**Design Document for Google Classroom API Integration**

**Objective:**

The goal of this document is to define all CRUD (Create, Read, Update, Delete) operations that interact with the Google Classroom API for managing courses, topics, and coursework. The operations aim to automate course creation, topic management, and coursework synchronization.

**1. Courses (Classes)**

**Use Case 1: Create or Update Course**

* **Description**: Create a new course or update an existing course based on the course name.
* **Operations**:
  + **Create**: If the course doesn't already exist, create a new course using the provided data.
  + **Read**: List all existing courses.
  + **Update**: If the course already exists, ensure that no changes are made if they are up-to-date.
  + **Delete**: Not applicable in this use case, as the script only manages course creation.
* **API Calls**:
  + courses().list() – Read all courses.
  + courses().create() – Create a new course.

**Use Case 2: List Existing Courses**

* **Description**: Retrieve all existing courses in Google Classroom.
* **Operations**:
  + **Read**: List all courses, including course ID and name.
* **API Calls**:
  + courses().list() – Read all courses.

**2. Topics (Course Topics)**

**Use Case 3: Create or Update Topics**

* **Description**: Create new topics or update existing topics for a given course. This ensures topics are synchronized with the desired state.
* **Operations**:
  + **Create**: If the topic doesn't already exist in the course, create a new topic.
  + **Read**: List all topics within a course.
  + **Update**: If the topic exists, no changes will be made unless the topic is updated.
  + **Delete**: Remove topics that are no longer present in the desired state (i.e., delete any topics not in the provided list).
* **API Calls**:
  + courses().topics().list(courseId) – Read existing topics.
  + courses().topics().create(courseId, body) – Create new topic.
  + courses().topics().delete(courseId, id) – Delete existing topic.

**Use Case 4: List Topics in a Course**

* **Description**: Retrieve the list of topics within a course to verify and check against desired topics.
* **Operations**:
  + **Read**: List all topics in a course.
* **API Calls**:
  + courses().topics().list(courseId) – Read all topics within a specific course.

**3. Coursework (Assignments)**

**Use Case 5: Create or Update Coursework**

* **Description**: Create new coursework or update existing coursework in a course's topic.
* **Operations**:
  + **Create**: If the coursework doesn't already exist, create a new assignment under a specific topic.
  + **Read**: List all coursework within a course.
  + **Update**: If the coursework exists but has changed (e.g., description or materials), delete the outdated coursework and create a new one.
  + **Delete**: Remove coursework that is not in the desired list (i.e., delete any assignments not specified).
* **API Calls**:
  + courses().courseWork().list(courseId) – Read all coursework in a course.
  + courses().courseWork().create(courseId, body) – Create new coursework.
  + courses().courseWork().delete(courseId, id) – Delete existing coursework.

**Use Case 6: List Coursework in a Course**

* **Description**: Retrieve all coursework (assignments) within a specific course to verify and compare with the desired assignments.
* **Operations**:
  + **Read**: List all assignments in a course.
* **API Calls**:
  + courses().courseWork().list(courseId) – Read all coursework in a course.

**Summary of CRUD Operations**

| **Entity** | **Create** | **Read** | **Update** | **Delete** |
| --- | --- | --- | --- | --- |
| **Course** | courses().create(body) | courses().list() | N/A | N/A |
| **Topic** | courses().topics().create(courseId, body) | courses().topics().list(courseId) | N/A | courses().topics().delete(courseId, id) |
| **Coursework** | courses().courseWork().create(courseId, body) | courses().courseWork().list(courseId) | Delete outdated and recreate coursework | courses().courseWork().delete(courseId, id) |

**Error Handling and Validation**

* **Invalid Course ID**: If a course ID is invalid during the operations, return an error message.
* **Topic Creation Failures**: If a topic creation fails, the system should handle it by logging an error and not proceeding with further topic or coursework operations for that course.
* **Permissions Errors**: Handle cases where the current authenticated user does not have permission to access, create, update, or delete resources in Google Classroom.

**Authentication and Authorization Flow**

* Use OAuth2 authentication with the credentials.json file.
* Store the authentication token in token.pickle for future use.
* Handle token expiration and refresh using the creds.refresh(Request()) method.

**Conclusion**

This design document outlines all the CRUD operations for courses, topics, and coursework using the Google Classroom API. It ensures proper management of Google Classroom resources, with idempotent operations to prevent unnecessary creation or deletion of entities.