C# Fundamentals

5. **File I/O and Exception HandlingObjective:Requirements:**
    - **Develop an application that reads from and writes to files.**
    - **Read text from a file (e.g., a log file or a simple CSV).**
    - **Process the data (for example, count words or lines).**
    - **Write the result to a new file.**
    - **Implement exception handling to manage file-related errors (such as FileNotFoundException or IOException).**

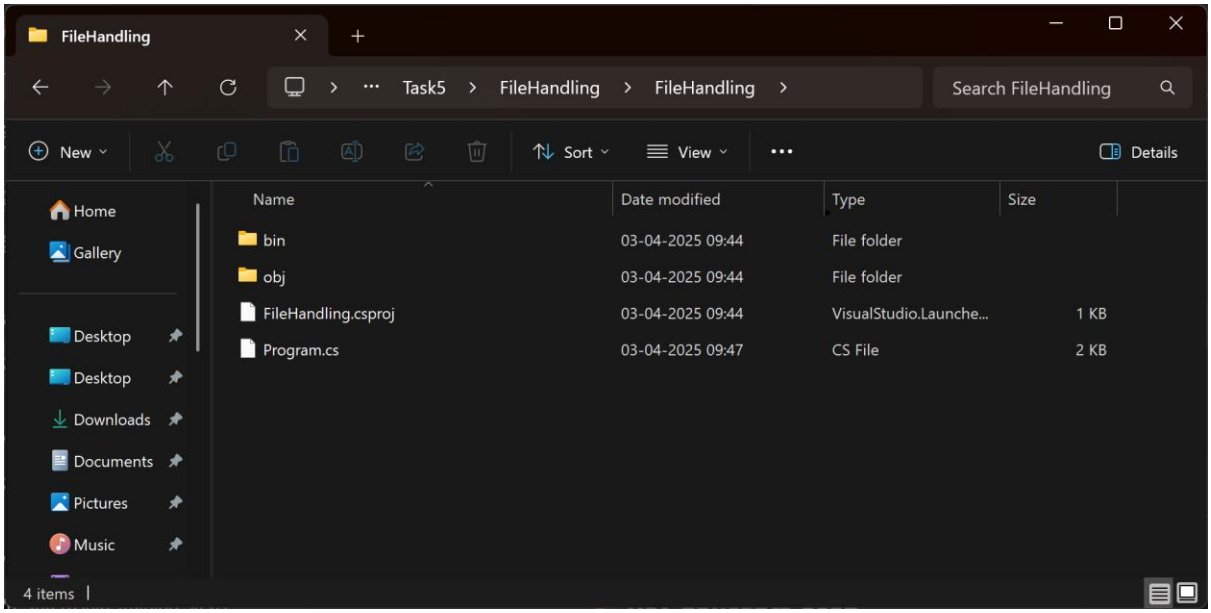
Program code:

```
using System;
using System.IO;

class FileIOExample
{
   static void Main()
   {
      string inputFile = "input.txt";  // Input file name
      string outputFile = "output.txt"; // Output file name

      try
      {
         // Check if input file exists
         if (!File.Exists(inputFile))
         {
            Console.WriteLine($"Error: The file '{inputFile}' does not exist.");
            return;
         }

         // Read all lines from the file
```

