Foundation Program Design

Program 1: Abstraction with YouTube Videos

Objective:

To create a system that stores YouTube video details and user comments using abstraction.

Classes and Responsibilities:

Video:

Attributes:

title (string)

author (string)

length (int, in seconds)

comments (list of Comment objects)

Methods:

add\_comment(comment: Comment): Adds a comment to the video.

get\_comment\_count() -> int: Returns the number of comments.

display\_info(): Displays video details and all comments.

Comment

Attributes:

username (string)

text (string)

Methods:

\_\_str\_\_(): Returns a formatted string of the comment.

Class Diagram:

+------------------+

| Video |

+------------------+

| - title: str |

| - author: str |

| - length: int |

| - comments: [] |

+------------------+

| + add\_comment() |

| + get\_comment\_count() |

| + display\_info() |

+------------------+

+------------------+

| Comment |

+------------------+

| - username: str |

| - text: str |

+------------------+

| + \_\_str\_\_() |

+------------------+

Program 2: Encapsulation with Online Ordering

Objective:

To implement a simple online ordering system using encapsulation.

Classes and Responsibilities:

Order

Attributes:

products (list of Product objects)

customer (Customer object)

Methods:

calculate\_total\_price() -> float: Computes total price including shipping.

generate\_packing\_label() -> str: Returns a packing label listing product details.

generate\_shipping\_label() -> str: Returns a shipping label with customer details.

Product

Attributes:

name (string)

product\_id (string)

price (float)

quantity (int)

Methods:

calculate\_total\_cost() -> float: Computes total cost for the product.

Customer

Attributes:

name (string)

address (Address object)

Methods:

is\_in\_usa() -> bool: Checks if the customer is in the USA.

Address

Attributes:

street (string)

city (string)

state (string)

country (string)

Methods:

is\_in\_usa() -> bool: Returns True if country is USA.

\_\_str\_\_(): Returns a formatted address string.

Class Diagram:

+------------------+

| Order |

+------------------+

| - products: [] |

| - customer: Customer |

+------------------+

| + calculate\_total\_price() |

| + generate\_packing\_label() |

| + generate\_shipping\_label() |

+------------------+

+------------------+

| Product |

+------------------+

| - name: str |

| - product\_id: str |

| - price: float |

| - quantity: int |

+------------------+

| + calculate\_total\_cost() |

+------------------+

+------------------+

| Customer |

+------------------+

| - name: str |

| - address: Address |

+------------------+

| + is\_in\_usa() |

+------------------+

+------------------+

| Address |

+------------------+

| - street: str |

| - city: str |

| - state: str |

| - country: str |

+------------------+

| + is\_in\_usa() |

| + \_\_str\_\_() |

+------------------+