# Content Management System

Create a scenario that models a **Content System** using *interfaces* and *abstraction*.

Guidelines:

- Create an interface defining the basic functionalities of a
  **content item** – should contains **Title** and **Display()** method.

- Then create an abstract class implementing common
  functionality for **content item**s. Add a *constructor* that
  receives *title* as parameter.

- Create two classes – **Article** and **Video** that represent a
  specific content item and implement abstraction already
  defined. Add **Author** property(only getter) for class *Article*
  and **DurationInMinutes** for class *Video*.

- For Content System testing create a class **ContentManager** that
  contains a collection of content **items**(List<?>) with
  possibility to add item by **AddContent(***{item}***)** and displays
  information about each item in the collection by
  **DisplayContent()** method.


At the end code bellow should work properly:

```csharp
class Program
{
    static void Main(string[] args)
    {
        // Creating content items: article and video
        Article article = new Article("Understanding Interfaces", "John Doe");
        Video video = new Video("Introduction to C#", 15);

        // Creating a content manager and adding content items to it
        ContentManager contentManager = new ContentManager();
        contentManager.AddContent(article);
        contentManager.AddContent(video);

        // Displaying content items
        contentManager.DisplayContent();

        Console.ReadKey();
    }
}
```