

Iteration Lab

Lab 1: ForEach

Open the **Song.cs** file in the \EntityClasses folder.

Add a new property named ***SongArtistAlbum***.

```
public string SongArtistAlbum { get; set; }
```

Open the **Program.cs** file and replace all the code to look like the following:

```
using LINQLab.EntityClasses;
using LINQLab.RepositoryClasses;

// Declare variables and fill data
List<Song> songs = SongRepository.GetAll();

// TODO: Write Your Query Here

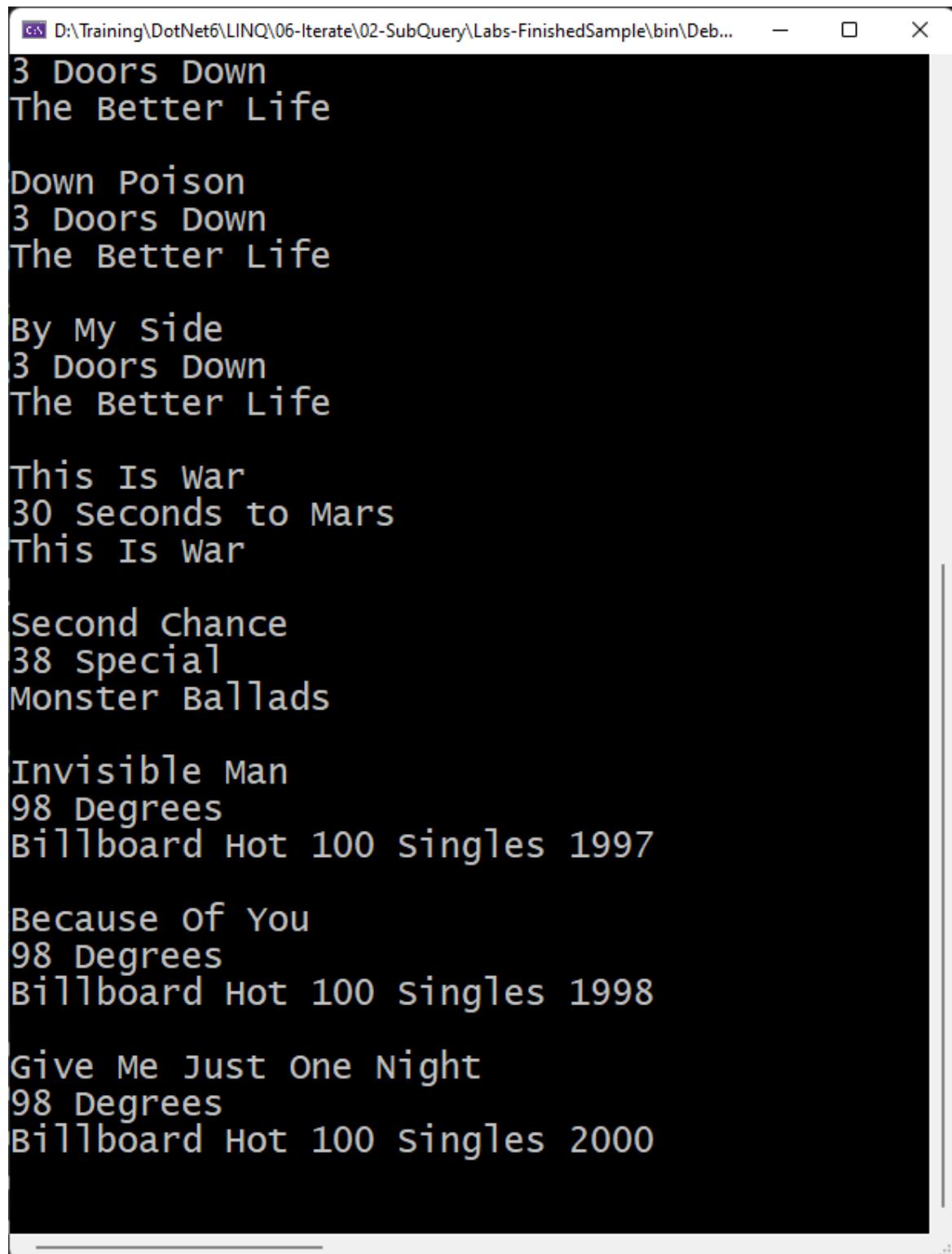
// Display the Results
foreach (var song in songs) {
    // Display the SongArtistAlbum property
    Console.WriteLine(song.SongArtistAlbum);
}

// Pause for Results
Console.ReadKey();
```

Write a query to iterate over the ***songs*** variable and concatenate the ***SongName***, ***Artist***, and ***Album*** properties into the ***SongArtistAlbum*** property delimited by an *Environment.NewLine*.

Try it Out

Run the application and view the results:



```
D:\Training\DotNet6\LINQ\06-Iterate\02-SubQuery\Labs-FinishedSample\bin\Deb...
3 Doors Down
The Better Life

Down Poison
3 Doors Down
The Better Life

By My Side
3 Doors Down
The Better Life

This Is War
30 Seconds to Mars
This Is War

Second Chance
38 special
Monster Ballads

Invisible Man
98 Degrees
Billboard Hot 100 singles 1997

Because of You
98 Degrees
Billboard Hot 100 singles 1998

Give Me Just One Night
98 Degrees
Billboard Hot 100 singles 2000
```

Lab 2: ForEach with Subquery

Open the **Song.cs** file in the \EntityClasses folder.

