CDAC MUMBAI

Module: MS .NET

Assignment-1 (Basic Object Orientation in MS .NET)

---

1. Personal Expense Tracker (Console Application with Class Library)

Problem Statement:
Create a console application for managing personal expenses.

- Implement a **Class Library** that defines classes like `Expense` and `ExpenseManager`.
- Use **project references** to link the library to the console application.
- Add methods to:
  - Add a new expense.
  - List all expenses.
  - Calculate total expenses.
  - Filter expenses by category.

2. Student Management System

Problem Statement:
Create a console-based system for managing student records.

- Define a `Student` class with properties like `Name`, `RollNumber`, `Marks`, and `Grade`.
- Use **get** and **set** for properties and implement **readonly** properties for derived values like `Grade`.
- Implement methods with:
  - **Method Overloading** to calculate the grade using either total marks or individual subject marks.
  - **Optional Parameters** for default grading criteria.
  - **Named and Positional Parameters** to add student records.

3. Inventory Management System

Problem Statement:
Build an inventory management system for a store.

- Define a `Product` class with properties like `ID`, `Name`, `Price`, and `Stock`.
- Use **Constructors** and **Object Initializers** to create instances.
- Add a method to calculate the total value of all items in stock using the **params** keyword for varying quantities.
- Implement a **Local Function** within a method to validate product stock.

4. Library System with Book Borrowing

Problem Statement:
Create a library management system that tracks book borrowing and returns.

- Create a `Book` class with properties `Title`, `Author`, and `IsBorrowed`.
- Implement a **Readonly Property** to calculate if a book is available for borrowing using property accessors.
- Add methods for:
    - Borrowing a book.
    - Returning a book.
    - Listing all books in the library.

5. File Cleanup Utility

Problem Statement:
Develop a utility to organize and clean up files in a directory.

- Create a console application that accepts file extensions as input and moves files to a specified directory.
- Implement a **Destructor** in a helper class to log cleanup activities when the object goes out of scope.
- Use a **Class Library** for reusable file operations like `MoveFile`, `DeleteFile`, and `GetFileInfo`.