2024-11-19

# Files

[Solutions for those exercises.](https:/princomp.github.io/solutions/io/files)

## Questions

1. What is the name of the class we use to read from files?
   * FileReader
   * StreamOpener
   * ReadFile
   * StreamReader
   * FileStream
2. What is of crucial importance?
   * Always call the Open method before reading from or writing into a file.
   * Always assume the user has C:\ as their main drive.
   * Always call the Close method after being done reading from or writing into a file.
   * Never read from or write into a file inside a try…catch block.

## Problems

1. Write a program that create a text file called HelloWorld.txt in its bin/Debug folder and store “Hello” followed by a name entered by the user in it.
2. Write a program that ask the user for a filename, makes sure the filename ends with “.txt” and does not begin with a “.” (otherwise, the file would be hidden on unix systems), does not match a file with the same name in the bin/Debug folder of your program, then create it in the bin/Debug folder of your program and write in it all the number from 1 to 1,000,000 (both included). Out of curiosity, what is the file size?
3. Execute the following program, then write a program that find the greatest number in the RandomNumber.txt file.

* string filePath = Path.Combine(  
   AppDomain.CurrentDomain.BaseDirectory,  
   "RandomNumbers.txt"  
   );  
   Random gen = new Random();  
   try  
   {  
   StreamWriter sw = new StreamWriter(filePath);  
   for (int i = 1; i <= 100; i++)  
   sw.WriteLine(gen.Next(1000));  
   sw.Close();  
   }  
   catch (Exception e)  
   {  
   Console.WriteLine("Exception: " + e.Message);  
   }

1. Suppose that at filePath is located a file where each line is either
   * a decimal (e.g., 12.4, -14, 0.34),
   * the word “STOP”,
   * some other string (“Test”, “The sky is blue”, “Ignore this line”, “My file contains”), that may contain the characters “STOP”.

* Write a program that displays the average of the decimals in the file knowing that
  + your program should ignore the values after a line containing “STOP” and only “STOP” if it is present,
  + all the other strings should simply be ignored.
* For example, for the following three files, “4.0”, “10.0” and “7.5” should be displayed, as (12.48 - 2.48 + 2) / 3 = 4 (13 been ignored), (15 + 5) / 2 = 10, and (11 + 4) / 2 = 7.5 (12 being ignored).
* ┌────────────────┐  
  │12.48 │  
  │This is a test │   
  │-2.48 │  
  │2 │  
  │STOP │  
  │13 │  
  └────────────────┘  
    
  ┌────────────────┐  
  │My file contains│  
  │STOP but │  
  │averages │  
  │15 │  
  │ and │  
  │5 │   
  └────────────────┘  
    
  ┌────────────────┐  
  │This 12 will be │  
  │ignored │  
  │but not │  
  │11 │  
  │ nor │  
  │4 │   
  └────────────────┘

1. Write a program that asks the user to enter a sentence, and store it in a file *where the maximum width is 40*: if the string entered is more than 40 characters long, then it should span over multiple lines of no more than 40 characters each. For example, if the user enters

* In publishing and graphic design, Lorem ipsum is a placeholder text commonly used to demonstrate the visual form of a document or a typeface without relying on meaningful content. Lorem ipsum may be used as a placeholder before the final copy is available.
* then the text file should contain
* In publishing and graphic design, Lorem   
  ipsum is a placeholder text commonly use  
  d to demonstrate the visual form of a do  
  cument or a typeface without relying on   
  meaningful content. Lorem ipsum may be u  
  sed as a placeholder before the final co  
  py is available.

1. Write a program that counts the number of words in itself! Ideally, empty lines should not count toward the word count.

* Hint: Program.cs is normally located at
* Path.Combine(  
   new DirectoryInfo(Directory.GetCurrentDirectory()).Parent.Parent.ToString(),  
   "Program.cs"  
  )