Practical Assignment on Properties of Class in C#

1. **Create a class `Student`:**
   - Properties: `Name` (string), `Age` (int), `Marks` (double).
   - Ensure that:
     - `Age` is between 18 and 30.
     - `Marks` cannot be negative.
   - Write a program to input and display student details.
2. **Write a class `Circle`:**
   - Properties: `Radius` (double), `Area` (read-only), and `Circumference` (read-only).
   - The `Radius` property should validate that the value is positive.
   - Compute `Area` and `Circumference` using the radius.
3. **Create a class `Employee`:**
   - Properties: `EmployeeID`, `Name`, and `Salary` (with private set).
   - Provide a method `UpdateSalary(double increment)` to modify the salary.
   - Test the class by creating objects and updating the salary.

4. **Create a class `BankAccount`:**
   - Properties: `AccountNumber` (string), `Balance` (double, read-only).
   - Implement methods `Deposit(double amount)` and `Withdraw(double amount)`.
   - Ensure the `Withdraw` method checks for sufficient balance.
5. **Develop a `Temperature` class:**
   - Properties: `Celsius` (double) and `Fahrenheit` (double).
   - Use property logic such that:
     - Setting `Celsius` updates `Fahrenheit`.
     - Setting `Fahrenheit` updates `Celsius`.
6. **Implement a `Rectangle` class:**
   - Properties: `Length`, `Breadth`, and `Area` (read-only).
   - Restrict `Length` and `Breadth` to positive values only.
   - Test by creating instances of the class and computing the area.

7. **Create a class `LibraryBook`:**
   - Properties: `Title`, `Author`, `IsAvailable` (bool, private set).
   - Add methods `Borrow()` and `Return()`.
   - Ensure that:
     - A book cannot be borrowed if it's already borrowed.
     - It cannot be returned if it wasn’t borrowed.
8. **Design a `Stock` class:**
   - Properties: `Symbol` (string), `Price` (double), `PriceChange` (read-only).
   - Maintain the previous price internally.

- Use property logic to calculate `PriceChange` whenever the `Price` is updated.

9. **Write a `Vehicle` class**:
   - Properties: `Make`, `Model`, `FuelCapacity` (double), and `FuelLevel` (double).
   - Implement logic in `FuelLevel` to ensure it doesn't exceed `FuelCapacity` or go below 0.
10. **Create a class `Person` with property `DateOfBirth`**:
    - Calculate and return the `Age` using the current date.
    - Ensure that the `DateOfBirth` is not a future date.

11. **Develop a `Product` class for an inventory system**:
    - Properties: `ProductName`, `Price`, `StockQuantity` (with validation).
    - Add a computed property `TotalValueInStock` (Price × StockQuantity).
    - Write a program to manage inventory (add products, update stock, calculate total inventory value).