Adding objects to a List<...>

Stack

(local variables)

Heap

(objects / class instances)

```
public class Song
{
    public int Ranking, Year;
    public string Title, Artist;
}
```

Stack

(local variables)

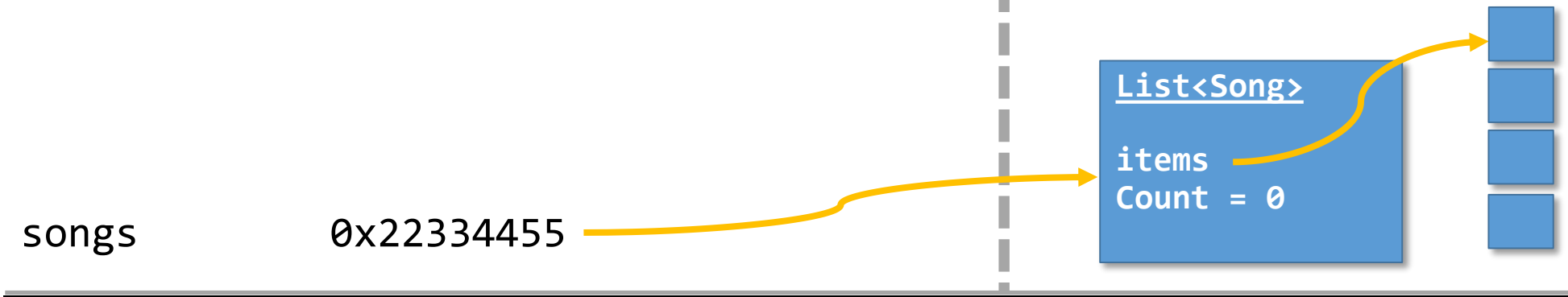
Heap

(objects / class instances)

```
List<Song> ReadSongs(string filename)
{
    List<Song> songs = new List<Song>();
    StreamReader reader = new StreamReader(filename);

    // read Song objects and add them to the list...

    reader.Close();
    return songs;

}
```

Stack
(local variables)

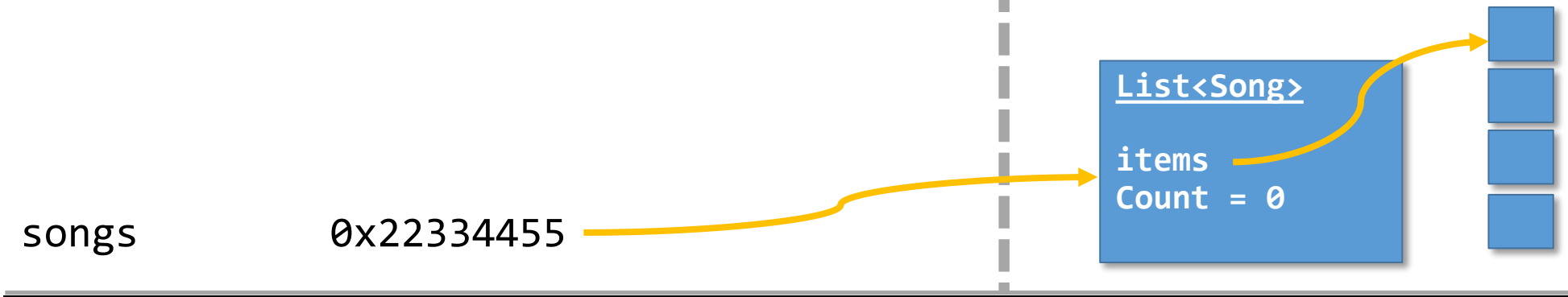
Heap
(objects / class instances)




```
List<Song> ReadSongs(string filename)
{
    List<Song> songs = new List<Song>();
    StreamReader reader = new StreamReader(filename);

    // read Song objects and add them to the list...

    reader.Close();
    return songs;

}
```

Stack
(local variables)