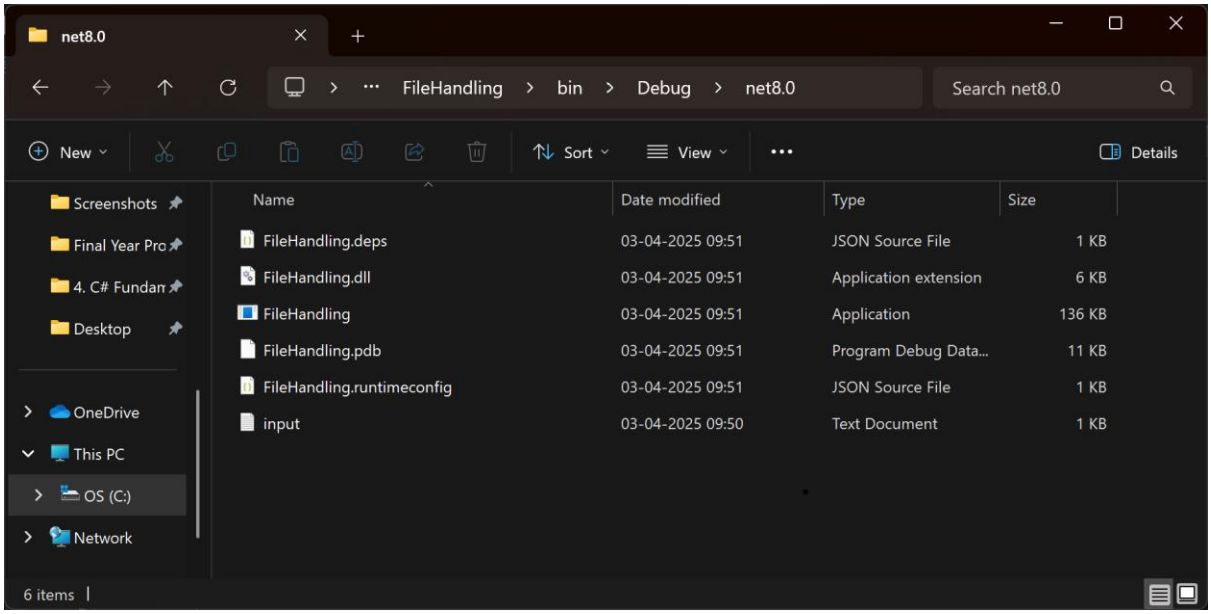
```
        string[] lines = File.ReadAllLines(inputFile);
        int lineCount = lines.Length;

        // Count words in the file
        int wordCount = 0;
        foreach (string line in lines)
        {
            // Splits each line into words using spaces/tabs and counts them.
            wordCount += line.Split(new char[] { ' ', '\t' },
StringSplitOptions.RemoveEmptyEntries).Length;
        }

        // Display results
        Console.WriteLine($"Total Lines: {lineCount}");
        Console.WriteLine($"Total Words: {wordCount}");

        // Write results to output file
        File.WriteAllText(outputFile, $"Total Lines: {lineCount}\nTotal Words: {wordCount}");

        Console.WriteLine($"Results saved in '{outputFile}'.");
    }
    catch (FileNotFoundException)
    {
        Console.WriteLine($"Error: File '{inputFile}' not found.");
    }
    catch (IOException ex)
    {
        Console.WriteLine($"I/O Error: {ex.Message}");
    }
    catch (Exception ex)
    {
```

```
            Console.WriteLine($"Unexpected Error: {ex.Message}");
        }
    }
}
```

Output:



input.txt is in the same folder as the .exe file (inside bin\Debug\net8.0).



Web_Changes.txt | Leveraged Pa | Gmail-Linkedin-Git | Deepu#@20(| city electroni | - hosts localh | input.txt.txt

