# Exercise: Streams and Files

This document defines in-class exercise problems from the [“Advanced C#“ Course @ Software University](http://softuni.bg/courses/advanced-csharp/). You are presented with some problems and certain steps you need to take in order to accomplish the tasks.

## Clean Up the Mess

As an aspiring programmer you were given some code to refactor. But, oh my, is it messed up! Take a look at it (the file is called "Mecanismo.cs", you can find it in the homework archive). Before you start any refactoring, you first need to make it readable at least to some extent. It will be a pain to do this by hand and you’re in a hurry. The solution is to write a program to do some of the work for you (after that it’s all Ctrl + K + D in Visual Studio). Thankfully, you now know how to work with files and you also have some knowledge of regular expressions (yes, working with text means knowing regex).

We’ve outlined several steps to clean up the file:

* Read the text from the file and store it in a string
* Replace all sequences of the pattern *"whitespace (zero or more) – new line – whitespace (zero or more)"* with a new line (prefer using **Environment.NewLine**)
* Separate all statements. This means replacing all occurrences of *';' + whitespace (zero or more)* with ';' followed by a new line. Now every code statement will be on its own line.
* Follow the best practices of quality code in C#, which means all opening and closing curly brackets should be on separate lines. Replace all occurrences of *"whitespace (zero or more) – curly bracket – whitespace (zero or more)"* with the same bracket surrounded by new lines.
* Dots are used to access members or methods, they shouldn’t be surrounded by spaces, so remove all spaces around the dots in the text.
* We can do even more, but let’s leave that to Visual Studio.
* Save the resulting text in a new file. You’ve gathered that the file is some sort of Engine for a program, so save it in the same directory under the name "Engine.cs".

The result should look something like this:

|  |
| --- |
| using System;  using Comandos;  namespace VehicleParkSystem2  {  class Mecanismo : IMecanismo  {  private exec ex;  Mecanismo ( exec ex )  {  this.ex = ex;  } |