CSV File Kata – Requirement 1

Rules

1. Strictly practice TDD: Red, Green, and Refactor.
2. Use SOLID principles.
3. Use provided interfaces and classes.

Given the following:

```
public interface IFileSystem
{
    void WriteLine(string fileName, string line);
}

public class Customer
{
    public string Name { get; set; }
    public string ContactNumber { get; set; }

    public override string ToString()
    {
        return string.Join(",", Name, ContactNumber);
    }
}
```

Install the NuGet package **CsvFile.Kata.Dependencies** to include the above code.

Implement the following:

Your team has come to you with an urgent requirement. They need to dump Customer objects, as per the Customer class, to disk in CSV format for a nightly job that imports the data into the CRM system for sales. You do not need to worry about writing headers as the CRM system has NO need for headers. You do NOT you need to worry about catering for commas in your data. You do NOT need to worry about checking for existing files or data as the customers will always be dumped to a clean folder. You MUST NOT write to actual disk, use the IFileSystem interface provided and mock an implementation when building your unit tests.

Hint

You should end up with a single public method after completing this requirement. Use the one-to-many green bar pattern to move from handling a single customer to a collection of customers mutating your single public method as you progress through the stages of the pattern.