**Plan to Reach Level 3 of the Mini Project**

**Step 1: Define Classes and Properties**

1. **Create Asset Base Class**:
   * Properties: **ModelName**, **PurchaseDate**, **PriceInUSD**
   * Constructor to initialize properties
2. **Create Derived Classes for Assets**:
   * **Laptop** and **MobilePhone** classes inheriting from **Asset**
   * Additional properties if needed
3. **Create Office Class**:
   * Properties: **OfficeName**, **Currency**
   * Constructor to initialize properties

**Step 2: Implement Currency Conversion**

1. **Create a CurrencyConverter Class**:
   * Method to convert USD to other currencies based on today's rates
   * Use a dictionary to store conversion rates

**Step 3: Input and Store Data**

1. **Create Methods for User Input**:
   * Methods to input asset details and office details
   * Store assets in a list

**Step 4: Sorting and Color-Coding**

1. **Implement Sorting Logic**:
   * Sort assets by office, then by purchase date
2. **Implement Color-Coding Logic**:
   * Mark items red if purchase date is less than 3 months away from 3 years
   * Mark items yellow if purchase date is less than 6 months away from 3 years

**Step 5: Display Data**

1. **Create Method to Display Assets**:
   * Display sorted and color-coded list of assets
   * Show prices in the appropriate currency for each office

**Step 6: Integrate All Components**

1. **Main Program Logic**:
   * Integrate all methods and classes
   * Ensure smooth flow from input to display

**Example Code Structure**

using System;

using System.Collections.Generic;

using System.Globalization;

public class Asset

{

public string ModelName { get; set; }

public DateTime PurchaseDate { get; set; }

public decimal PriceInUSD { get; set; }

public Asset(string modelName, DateTime purchaseDate, decimal priceInUSD)

{

ModelName = modelName;

PurchaseDate = purchaseDate;

PriceInUSD = priceInUSD;

}

}

public class Laptop : Asset

{

public Laptop(string modelName, DateTime purchaseDate, decimal priceInUSD)

: base(modelName, purchaseDate, priceInUSD) { }

}

public class MobilePhone : Asset

{

public MobilePhone(string modelName, DateTime purchaseDate, decimal priceInUSD)

: base(modelName, purchaseDate, priceInUSD) { }

}

public class Office

{

public string OfficeName { get; set; }

public string Currency { get; set; }

public Office(string officeName, string currency)

{

OfficeName = officeName;

Currency = currency;

}

}

public static class CurrencyConverter

{

private static readonly Dictionary<string, decimal> ConversionRates = new Dictionary<string, decimal>

{

{ "USD", 1.0m },

{ "EUR", 0.85m },

{ "JPY", 110.0m }

};

public static decimal ConvertToCurrency(decimal amountInUSD, string currency)

{

if (ConversionRates.ContainsKey(currency))

{

return amountInUSD \* ConversionRates[currency];

}

throw new ArgumentException("Unsupported currency");

}

}

public class Program

{

public static void Main()

{

// Input and store data

// Sort and color-code data

// Display data

}

}