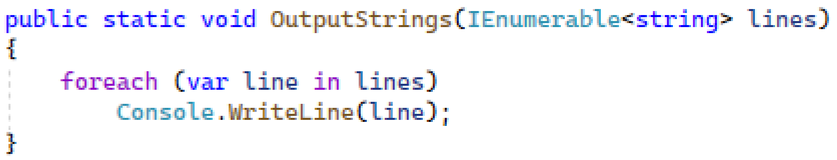
Lab 4

CS321L Winter 2023, Professor Christopher Diggins

Write this function and two other versions. Once using a while loop, and another using a for loop. Don't use an integer index. Instead use the enumerator object returned from "GetEnumerator()". Read the documentation at: https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.ienumerable-1.getenumerator?view=net-7.0



```
public static void OutputStrings(IEnumerable<string> lines)
{
    foreach (var line in lines)
        Console.WriteLine(line);
}
```

Write a unit-test for each function that tests the function on each of the three inputs:

OutputStrings(new[] { "apples", "bananas", "grapes" });

OutputStrings(new List<string>() { "planes", "trains", "cars" });

OutputStrings(MyGenerator());

The function MyGenerator() should look like this:

```
IEnumerable<string> MyGenerator() {
  yield return "my";
  yield return "dog";
  yield return "loves";
  yield return "bananas";
}
```

The functions and properties you need are:

- IEnumerator<T> IEnumerable<T>.GetEnumerator()
- bool IEnumerator<T>.MoveNext();
- T IEnumerator<T>,CurrentValue;

Remember that calling the CurrentValue property before MoveNext() has been called at least once, is invalid and will always cause an error.

Submission:

- The source file
- Screenshot of the test output

Grading

- **3 points**