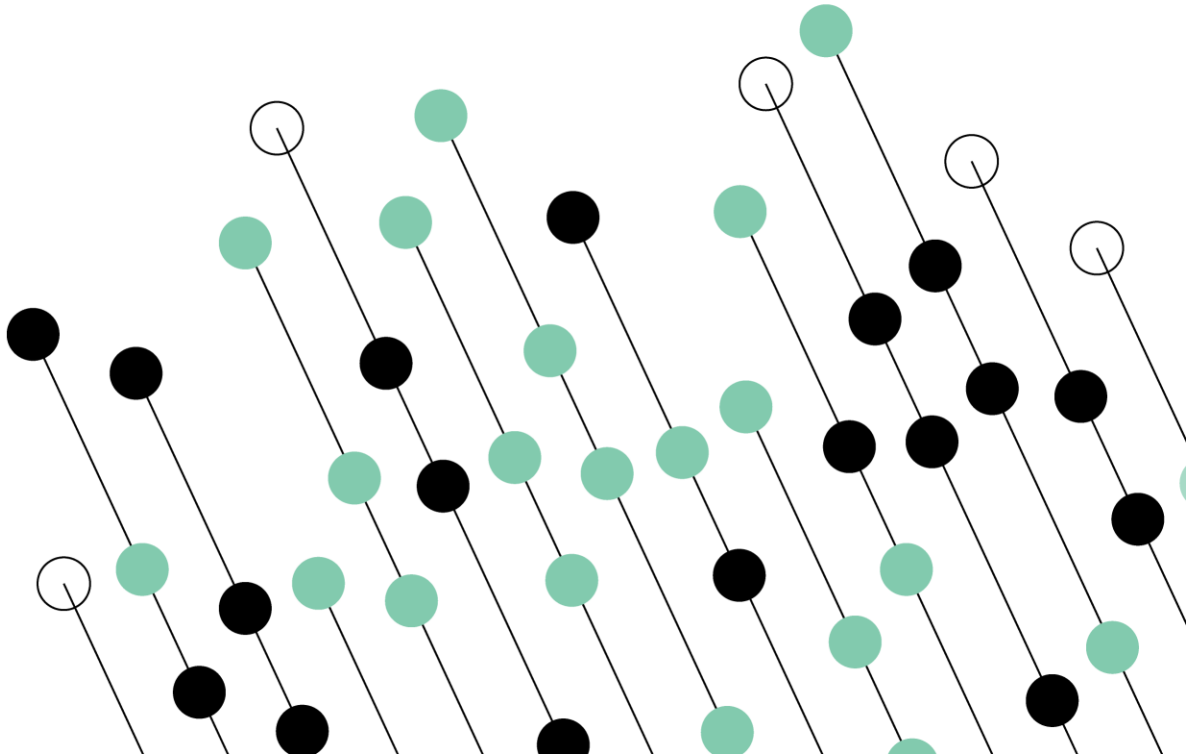
Working with files



Contents

Objectives

- Describe the basics of IO in C#
- Read data from and write data to a file

Contents

- Accessing files and directories using the file class

Hands-on labs

- Write code to read / write to a file
- Read and write JSON data

Basics of file I/O

- **C# provide classes to perform I/O.**
- Referred to an Input-Output channel as a '**stream**'.

- **Stream** represents an input source or an output destination like:
  - disk files, devices.
  - other programs.
  - memory arrays.

Code examples usin System.IO.File

```
// Read all text
string text = File.ReadAllText("example.txt");
Console.WriteLine(text);
```

```
// Read all lines as array
string[] lines = File.ReadAllLines("example.txt");
foreach (string line in lines)
    Console.WriteLine(line);
```

```
// Write(overwrite) text
File.WriteAllText("example.txt", "Hello, world!");
```

```
// Append text
File.AppendAllText("example.txt", "\nAppended line");
```

Using Using StreamReader / StreamWriter

Use Streams for more control, (e.g., encoding or large files)

```
// Read line by line
using (StreamReader reader = new StreamReader("example.txt")) {
    string line;
    while ((line = reader.ReadLine()) != null)   {
        Console.WriteLine(line);
    }
}
```

```
// Write line by line
using (StreamWriter writer = new StreamWriter("example.txt")) {
    writer.WriteLine("First line");
    writer.WriteLine("Second line");
}
```

Append using StreamWriter

```
// Append mode
using (StreamWriter writer = new StreamWriter("example.txt", append: true))
{
    writer.WriteLine("Appended later");
}
```

File info

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

Console.WriteLine($"Size: {fi.Length} bytes");