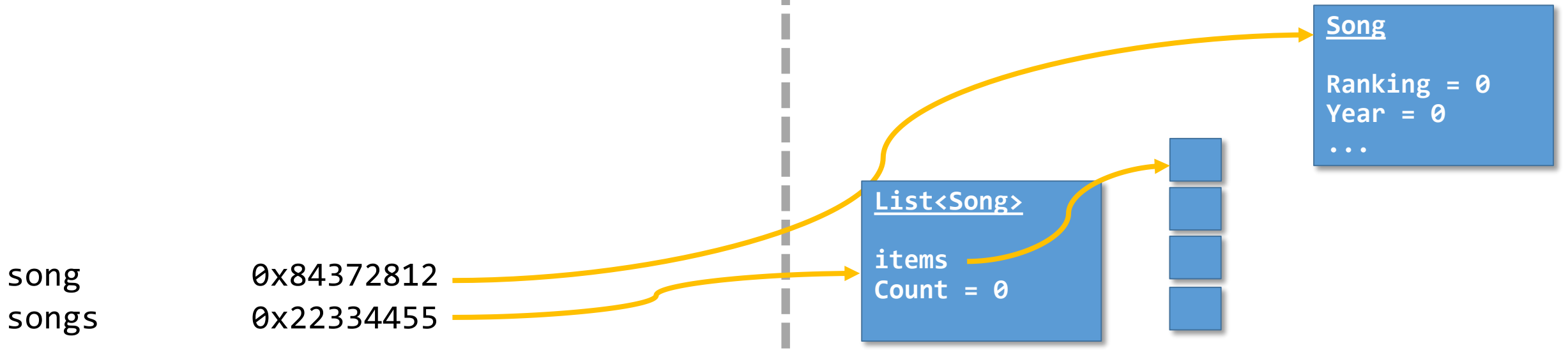
Heap
(objects / class instances)




```
Song song = new Song();
while (!reader.EndOfStream)
{
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);

  songs.Add(song);
}
```

Stack
(local variables)

Heap
(objects / class instances)

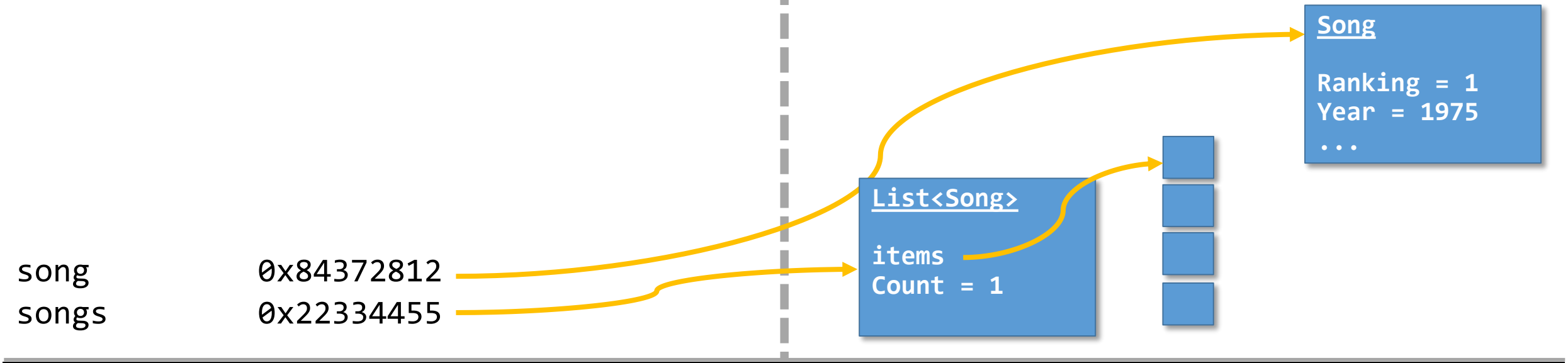



```
Song song = new Song();
while (!reader.EndOfStream)
{
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);

  songs.Add(song);
}
```

A new Song object is created. It can be filled and added to the songs-list.

Stack
(local variables)

Heap
(objects / class instances)

