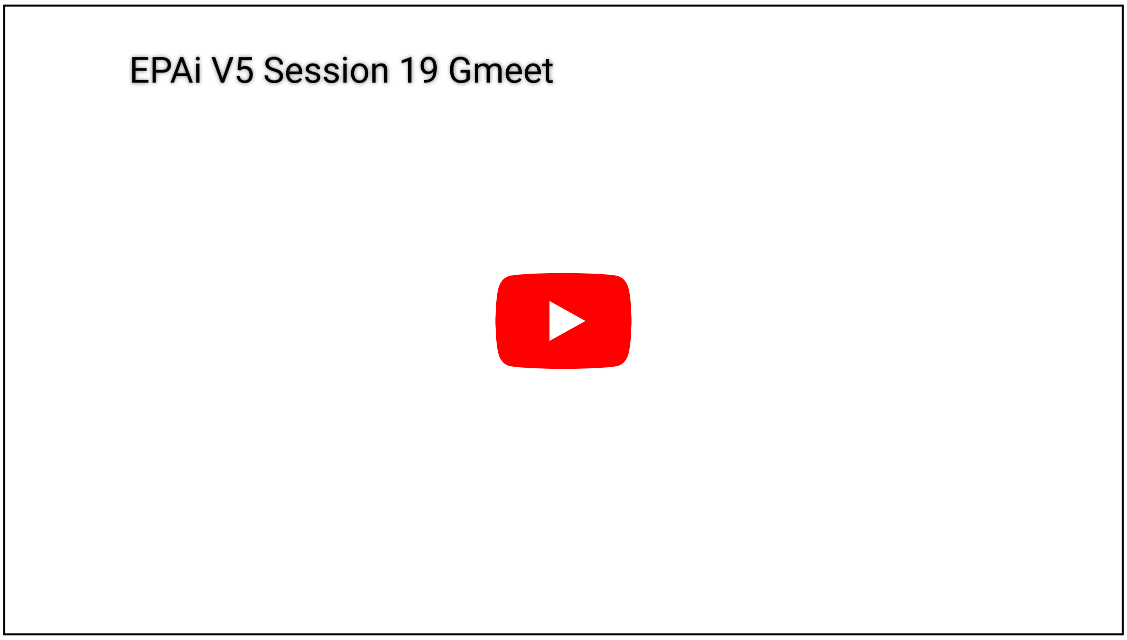
1. Create a Python file **smart_device.py (https://canvas.instructure.com/courses/9532398/files/276555232?wrap=1)** ⤓ **(https://canvas.instructure.com/courses/9532398/files/276555232/download?download_frd=1)** that contains the implementation of the SmartDevice class.
2. Use the provided test_smart_device.py file to validate your code
3. Use GitHub actions to automatically run tests whenever you push your code. The GitHub Actions workflow file is also **provided (https://canvas.instructure.com/courses/9532398/files/276555249?wrap=1)** ⤓ **(https://canvas.instructure.com/courses/9532398/files/276555249/download?download_frd=1)** .
4. Once you've passed all the tests, please share your GitHub Actions Link with us along with a screenshot (especially if you are clearing the tests on your desktop, then screenshot and GitHub code link)!

Video

Studio


EPAi V5 Session 19 Studio

GMeet


EPAi V5 Session 19 Gmeet