**Abstraction with YouTube videos**

**1-Class Comment**

* Responsibility: Represents a comment on a video.
* Methods:
  + Comment(string name, string text): Constructor to initialize a comment.
* Attributes:
  + string Name: Name of the commenter.
  + string Text: Comment text.

**2-Class Video**

* Responsibility: Represents a video and manages its comments.
* Methods:
  + Video(string title, string author, int length): Constructor.
  + void AddComment(string name, string text): Adds a comment to the video.
  + int GetCommentCount(): Returns the number of comments.
  + void DisplayInfo(): Displays video information and its comments.
* Attributes:
  + string Title: Title of the video.
  + string Author: Video creator's name.
  + int Length: Duration in seconds.
  + List<Comment> Comments: List of comments associated with the video.

**3-Class Program**

* + Responsibility: Orchestrates the program execution.
  + Methods: static void Main(): Entry point, creates videos, and displays information.

**Program Workflow:**

1. Create Videos:
   * Instantiate 3-4 Video objects with title, author, and length.
2. Add Comments:
   * Each video gets 3-4 Comment objects (name + text).
3. Store Videos:
   * Save all videos in a list.
4. Display Information:
   * Loop through the list and call DisplayInfo() to print:
     + Video title, author, and length.
     + Number of comments.
     + All comments.

[**Encapsulation with Online Ordering**](https://byui-cse.github.io/cse210-ww-course/week04/foundation-encapsulation.html)

**1-Class Address**

* Attributes:
  + street: string
  + city: string
  + state: string
  + country: string
* Methods:
  + IsInUSA() -> bool → Returns True if country is "USA".
  + GetFullAddress() -> string → Returns formatted address.

**2- Class Customer**

* Attributes:
  + name: string
  + address: Address
* Methods:
  + IsInUSA() -> bool → Calls IsInUSA() from Address.
  + GetShippingLabel() -> string → Returns customer name + full address.

**3-Class Product**

* **Attributes:**
  + name: string
  + productID: int
  + price: float
  + quantity: int
* **Methods:**
  + GetTotalCost() -> float → Returns price \* quantity.

**4-Class Order**

* **Attributes:**
  + customer: Customer
  + products: List<Product>
* **Methods:**
  + GetTotalCost() -> float → Sums product costs + shipping fee.
  + GetPackingLabel() -> string → Returns list of products with name & ID.
  + GetShippingLabel() -> string → Calls GetShippingLabel() from Customer.

**Program Workflow:**

1. Create Customers → Make 2 customers with names and addresses.
2. Create Products → Add 2-3 products for each order.
3. Create Orders → Make 2 orders, each with a customer and products.
4. Show Results:
   * Packing Label → List product names & IDs.
   * Shipping Label → Show customer name & address.
   * Total Price → Sum of product costs + shipping fee.