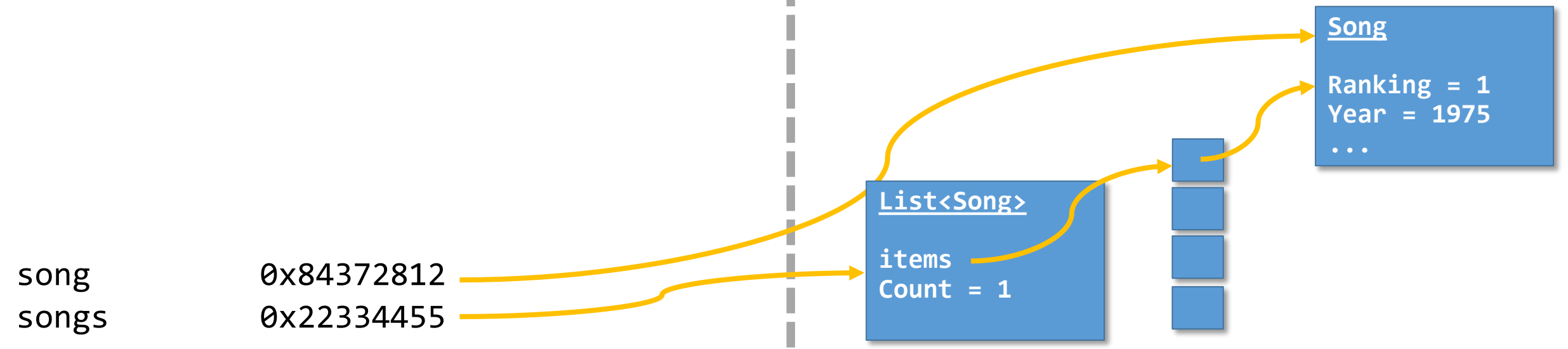



```
Song song = new Song();
while (!reader.EndOfStream)
{
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);

  songs.Add(song);
}
```

The Song is filled with information, read from the file.

Stack
(local variables)

Heap
(objects / class instances)



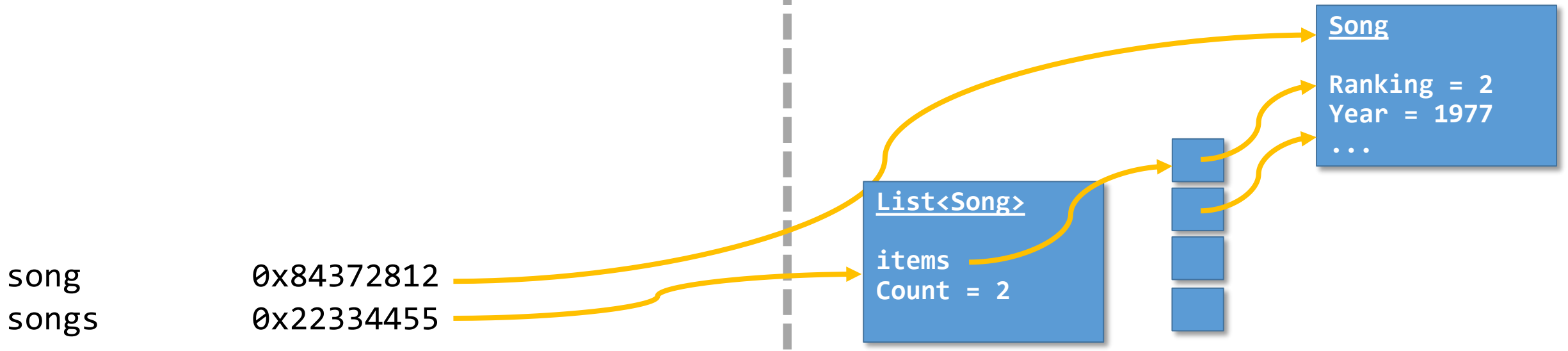

```
Song song = new Song();
while (!reader.EndOfStream)
{
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);

  songs.Add(song);
}
```

1st time

Now the Song object is added to the songs-list.

Stack
(local variables)

Heap
(objects / class instances)



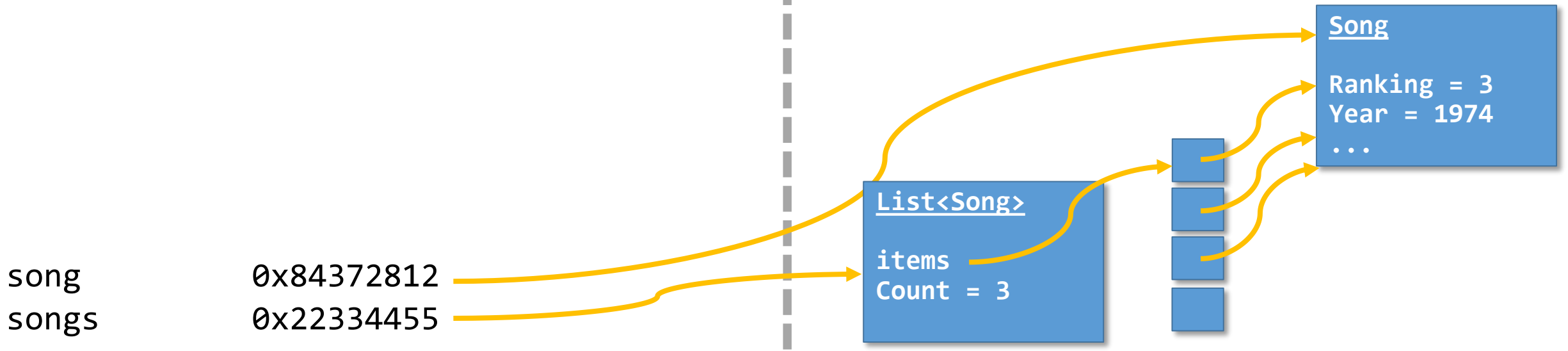

```
Song song = new Song();
while (!reader.EndOfStream)
{
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);

  songs.Add(song);
}
```

