Music Library System

Create a scenario that models a **Music Library** using ***interfaces*** and ***abstraction***.

Guidelines:

- Create an interface defining the basic functionalities of a **music item** – should contains **Title**, **Artist**, and **Year** properties(only getters) and **Play()** method.

- Then create an abstract class **implementing** common functionality for **music items**. Add a ***constructor*** that receives ***title***, ***artist***, and ***years*** as parameters.

- Create two classes – **Song** and **Soundtrack** that represent a specific music item and implement abstraction already defined. Add **HasCover** property(only getter) for class *Song*.

- For Music System testing create a class **Album** that contains a collection of music **items**(List<?>) with possibility to add item by **AddMusicItem(***{item}***)** and plays all music items in the collection by **Play()** method.

At the end code bellow should work properly:

```
class Program
{
    static void Main(string[] args)
    {
        // Creating music items: song and soundtrack
        Song song = new Song("Live Is Life", "Opus", 1991, true);
        Soundtrack soundtrack = new Soundtrack("Titanic", "James Horner", 1997);

        // Creating an album and adding music items to it
        Album myAlbum = new Album();
        myAlbum.AddMusicItem(song);
        myAlbum.AddMusicItem (soundtrack);

        // Play items in the album
        myAlbum.Play();

        Console.ReadKey();
    }
}
```