Console.WriteLine($"Created: {fi.CreationTime}");
```

Directory class example

```
string path = @"C:\Windows";

string[] files = Directory.GetFiles(path);

foreach (string file in files)
{
    Console.WriteLine(file);
}
```

Events

```
FileSystemWatcher watcher = new FileSystemWatcher();
watcher.Path = @"C:\MyFolder";

watcher.IncludeSubdirectories = false;

watcher.NotifyFilter = NotifyFilters.FileName | NotifyFilters.LastWrite;

watcher.Filter = "*.*";

// Subscribe to events
watcher.Created += OnCreated;
watcher.Deleted += OnDeleted;
watcher.Renamed += OnRenamed;

// Start watching
watcher.EnableRaisingEvents = true;
```

```
private static void OnCreated(object sender,
                              FileSystemEventArgs e) {
    // code
}
```

What is JSON?

- **JSON stands for** Java**S**cript **O**bject **N**otation.
- **It is used to** store objects as text and transport data usually from a server to a client.
  - Restful requests often return JSON content.


```
[] Search
  {} 0
    Title : "Star Wars: Episode IV - A New Hope"
    Year : "1977"
    imdbID : "tt0076759"
    Type : "movie"
    Poster : "https://m.media-amazon.com/images/M/MV5BNzVlY2MwMjktM2E4OS00Y2Y3LWE3ZjctYzhkZGM3YzA1ZWM2XkEyXkFqcGdeQXVyNzkwMjQ5NzM@._V1_SX300.jpg"
  {} 1
  {} 2
  {} 3
  {} 4
  {} 5
  {} 6
  {} 7
  {} 8
  {} 9
  totalResults : "3269"
  Response : "True"
```


http://www.omdbapi.com/?apikey=ef5e4257&s=star

Processing JSON using C# Core

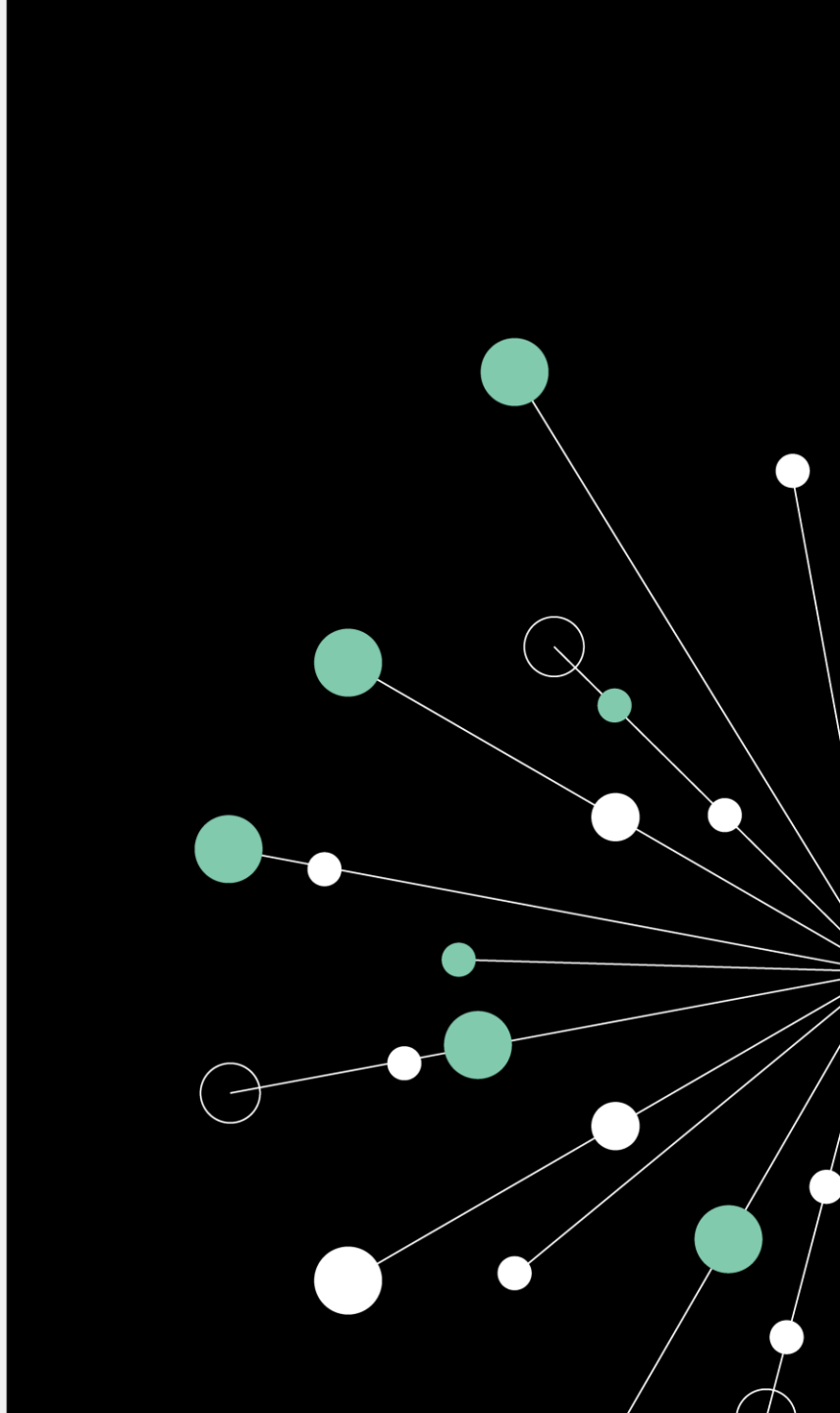
```
using System.Text.Json;

