|  |
| --- |
| **Day 15 - Assignment**  **By Manoj Karnatapu - NBHealthCareTechnologies** |

|  |
| --- |
| **Assignment 1** |
| **Write a C# Code with at least 10 methods, in File Operations.** |
| **Answer** |
| |  | | --- | | **Code** | | using System;  using System.IO;  using System.Text;  // Author : Manoj.Karnatapu  // Purpose : File Operations Using atleast 10 Methods.  // For Reference, Check Day15Project1 in the same Repository.  namespace Day15Project1  {  internal class Program  {  static void Main(string[] args)  {  // Assigning a File Path using Verbatim String Manipulation method.  string fileName = @"D:\C#\GitHub\FileOperations.txt";  // Creating a File with Given File Path  StreamWriter sw = File.CreateText(fileName);  // When we Use WriteLine Method in StreamWriter class,  // It will Write in New Line each time it is called.  sw.WriteLine("Hi this is my first File Operations using C# code");  // The below statement is to have an empty line break in the file.  sw.WriteLine("");  sw.WriteLine("Writing Data using StringWriter, by using WriteLine Method.");  // When We Use Write Method in StreamWriter Class,  // it will start from the place of Cursor Exits after the last update, If Any.  sw.Write("This Line is by Write Method");  sw.Write("This is Second Line using Write Method\n");  sw.Close();  // Appending a Text  sw = File.AppendText(fileName);  sw.WriteLine("This");  sw.WriteLine("is Extra");  sw.WriteLine("Text");  Console.WriteLine("\nFile Appending is Done by Append\_Text Method.");  // We Need to Close the File, When ever we Create/open/Read a File, in file Operations.  sw.Close();  // Reading a File Using OpenText() Method  StreamReader sr = File.OpenText(fileName);  string s;  Console.WriteLine("\n");  while ((s = sr.ReadLine()) != null)  Console.WriteLine(s);    sr.Close();  string fileNewPath = @"D:\FileOperations.txt";  // Moving a File From One Path to Another Path  if(File.Exists(fileNewPath))  File.Delete(fileNewPath);  Console.WriteLine("\nAlready the File is Present, So Deleting the old file & Creating a New File.");  File.Move(fileName, fileNewPath);  Console.WriteLine("\nFile Moved to New Path, Successfuly [D:/FileOperations.txt]");  // Copying the File From NewPAth to Old path to make a Duplicate.  /\*if (File.Exists(fileName))  File.Delete(fileName);  Console.WriteLine("\nAlready the File is Present, So Deleting the old file & Creating a New File.");\*/  File.Copy(fileNewPath, fileName);  Console.WriteLine("\nFile Copying is Done Successfully, to old Path\n");    // Opening a Text File  File.OpenText(fileName).Close();  Console.WriteLine("\nFile opened Successfully, without any Errors");  string appendText = "This is an Extra text from Append\_All\_Text Method";  File.AppendAllText(fileName, appendText, Encoding.UTF8);  Console.WriteLine("\nFile Appended with Extra Text , Successfully");  string readText = File.ReadAllText(fileName);  Console.WriteLine("\n\n Reading All Text From the File\n");  Console.WriteLine(readText);  Console.ReadLine();  }  }  } | | **Output** | |

