**File Handling**

Program 1 :  
  
  
using System;

using System.IO;

namespace \_05Files\_Reading\_DSPS1

{

internal class Program

{

static void Main(string[] args)

{

// Reading the entire content of the file and printing it

StreamReader input = File.OpenText("Rapunzel.txt");

string text = input.ReadToEnd();

input.Close();

Console.WriteLine("=== Entire File Content ===");

Console.WriteLine(text);

Console.WriteLine();

// Reading and printing the file line by line using a while loop

input = File.OpenText("Rapunzel.txt");

Console.WriteLine("=== Line by Line (while loop) ===");

while (!input.EndOfStream)

{

Console.WriteLine("--> " + input.ReadLine());

}

input.Close();

Console.WriteLine();

// Reading and printing the file line by line using a foreach loop

Console.WriteLine("=== Line by Line (foreach loop) ===");

foreach (string item in File.ReadLines("Rapunzel.txt"))

{

Console.WriteLine("--> " + item);

}

Console.WriteLine();

// Reading and printing the file character by character using a while loop

input = File.OpenText("Rapunzel.txt");

Console.WriteLine("=== Character by Character (while loop) ===");

while (!input.EndOfStream)

{

Console.Write((char)input.Read() + " ");

}

input.Close();

Console.WriteLine();

Console.WriteLine();

// Reading and printing the file character by character using a foreach loop

Console.WriteLine("=== Character by Character (foreach loop) ===");

foreach (char c in File.ReadAllText("Rapunzel.txt"))

{

Console.Write(c + "\*");

}

Console.WriteLine();

}

}

}

Program 2 : create in bin folder and other specific folder , append and delete

using System;

using System.IO;

namespace \_05FilesIO\_DSPS2

{

internal class Program

{

static void Main(string[] args)

{

// Create and write to a text file in the bin folder

StreamWriter sr = File.CreateText("file.txt");

sr.WriteLine("This is a text file"); // Writing text to the file and opening a new line

sr.Write("Hello"); // Writing text to file, no new line

sr.Write(" My name is Jean-Baptiste!");

sr.WriteLine(" My name is Dinesh Pokhrel!");

sr.Close();

// Create a file in a different location

string customPath = @"C:\Users\dines\Desktop\ichchha\file.txt";

sr = File.CreateText(customPath);

sr.WriteLine("Test file in a different location.");

sr.Close();

// Create a file on the desktop

string desktopFolder = Environment.GetFolderPath(Environment.SpecialFolder.Desktop); // Get the Desktop folder path

string desktopPath = Path.Combine(desktopFolder, "file.txt"); // Combine the path with the filename

sr = File.CreateText(desktopPath);

sr.WriteLine("Hello from the Desktop file.");

sr.Close();

// Append text to the Desktop file if it exists

if (File.Exists(desktopPath))

{

sr = File.AppendText(desktopPath); // Open the file for appending

sr.WriteLine("Testing more text to add.");

sr.WriteLine("Even more text.");

sr.Close();

}

// Append text to the custom path file if it exists

if (File.Exists(customPath))

{

sr = File.AppendText(customPath); // Open the file for appending

sr.WriteLine("Hello from the custom path file.");

sr.Close();

}

// Create and delete a file in the Music folder

string musicFolder = Environment.GetFolderPath(Environment.SpecialFolder.MyMusic);

string musicFilePath = Path.Combine(musicFolder, "WEIRDFILE.txt");

sr = File.CreateText(musicFilePath);

sr.Close();

// Delete the file in the Music folder if it exists

if (File.Exists(musicFilePath))

{

File.Delete(musicFilePath);

Console.WriteLine("File deleted: " + musicFilePath);

}

}

}

}

Program 3 :  
  
using System;

using System.IO;

namespace HarryPotterLineExtractor

{

internal class Program

{

static void Main(string[] args)

{

// Count the total number of lines in the file

int totalLines = CountLines("Harry Potter and the Sorcerer.txt");

Console.WriteLine($"Total number of lines in the file: {totalLines}");

// Ask the user for the number of lines to copy

Console.Write("Enter the number of lines to copy: ");

if (!int.TryParse(Console.ReadLine(), out int linesToCopy) || linesToCopy <= 0)

{

Console.WriteLine("Invalid number of lines.");

return;

}

// Copy the specified number of lines to new files using two different methods

CopyLinesToFile("Harry Potter and the Sorcerer.txt", $"harry-{linesToCopy}.txt", linesToCopy);

CopyLinesToFileUsingStreams("Harry Potter and the Sorcerer.txt", $"harry-{linesToCopy}-option2.txt", linesToCopy);

}

// Method to count the number of lines in a file

static int CountLines(string filePath)

{

int count = 0;

foreach (string line in File.ReadLines(filePath))

{

count++;

}

return count;

}

// Method to copy a specified number of lines to a new file

static void CopyLinesToFile(string inputFilePath, string outputFilePath, int numberOfLines)

{

using (StreamWriter writer = File.CreateText(outputFilePath))

{

int count = 0;

foreach (string line in File.ReadLines(inputFilePath))

{

if (count < numberOfLines)

{

writer.WriteLine(line);

count++;

}

else

{

break;

}

}

}

Console.WriteLine($"Created {outputFilePath} with {numberOfLines} lines.");

}

// Alternative method to copy a specified number of lines to a new file using StreamReader and StreamWriter

static void CopyLinesToFileUsingStreams(string inputFilePath, string outputFilePath, int numberOfLines)