2nd time

In the 2nd iteration of the loop, the same Song object is changed and added to the songs-list.

Stack
(local variables)

Heap
(objects / class instances)




```
Song song = new Song();
while (!reader.EndOfStream)
{
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);
  songs.Add(song);
}
```

3rd time

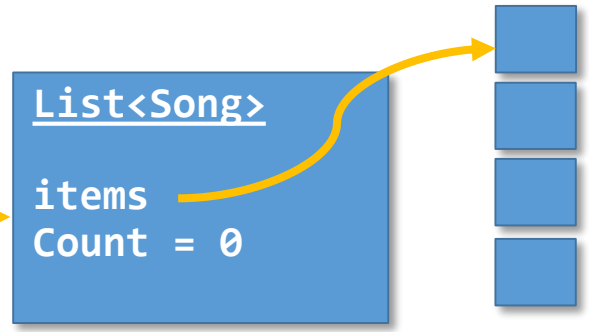
In the 3rd iteration of the loop, the same Song object is changed and added to the songs-list.

As can be seen, the 3 list-entries all reference the same Song object!! Not good!

Stack
(local variables)

Heap
(objects / class instances)

songs 0x22334455

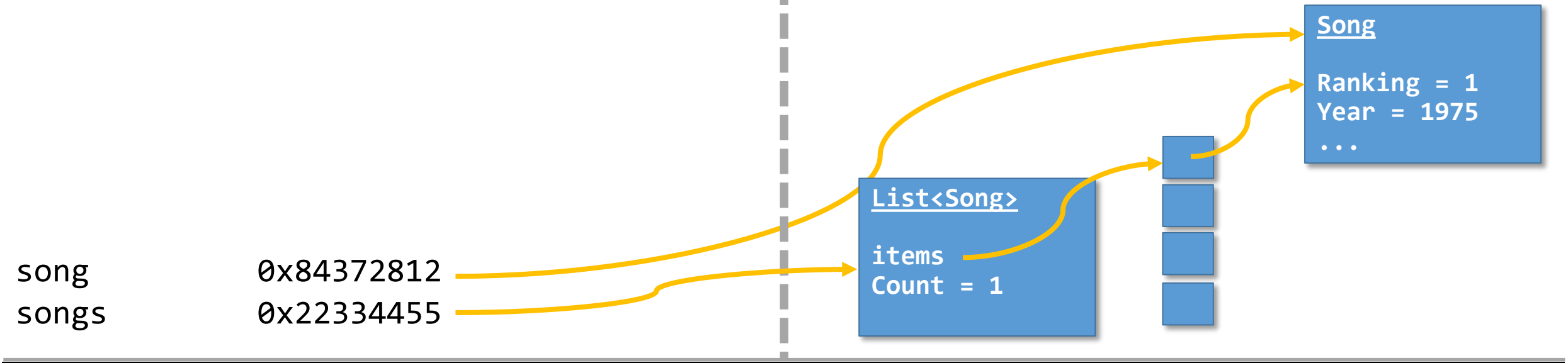



```
while (!reader.EndOfStream)
{
  Song song = new Song();
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);

  songs.Add(song);
}
```

Now, the Song object is created inside the loop. Let's see what happens now.