|  |
| --- |
| **Assignment 2** |
| **Write a C# Code, to copy from one folder to another folder, by scheduling the job using Task Schedular in Windows OS.** |
| **Code** |
| using System;  using System.IO;  // Author : Manoj.Karnatapu  // Purpose : Write a C# Code, to copy from one folder to another folder, by scheduling the job using Task Schedular in Windows OS.  // For Reference, Check Day15Project2 in the same Repository.  namespace Day15Project2  {  internal class Program  {  static void Main(string[] args)  {  string filePath = @"D:\C#\GitHub\FileOperations.txt";  string TaskSchedularPath = @"D:\C#\TaskSchedular\TaskSchedularPath.txt";  if(File.Exists(TaskSchedularPath))  File.Delete(TaskSchedularPath);  File.Copy(filePath, TaskSchedularPath);  Console.WriteLine("File Copying is Done");  Console.ReadKey();  }  }  } |
| **Output** |
| **(1). Open Task Schedular in Windows App Menu.**  **(2). Select “Create Task” Option from the Right-Side Menu.**    **(3). Give Name as : File Copying & Description as : File Copying @ 8:05pm Everyday from one folder to another file/folder.**      **(4). Select Triggers Tab & choose Time and Date as per your choice.**  **Select Daily in the Left-side Pane. (I chosen One Time as per my choice of running only once. ) And Press “OK” button.**    **(5). Now the New Trigger is Created.**      **(6). Go To “Actions” Tab. Click on “New” to create an action to be trigged by the task schedular.**    **(7). Now Chose “Browse” button, to add the Exe file of Our Project Code to be executed by the task schedular on the given trigger time.**    **(8). Select the Exe file of your Project code Created when we Rebuild.**    **(9). Press “OK” Button When Selection of EXE file is Done.**    **(10). Final Step. Select the “OK” Button Which is Highlighted.**    **(11). Task Schedular is Scheduled, and When the timer Hits the trigged Value given in Step (4). The Output is Shown in the below image.**    **(12). Hence the Task Schedular Working Fine and Created a New file in the Given Location as per the Code Written.** |

|  |
| --- |
| **Assignment 3** |
| **Write a C# Code, to write data into file/append the data Using Stream Writer Class.** |
| **Code** |
| using System;  using System.IO;  // Author : Manoj.Karnatapu  // Purpose : Write a C# Code, to write data into file/append the data Using Stream Writer Class.  // For Reference, Check Day15Project3 in the same Repository.  namespace Day15Project3  {  internal class Program  {  static void Main(string[] args)  {  // We are Creating a File using streamWriter Class. It will over Ride Each time we run the code  StreamWriter sw = new StreamWriter(@"D:\C#\GitHub\StreamWriterExample.txt");  sw.WriteLine("Hi this is Manoj");  sw.WriteLine("This is a File Operation using StreamWriter with WriteLine Method");  sw.Close();  Console.WriteLine("\n Writing File Is done, by using Stream Writer Class by writeLine Method.\n");  // We are Creating a File using streamWriter Class. It will not over Ride Each time when we run the code.  // Instead it will Append the Text of Lines, Each time We Run the Code.  StreamWriter writer = new StreamWriter(@"D:\C#\GitHub\StreamWriterAppendExample.txt", true);  writer.WriteLine("This is From New object of Stream Writer,");  writer.WriteLine("using Append by assigning True, while creating object for StreamWriter");  writer.Close();  Console.WriteLine("\n Appending is done, by using Stream Writer Class, By Enabling Append Method(true).");  Console.ReadKey();  }  }  } |
| **Output** |
|  |

|  |
| --- |
| **Assignment 4** |
| **Write a C# Code, To Read Data From a File, Using File Operations.** |
| **Code** |
| using System;  using System.IO;  // Author : Manoj.Karnatapu  // Purpose : Write a C# Code, To Read Data From a File, Using File Operations by Stream Writer Class.  // For Reference, Check Day15Project4 in the same Repository  namespace Day15Project4  {  internal class Program  {  static void Main(string[] args)  {  // We are Creating a File using streamWriter Class. It will not over Ride Each time when we run the code.  // Instead it will Append the Text of Lines, Each time We Run the Code.  StreamWriter writer = new StreamWriter(@"D:\C#\GitHub\StreamWriterAppendExample.txt", true);  writer.WriteLine("This is From New object of Stream Writer,");  writer.WriteLine("using Append by assigning True, while creating object for StreamWriter");  writer.Close();  Console.WriteLine("\n Appending is done, by using Stream Writer Class, By Enabling Append Method(true).");  // We Are Reading the File Content using Stream Writer Class From System.IO namespace.  StreamReader reader = new StreamReader(@"D:\C#\GitHub\StreamWriterAppendExample.txt");  reader.ReadToEnd();  reader.Close();  Console.WriteLine("\n Reading is done, by using Stream Writer Class, by ReadToEnd Method.");    Console.ReadKey();  }  }  } |
| **Output** |
|  |

|  |
| --- |
| **Assignment 5** |
| **Write a C# Code, for Quiz Application & store the Scores in Flat File.** |
| **Code** |
|  |
| **Output** |
|  |