Add a new property named **GenreName**.

```
public string GenreName { get; set; }
```

Open the **Program.cs** file and add a new variable, named **genres**, that loads the data from the MusicGenreRepository.

```
List<MusicGenre> genres = MusicGenreRepository.GetAll();
```

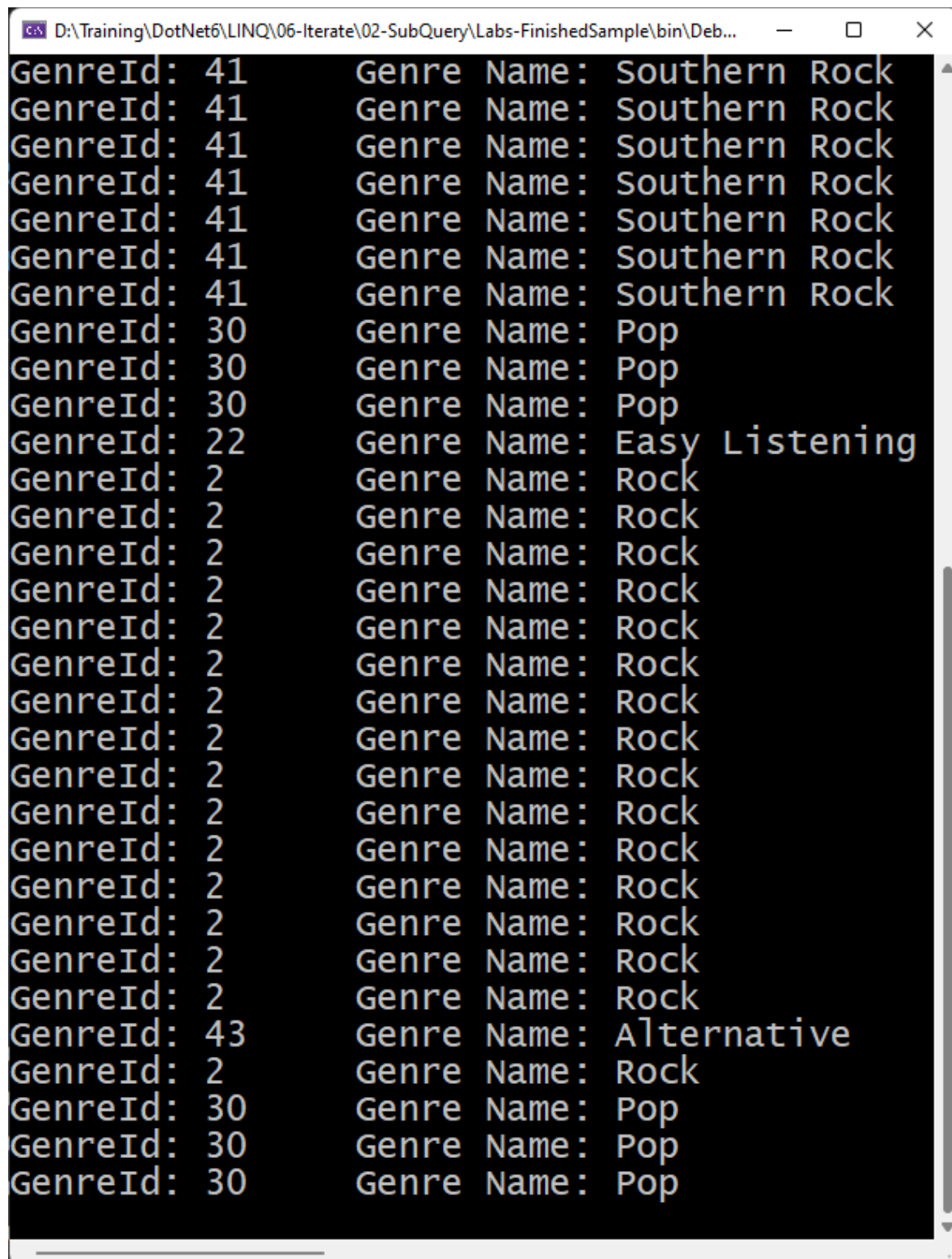
Write a query that sets the **GenreName** property for each song by looking up the **GenreId** in the **genres** variable.

Replace the existing loop with the one shown below to display the **GenreId** and **GenreName** properties from the **songs** variable

```
// Display the Results
foreach (var song in songs) {
    // Display the Genre Information
    Console.WriteLine($"GenreId: {song.GenreId}\tGenre Name:
{song.GenreName}");
}
```

Try it Out

Run the application and view the results:



A screenshot of a console window with a black background and white text. The window title bar shows the path 'D:\Training\DotNet6\LINQ\06-Iterate\02-SubQuery\Labs-FinishedSample\bin\Deb...'. The console displays a list of 28 rows, each containing a 'GenreId' and a 'Genre Name'. The data is as follows:

GenreId	Genre Name
41	Southern Rock
41	Southern Rock
41	Southern Rock
41	Southern Rock
41	Southern Rock
41	Southern Rock
41	Southern Rock
30	Pop
30	Pop
30	Pop
22	Easy Listening
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
2	Rock
43	Alternative
2	Rock
30	Pop
30	Pop
30	Pop