string json = File.ReadAllText("person.json");

Person person = JsonSerializer.Deserialize<Person>(json)!;

Console.WriteLine($"{person.Name} is {person.Age} years old.");
```

```
public class Person {
    public string Name { get; set; }
    public int Age { get; set; }
}
```

```
[
  { "Name": "Mike", "Age": 35 },
  { "Name": "Sarah", "Age": 29 }
]
```

In this chapter, we reviewed:

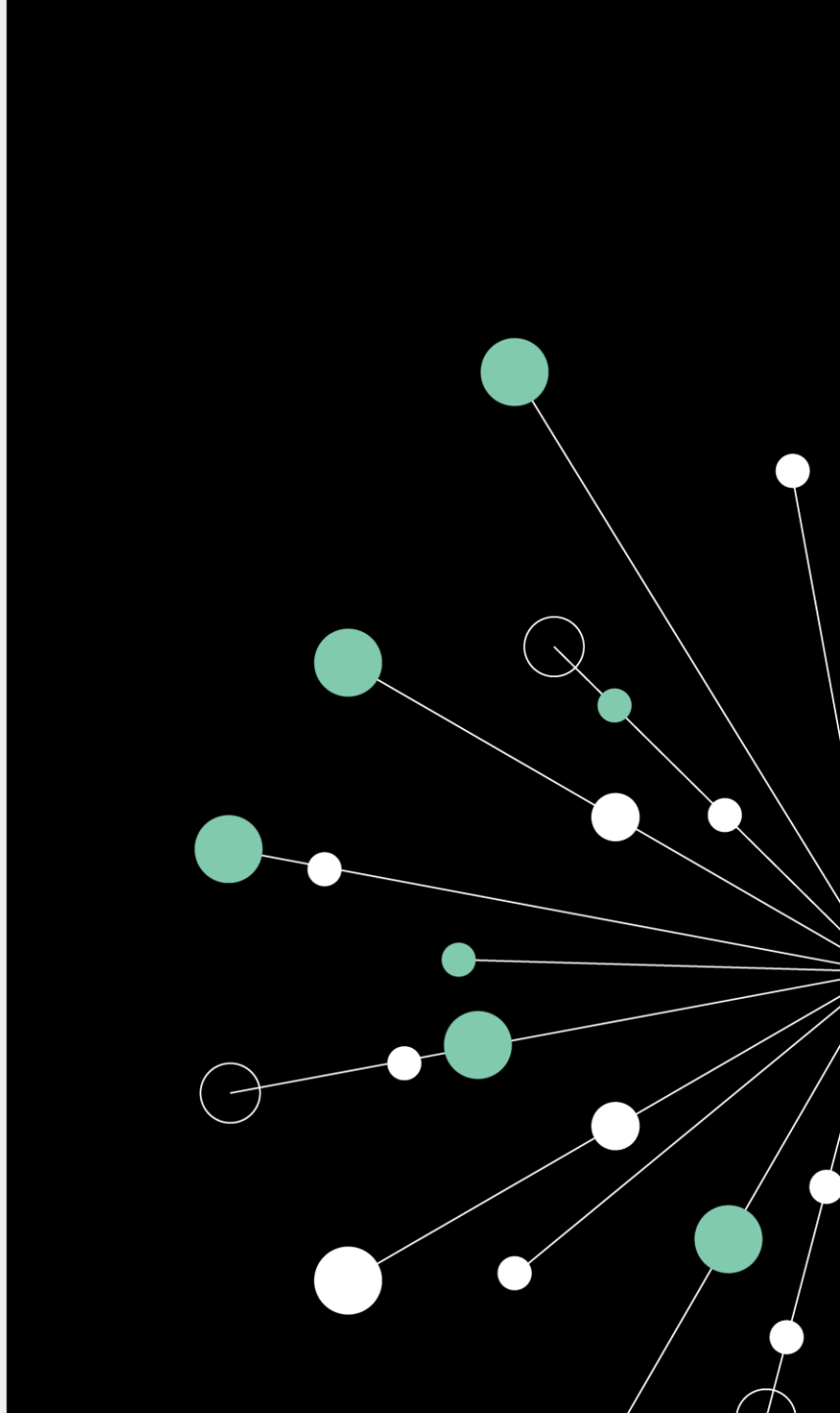
- Using streams to read and write data
- How to read JSON data from a file



QA

Hands-on labs

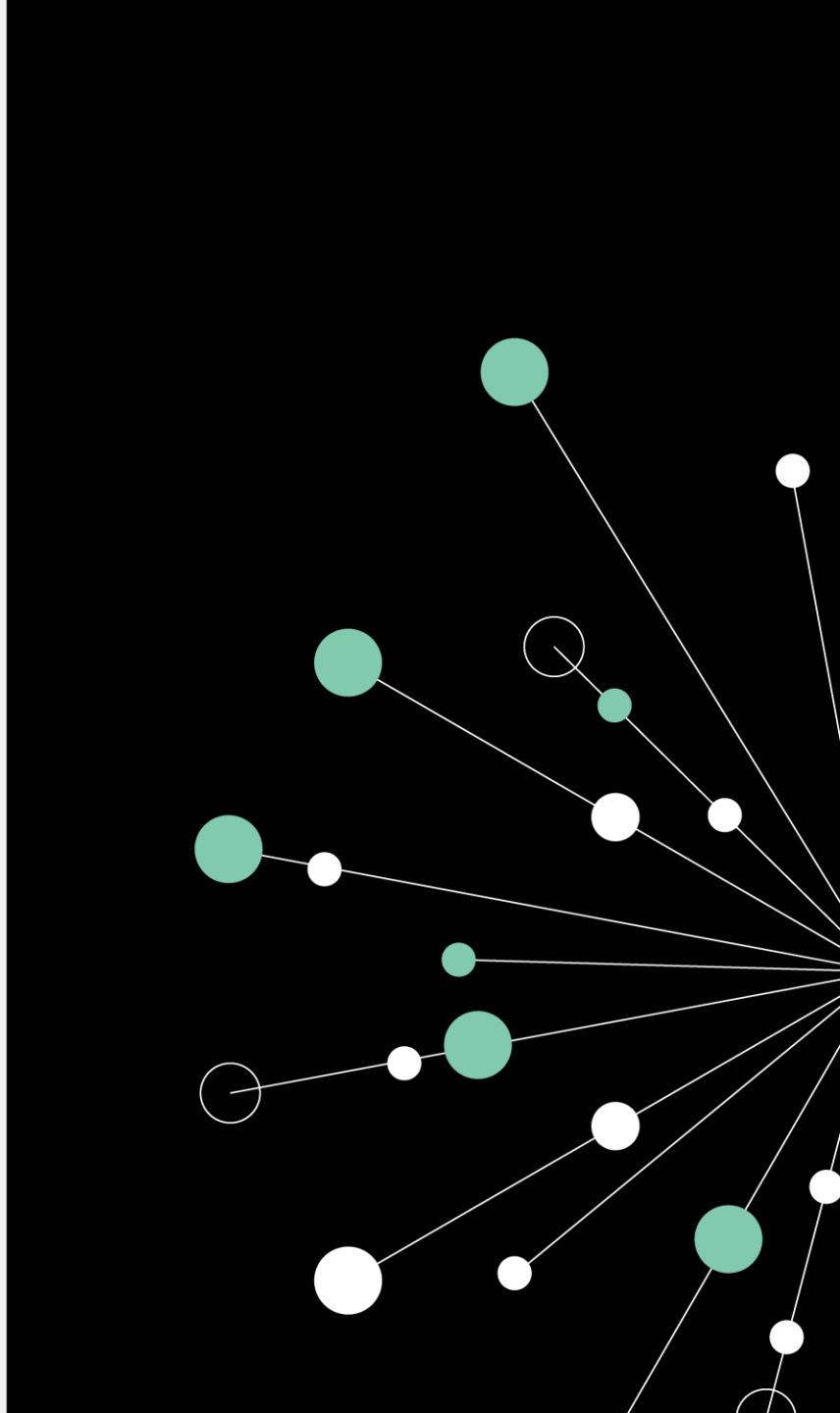
- Write code that investigates file system IO classes



Lab

- Accessing text files and processing JSON data using Java

Duration: 40 minutes



QA