**Program 1: Abstraction with YouTube Videos**

Description of Classes

**1. Video Class**

Attributes:

title (string): The title of the video.

author (string): The author of the video.

length (int): The length of the video in seconds.

comments (List<Comment>): A list to store comments associated with the video.

Methods:

GetCommentCount(): This method returns the number of comments on the video.

The Video class contains a list of Comment objects, demonstrating a one-to-many relationship.

**2. Comment Class**

Attributes:

name (string): The name of the person who made the comment.

text (string): The text of the comment.

Comment class is simple with only properties to hold the commenter's name and the comment text.

Relationships:

The Video class has a one-to-many relationship with the Comment class, meaning each video can have multiple comments.

**Program 2: Encapsulation with Online Ordering**

Description of Classes

**1. Order Class**

Attributes:

products (List<Product>): A list of products in the order.

customer (Customer): The customer who placed the order.

Methods:

GetTotalCost(): Calculates the total cost of the order.

GetPackingLabel(): Returns a string for the packing label.

GetShippingLabel(): Returns a string for the shipping label.

**2. Product Class**

Attributes:

name (string): The name of the product.

id (string): The product ID.

price (double): The price per unit of the product.

quantity (int): The quantity of the product.

Methods:

GetTotalCost(): Computes the total cost of the product.

**3. Customer Class**

Attributes:

name (string): The name of the customer.

address (Address): The address of the customer.

Methods:

IsInUSA(): Returns whether the customer lives in the USA or not.

**4. Address Class**

Attributes:

street (string): The street address.

city (string): The city.

state (string): The state or province.

country (string): The country.

Methods:

IsInUSA(): Returns whether the address is in the USA or not.

GetFullAddress(): Returns a string with all fields combined appropriately.

Summary:

Order Class: Manages a list of products and customer information, computes total cost, and provides packing and shipping labels.

Product Class: Contains details about a product and computes its total cost.

Customer Class: Contains customer details and checks if the customer is in the USA.

Address Class: Manages address details and checks if the address is in the USA.

This code will follow the encapsulation principle, ensuring that all member variables are private and accessed via public methods or properties. It creates instances of orders with products, and displays packing labels, shipping labels, and total costs as specified.

**Program 3: Inheritance with Event Planning**

Description of Classes

**1.Event Class:**

The base class containing common attributes and methods for all types of events.

Attributes:

title (string): The title of the event.

description (string): Description of the event.

date (DateTime): Date of the event.

time (TimeSpan): Time of the event.

address (Address): Address of the event.

Methods:

GenerateStandardMessage(): Returns a string with standard event details.

GenerateFullMessage(): Returns a string with full event details.

GenerateShortDescription(): Returns a string with a short description of the event.

**2.Lecture Class:**

Derived class from Event for lecture-type events.

Attributes:

speaker (string): Name of the speaker.

capacity (int): Capacity of the lecture.

**3.Reception Class:**

Derived class from Event for reception-type events.

Attributes:

rsvpEmail (string): Email address for RSVP.

**4.OutdoorGathering Class:**

Derived class from Event for outdoor gathering-type events.

Attributes:

weatherForecast (string): Forecast of the weather.

Each class is encapsulated with private member variables and provides methods to generate different types of event messages.

**Program 4: Polymorphism with Exercise Tracking**

Description of Classes

**1.Activity Class:**

The base class containing common attributes and methods for all types of activities.

Attributes:

date (DateTime): Date of the activity.

minutes (int): Duration of the activity in minutes.

Methods:

virtual GetDistance(): double: Returns the distance covered during the activity.

virtual GetSpeed(): double: Returns the speed achieved during the activity.

virtual GetPace(): double: Returns the pace of the activity.

**2.Running Class:**

Derived class from Activity for running activities.

Attributes:

distance (double): Distance covered during the run.

**3.Cycling Class:**

Derived class from Activity for cycling activities.

Attributes:

speed (double): Speed achieved during the cycling activity.

**4.Swimming Class:**

Derived class from Activity for swimming activities.

Attributes:

laps (int): Number of laps swum during the activity.