**Program 1: Abstraction with YouTube Videos**

1. **Class Video:**

|  |
| --- |
| Video |
| \_title :string |
| \_author :string |
| \_lenght :int |
| \_comments : List<Comment> |
| GetNumberOfComments():int |
| GetComments():List<Comment> |
| AddComment(Comment):void |

1. **Class Comment**

|  |
| --- |
| Comment |
| \_commenterName : string |
| \_text : string |
| Constructor (commenterName: string, text: string) |

1. **Class Program :**
2. Create Video objects.
3. Add Comment objects to Video objects.
4. Display each video's details and comments.

**Program 2: Encapsulation with Online Ordering**

* 1. **Class Order:**

|  |
| --- |
| Order |
| \_productList: List<Product> |
| \_customer : Customer |
| Public Order(productlist : List<Product>, customer : Customer) |
| GetTotalCost(): float  GetPackingLabel(): string |
| GetShippingLabel(): string |

* 1. **Product**

|  |
| --- |
| Product |
| \_name : string |
| \_productId : string |
| \_price : float |
| \_quantity : int |
| Public Product(name : string, id:string, price: float, quantity: int) |
| GetTotalCost(): decimal |

* 1. **Customer**

|  |
| --- |
| Customer |
| \_name : string |
| \_address : Address |
| public Customer( name : string, address:Address) |
| IsInGhana(): bool |

* 1. Address

|  |
| --- |
| Address |
| \_street : string |
| \_city: string |
| \_state : string |
| \_country : string |
| public Address( street:string, city:string, state : string, country: string) |
| IsInGhana() : bool |
| DisplayAddress() : string |

**Class Program**

1. Create Product objects.
2. Create Address and Customer objects.
3. Create Order objects and add Product objects to them.
4. Calculate total cost, get packing label, and get shipping label.

**Program 3: Inheritance with Event Planning**

**Class Event**

|  |
| --- |
| **Event(base class)** |
| \_title : string |
| \_description : string |
| \_date : string |
| \_time : string |
| \_address : Address |
| public Event(title : string, description: string, date:string, time: string, address: string) |
| DisplayStdDetails() : string |
| ShortDescripton() : string |

|  |  |  |
| --- | --- | --- |
| **Lectures:Event**  **(derived class)** | **Receptions : Event (derived class)** | **Outdoor : Event**  **(derived class)** |
| \_speaker : string  \_capcity :string | \_email : string | \_weather : string |
| public Lectures(title : string, description: string, date:string, time: string, address: string, speaker:string, capcity:string) | public Receptions(title : string, description: string, date:string, time: string, address: string, email:string) | public Outdoor(title : string, description: string, date:string, time: string, address: string, weather :string) |
| FullDetais() :string | FullDetais() :string | FullDetais() :string |

|  |
| --- |
| Address **(derived class)** |
| \_street : string |
| \_city: string |
| \_state : string |
| \_country : string |
| public Address( street:string, city:string, state : string, country: string) |
| FromUS() : bool |
| DisplayAddress() : string |
|  |

#### Class Program

Create Address objects.

Create Event objects (Lecture, Reception, OutdoorGathering).

Set event details.

Display standard details, full details, and short description.

**Program 4: Polymorphism with Exercise Tracking**

**Class Diagram**

|  |
| --- |
| Address **(Base class)** |
| \_date: string |
| \_length: int |
| GetDistance(): virtual double |
| GetSpeed(): virtual double |
| GetPace(): virtual double  GetSummary(): virtual string |

|  |
| --- |
| Running **(derived class)** |
| Distance: float |
|  |
| GetDistance(): override double |
| GetSpeed(): overide double |
| GetPace(): override double  GetSummary(): override string |

|  |
| --- |
| Cycling **(derived class)** |
| Speed: float |
|  |
| GetDistance(): override double |
| GetSpeed(): overide double |
| GetPace(): override double  GetSummary(): override string |

|  |
| --- |
| Swimming **(derived class)** |
| laps: int |
|  |
| GetDistance(): override double |
| GetSpeed(): overide double |
| GetPace(): override double  GetSummary(): override string |

#### Class Program

1. Create Activity objects (Running, Cycling, Swimming).
2. Set activity details.
3. Store activities in a list.
4. Iterate through the list and display summaries.