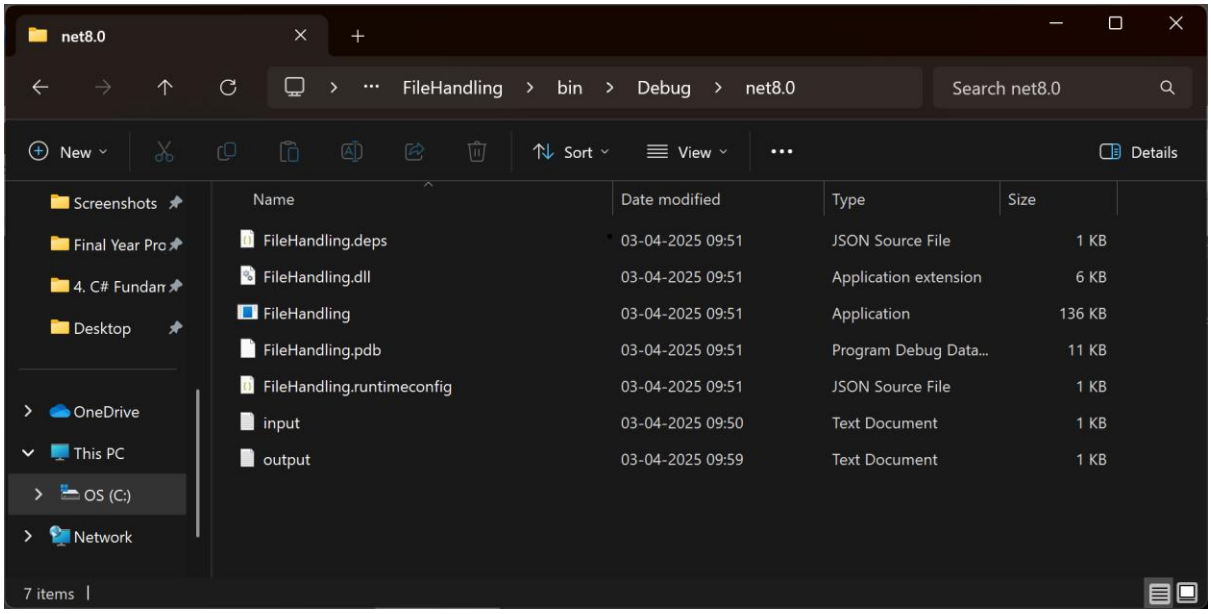
File Edit View

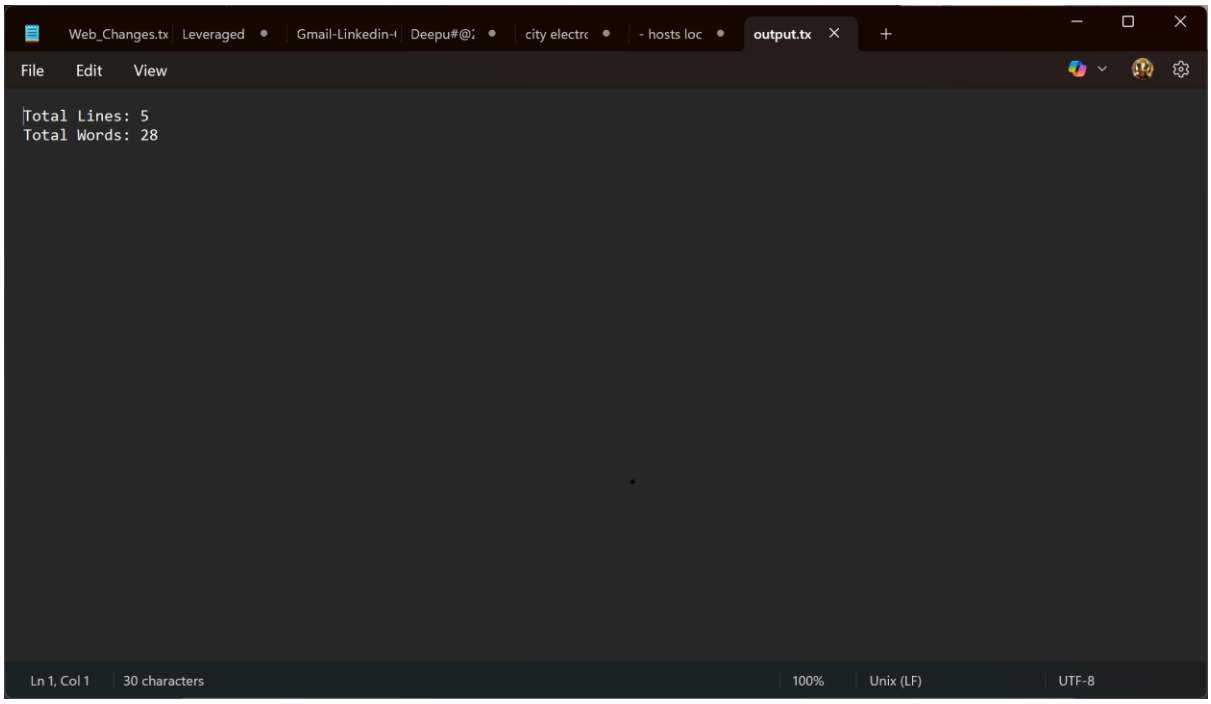
```
Hello, this is a sample text file.
It contains multiple lines.
Each line has some words.
C# is great for file handling.
Let's count the words and lines!
```

Ln 6, Col 1 | 153 characters | 100% | Windows (CRLF) | UTF-8

Microsoft Visual Studio Debu

```
Total Lines: 5
Total Words: 28
Results saved in 'output.txt'.

C:\Rohith\Backup\Desktop\Presidio\Pre-Training\4. C# Fundamentals\Task5\FileHandling\FileHandling\bin\Debug\net8.0\FileH
andling.exe (process 23312) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```


net8.0
FileHandling > bin > Debug > net8.0
Search net8.0
New
Sort
View
Details
Screenshots
Final Year Pro
4. C# Fundam
Desktop
OneDrive
This PC
OS (C:)
Network
Name
Date modified
Type
Size
FileHandling.deps 03-04-2025 09:51 JSON Source File 1 KB
FileHandling.dll 03-04-2025 09:51 Application extension 6 KB
FileHandling 03-04-2025 09:51 Application 136 KB
FileHandling.pdb 03-04-2025 09:51 Program Debug Data... 11 KB
FileHandling.runtimeconfig 03-04-2025 09:51 JSON Source File 1 KB
input 03-04-2025 09:50 Text Document 1 KB
output 03-04-2025 09:59 Text Document 1 KB
7 items


Web_Changes.tx
Leveraged
Gmail-Linkedin-
Deepu#@
city electrc
- hosts loc
output.tx
File
Edit
View
Total Lines: 5
Total Words: 28
Ln 1, Col 1
30 characters
100%
Unix (LF)
UTF-8