Stack
(local variables)

Heap
(objects / class instances)



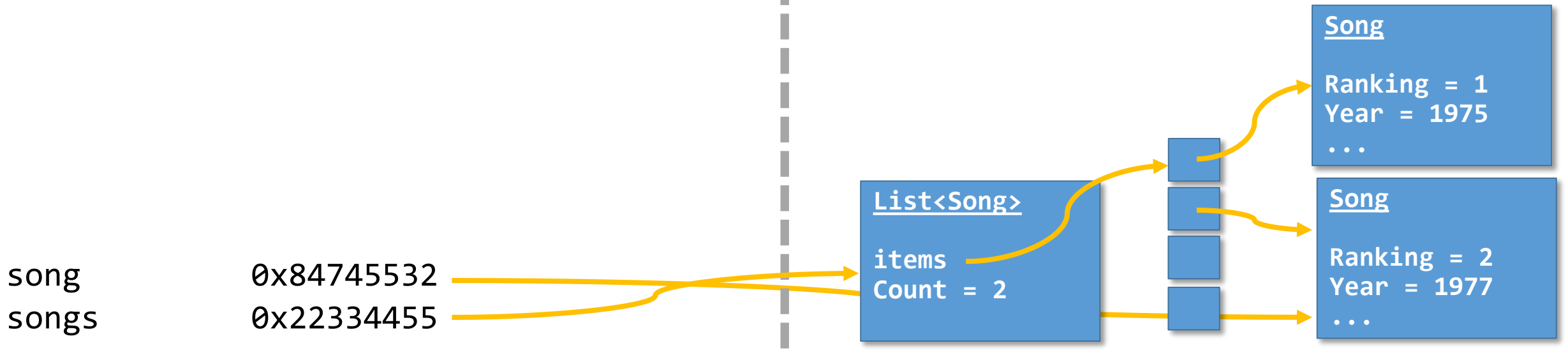

```
while (!reader.EndOfStream)
{
  Song song = new Song();
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);

  songs.Add(song);
}
```

1st time

The first Song object is added to the songs-list.

Stack
(local variables)

Heap
(objects / class instances)



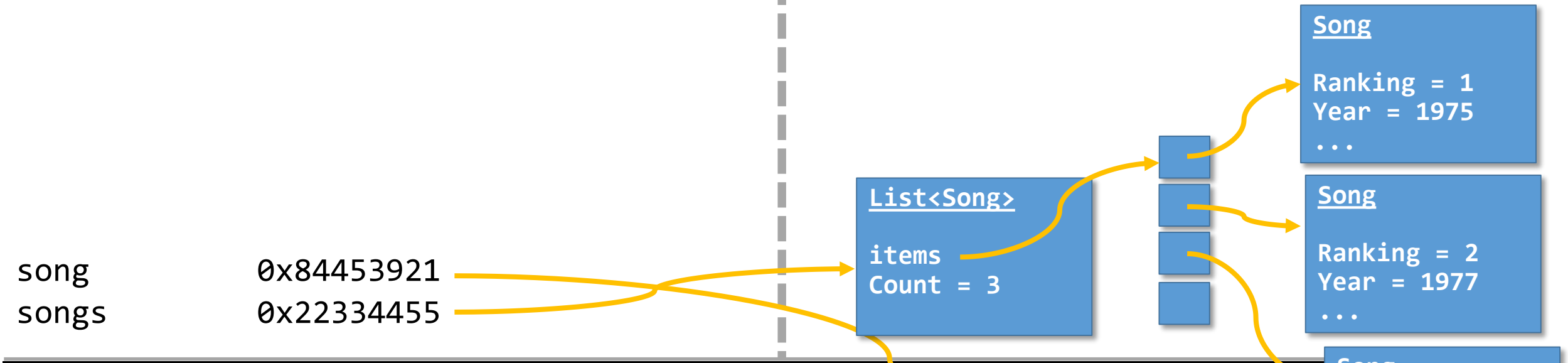

```
while (!reader.EndOfStream)
{
  Song song = new Song();
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);

  songs.Add(song);
}
```

2nd time

In the 2nd iteration of the loop, a new Song object is added to the songs-list.

Stack
(local variables)

Heap
(objects / class instances)




```
while (!reader.EndOfStream)
{
  Song song = new Song();
  line = reader.ReadLine();
  string[] fields = line.Split(';');
  song.Ranking = int.Parse(fields[0]);
  song.Title = fields[1];
  song.Artist = fields[2];
  song.Year = int.Parse(fields[3]);
  songs.Add(song);
}
```

3rd time

In the 3rd iteration of the loop, another new Song object is added to the songs-list.

Since the Song objects are now created inside the loop, the list will contain different objects! Good!