**FileInfo Class Exmaples**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

using System.IO;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// CreateFile();

// MoveFile1();

DeleteFile();

}

static void CreateFile()

{

FileInfo file = new FileInfo("E:\\C#\\Sample.txt");

file.Create();

}

static void MoveFile1()

{

FileInfo file1 = new FileInfo("E:\\C#\\Sample.txt");

file1.MoveTo("E:\\C#\\Sample-New.txt");

}

static void DeleteFile()

{

FileInfo file = new FileInfo("E:\\C#\\Sample-New.tx t");

file.Delete();

}

}

}

**File Attributes Example:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

using System.IO;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

FileInfo file = new FileInfo("E:\\C#\\ResumeDetails.xlsx");

String name = file.Name;

Console.WriteLine("Name of the File :" + name);

string fullname=file.FullName;

Console.WriteLine("Full Name of the File :" + fullname);

long size = file.Length;

Console.WriteLine("Size of the File :" + size);

Console.ReadLine();

}

}

}

**DirectoryInfo Class :**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

using System.IO;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// CreateDir();

// MoveDir();

DeleteDir();

}

static void CreateDir()

{

DirectoryInfo dir = new DirectoryInfo("E:\\C#\\Sample");

dir.Create();

}

static void MoveDir()

{

DirectoryInfo dir = new DirectoryInfo("E:\\C#\\Sample");

dir.MoveTo("E:\\C#\\Sample-New");

}

static void DeleteDir()

{

DirectoryInfo dir = new DirectoryInfo("E:\\C#\\Sample-New");

dir.Delete();

}

}

}

**File Collections and Directory Collections:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

using System.IO;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// FileCollections();

DirectoryCollections();

Console.ReadLine();

}

static void FileCollections()

{

DirectoryInfo dir = new DirectoryInfo("E:\\C#");

foreach (FileInfo file in dir.GetFiles())

{

Console.WriteLine(file.FullName);

if (file.FullName.EndsWith(".xlsx"))

{

file.Delete();

}

}

}

static void DirectoryCollections()

{

DirectoryInfo dir = new DirectoryInfo("E:\\C#");

foreach (DirectoryInfo dir1 in dir.GetDirectories())

{

Console.WriteLine(dir1.FullName);

}

}

}

}

**StreamReader and StreamWriter classes**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

using System.IO;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// WriteContent();

ReadContent();

Console.ReadLine();

}

static void WriteContent()

{

StreamWriter writer = new StreamWriter("E:\\C#\\Test1.txt");

writer.WriteLine("This is an example for File operations ...");

writer.WriteLine("TextWriter is the Abstract Class in C#");

writer.Flush();

writer.Close();

}

static void ReadContent()

{

string line = null;

StreamReader reader = new StreamReader("E:\\C#\\Test1.txt");

while((line=reader.ReadLine())!=null)

{

Console.WriteLine(line);

}

reader.Close();

}

}

}

**StringWriter and StringReader classes:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

using System.IO;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

WriteReadContent();

Console.ReadLine();

}

static void WriteReadContent()

{

string str = "Bangalore is a capital city of Karnakata \n";

str+=" It is near to Mysore around 120 KM \n";

str += " There are many Parks and Palaces in Bangalore \n";

StringBuilder sb = new StringBuilder();

StringWriter writer = new StringWriter(sb);

writer.WriteLine(str);

writer.Flush();

writer.Close();

StringReader reader = new StringReader(sb.ToString());

while(reader.Peek() > -1)

{

Console.WriteLine(reader.ReadLine());

}

reader.Close();

}

}

}

**Serialization and de Serialization**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CSharpTestingExamples.datastructures

{

[Serializable]

class Employee1

{

public string ename;

public int id;

public string job;

public Employee1(string ename,int id,string job)

{

this.ename = ename;

this.id = id;

this.job = job;

}

public void showEmpName()

{

Console.WriteLine("Employee Name :" + this.ename);

}

public void showEmpID()

{

Console.WriteLine("Employee ID :" + this.id);

}

public void showEmpJob()

{

Console.WriteLine("Employee Job :" + this.job);

}

}

}

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

using System.IO;

using CSharpTestingExamples.datastructures;

using System.Runtime.Serialization.Formatters.Binary;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

Employee1 obj = new Employee1("Santosh", 7900, "Manager");

try

{

BinaryFormatter formatter = new BinaryFormatter();

FileStream stream = new FileStream("E:\\C#\\Example.txt", FileMode.Create, FileAccess.Write);

formatter.Serialize(stream, obj);

stream.Close();

//Deserialization

FileStream stream1 = new FileStream("E:\\C#\\Example.txt", FileMode.Open, FileAccess.Read);

Employee1 obj1=(Employee1) formatter.Deserialize(stream1);

obj1.showEmpName();

obj1.showEmpID();

obj1.showEmpJob();

stream1.Close();

}

catch(Exception e)

{

Console.WriteLine(e.ToString());

}

Console.ReadLine();

}

}

}