{

using (StreamReader reader = File.OpenText(inputFilePath))

using (StreamWriter writer = File.CreateText(outputFilePath))

{

int count = 0;

while (count < numberOfLines && reader.ReadLine() is string line)

{

writer.WriteLine(line);

count++;

}

}

Console.WriteLine($"Created {outputFilePath} with {numberOfLines} lines using StreamReader/Writer.");

}

}

}

Program 4:

using System;

using System.IO;

using System.Text.RegularExpressions;

namespace HarryPotterCharacterExtractor

{

internal class Program

{

static void Main(string[] args)

{

// Part 1: Extracting and printing names of characters from Gryffindor

ExtractGryffindorCharacters("Characters.csv");

// Part 2: Extracting and printing all birth years from the file

ExtractBirthYears("Characters.csv");

}

static void ExtractGryffindorCharacters(string filePath)

{

// Read through the file line by line

foreach (var line in File.ReadLines(filePath))

{

// Check if the line contains "Gryffindor"

if (line.Contains("Gryffindor"))

{

// Find the index of the first semicolon to determine where the first field ends

int firstSemicolonIndex = line.IndexOf(';');

// Extract a substring of the line starting from the character after the first semicolon

string afterFirstSemicolon = line.Substring(firstSemicolonIndex + 1);

// Find the index of the second semicolon inside the substring

int secondSemicolonIndex = afterFirstSemicolon.IndexOf(';');

// Extract and print the substring that starts from the beginning of the substring and ends just before the second semicolon

string characterName = afterFirstSemicolon.Substring(0, secondSemicolonIndex);

Console.WriteLine(characterName);

}

}

}

static void ExtractBirthYears(string filePath)

{

// Define a regex pattern to match 4-digit numbers

Regex yearRegex = new Regex(@"\b\d{4}\b");

// Read the entire content of the file

string fileContent = File.ReadAllText(filePath);

// Find all matches of the regex pattern in the file content

MatchCollection matches = yearRegex.Matches(fileContent);

// Print each matched year

foreach (Match match in matches)

{

Console.WriteLine(match.Value);

}

}

}

}

Program 5:

using System;

using System.IO;

using System.Text.RegularExpressions;

namespace \_05Rapunzel01\_DSPS1

{

internal class Program

{

static void Main(string[] args)

{

// Read all text from the file

string text = File.ReadAllText("Rapunzel.txt");

// Count the number of characters

Console.WriteLine($"# characters: {text.Length}");

// Count the number of lines using ReadAllLines

Console.WriteLine($"# lines: {File.ReadAllLines("Rapunzel.txt").Length}");

// Count the number of lines using StreamReader and a while loop

StreamReader sr = File.OpenText("Rapunzel.txt");

int lines = 0;

while (!sr.EndOfStream)

{

sr.ReadLine();

lines++;

}

sr.Close();

Console.WriteLine($"# lines: {lines}");

// Count the number of 'A's and 'a's using StreamReader and a while loop

int countA = 0;

sr = File.OpenText("Rapunzel.txt");

while (!sr.EndOfStream)

{

char c = (char)sr.Read();

if (c == 'A' || c == 'a')

{

countA++;

}

}

sr.Close();

Console.WriteLine($"# A's or a's: {countA}");

// Count the number of 'A's and 'a's using foreach and ToLower

countA = 0;

foreach (char item in text.ToLower())

{

if (item == 'a')

{

countA++;

}

}

Console.WriteLine($"# A's or a's: {countA}");

// Count the number of 'A's and 'a's using a for loop

countA = 0;

for (int i = 0; i < text.Length; i++)

{

if (text[i] == 'A' || text[i] == 'a')

{

countA++;

}

}

Console.WriteLine($"# A's or a's: {countA}");

// Count the number of words by counting spaces

int countWord = 0;

foreach (char item in text)

{

if (item == ' ')

{

countWord++;

}

}

Console.WriteLine($"# words: {countWord}");

// Count the number of words using a while loop and checking for non-alphabet characters

countWord = 0;

sr = File.OpenText("Rapunzel.txt");

while (!sr.EndOfStream)

{

char c = (char)sr.Read();

if (!"abcdefghijklmnopqrstuvwxyz".Contains(Char.ToLower(c)))

{

if (c != '\'')

{

countWord++;

}

}

}

sr.Close();

Console.WriteLine($"# words: {countWord}");

// Count the occurrences of the word "Rapunzel" using foreach and building words character by character

string word = "";

int countRapunzel = 0;

foreach (char c in text.ToLower())

{

if ("abcdefghijklmnopqrstuvwxyz".Contains(c))

{

word += c;

}

else

{

if (word == "rapunzel")

{

countRapunzel++;

}

word = "";

}

}

Console.WriteLine($"# rapunzels: {countRapunzel}");

// Count the occurrences of the word "Rapunzel" using Regex

Regex regex = new Regex("rapunzel", RegexOptions.IgnoreCase);

MatchCollection matches = regex.Matches(text);

Console.WriteLine($"# rapunzels: {matches.Count}");

// Count the number of words using Regex

regex = new Regex(@"\w+", RegexOptions.IgnoreCase);

matches = regex.Matches(text);

Console.WriteLine($"# words: {matches.Count}");

// Replace the word "Rapunzel" with "Anthony" and print the result

regex = new Regex("rapunzel", RegexOptions.IgnoreCase);

Console.WriteLine(regex.Replace(text, "Anthony"));

}

}

}