**Assignment 1**

In this assignment you will implement a Java or C# application to analyse sales data from a file, and get various statistics from it.

**Familiarization**

Take a look at SalesRecords.csv. This is a comma-separated-value (CSV) file that contains random sales data for a hypothetical company. Open this file in a text editor to familiarize yourself with its structure. Note the following points:

* The first line contains "column headings", which explain the meaning of each field
* Each of the remaining lines represents a separate sale

**Requirements**

Your task is to implement a C# or Java application to load this file into memory and calculate various statistical information. There are some important points you should bear in mind:

* Your application should receive the name of the data file as a command-line argument. If the command line doesn't specify a file name, or if the data file cannot be opened for whatever reason, terminate the program immediately with a suitable error message.
* You must implement CSV-file-parsing logic to read all the records from the data file into memory.
* The current data file is CSV, but in future your application might need to support other data file formats (e.g. JSON, XML, fixed-record format, etc.). Therefore, your application must be written in a "pluggable" fashion so that other types of file-parser logic might be implemented in the application in future. Hint: think "interfaces"…
* The application must offer various statistical calculations. It's up to you to decide what kinds of statistical analyses are appropriate. Your application should be flexible enough such that other kinds of analyses could be implemented and plugged into the application in future.
* Implement some kind of simple menu-driven approach to enable the user to decide what statistical information to display on the screen.
* Also implement a mechanism that allows the user to export summary information to a report document. The summary information might contain max/min costs, total revenue, average units sold, etc. It's up to you (or even maybe the user?) to decide what summary information to generate in the output file. For added credit, you might also allow the user to choose the output file format (e.g. CSV, JSON, etc.)

**Additional information**

* The Java and C# course materials contain chapters on file handling, so have a look at these chapters and demos to help you get started.
* You might also want to investigate any third-party libraries (i.e. NuGet packages in C#, and JAR files in Java) to help parse certain file formats.
* Use a GitHub repository to share work with your other team member(s).
* Remember to test your code as thoroughly as possible in the time available.
* Strive to adopt good OO practices, making appropriate use of classes, collections, inheritance, interfaces, etc.

**Presentations**

On Friday 26 February you'll be invited to present your solution to the other teams. Aim for about 20 minutes of talking plus 5 minutes of Q&A. During your presentation:

* Explain your key design choices (e.g. uses of interfaces, inheritance, classes, etc.)
* Show the code organisation in GitHub
* Show as much code and tests as you think appropriate for a technical audience
* Run the application to demonstrate it actually works (!)
* Discuss how you might extend the application in future to make it better, more flexible, more robust, etc. Also point out any limitations and/or simplifications you might have made (and why)