

Exercises

November 14th 2017

This exercise sheet can be downloaded as [PDF](#)

Exercise 1

Make a UTF-8 text file with some words in Danish. Be sure to use plenty of special Danish characters. You may consider to write a simple C# program to create the file. You may also create the text file in another way.

In this exercise you should avoid writing a byte order mark (BOM) in your UTF-8 text file. (A BOM in the UTF-8 text file may short circuit the decoding we are asking for later in the exercise). One way to avoid the BOM is to denote the UTF-8 encoding with new UTF8Encoding(), or equivalently new UTF8Encoding(false). You may want to consult the constructors in class UTF8Encoding for more information.

Now write a C# program which systematically - in a loop - reads the text file six times with the following objects of type Encoding: ISO-8859-1, UTF-7, UTF-8, UTF-16 (Unicode), UTF32, and 7 bits ASCII.

More concretely, I suggest you make a list of six encoding objects. For each encoding, open a TextReader and read the entire file (with ReadToEnd, for instance) with the current encoding. Echo the characters, which you read, to standard output.

You should be able to recognize the correct, matching encoding (UTF-8) when you see it.

Exercise 2

Write a program that tosses a Die 1000 times, and writes the outcome of the tosses to a textfile. Use a TextWriter to accomplish the task.

Write another program that reads the text file. Report the number of ones, twos, threes, fours, fives, and sixes.

Exercise 3

From the previous lecture, we used StreamWriter.cs file, now modify it to use a binary writer instead of a StreamWriter.

Exercise 4

Go on to [codecademy.com](https://www.codecademy.com)

Create an account

Use some time to look at [Learn HTML](#) and [Learn CSS](#)

This knowledge will be required for making websites 😊