C# Tutorial Notes for Beginners

1. Introduction to File I/O

File I/O (Input/Output) in C# allows you to read from and write to files. This is useful for storing data persistently on disk.

2. Namespaces Required

You need to include the following namespaces:

```
using System;
using System.IO;
```

3. Writing to a File

```
// Writes text to a file (overwrites if file exists)
File.WriteAllText("example.txt", "Hello, File!");

// Appends text to the file
File.AppendAllText("example.txt", "\nAppended text.");

// Write multiple lines
string[] lines = { "Line1", "Line2", "Line3" };
File.WriteAllLines("lines.txt", lines);
```

4. Reading from a File

```
// Read entire content as string
string content = File.ReadAllText("example.txt");
```

```
Console.WriteLine(content);


// Read all lines as string array

string[] lines = File.ReadAllLines("lines.txt");

foreach (string line in lines)

    Console.WriteLine(line);
```

5. Using StreamWriter

```
using (StreamWriter writer = new StreamWriter("data.txt"))

{

    writer.WriteLine("First line");

    writer.WriteLine("Second line");

}
```

6. Using StreamReader

```
using (StreamReader reader = new StreamReader("data.txt"))

{

    string line;

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

    {

        Console.WriteLine(line);

    }

}
```

7. Checking File or Directory Existence

```
bool fileExists = File.Exists("data.txt");

bool dirExists = Directory.Exists("myFolder");
```

8. Creating Directories and Files

```
Directory.CreateDirectory("NewFolder");

File.Create("NewFolder/newfile.txt").Close();
```

9. Copying, Moving, Deleting Files

```
File.Copy("source.txt", "copy.txt", true);

File.Move("copy.txt", "moved.txt");

File.Delete("moved.txt");
```

10. Getting File Info

```
FileInfo fi = new FileInfo("data.txt");

Console.WriteLine("Size: " + fi.Length);

Console.WriteLine("Created: " + fi.CreationTime);
```

11. Exception Handling in File I/O

```
try

{

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

}

catch (FileNotFoundException e)
```

```
{
    Console.WriteLine("File not found: " + e.Message);
}
catch (IOException e)
{
    Console.WriteLine("IO Error: " + e.Message);
}
```

12. Summary

- Use `File` class for simple operations (read/write).

- Use `StreamReader`/`StreamWriter` for more control.

- Always handle exceptions when dealing with files.