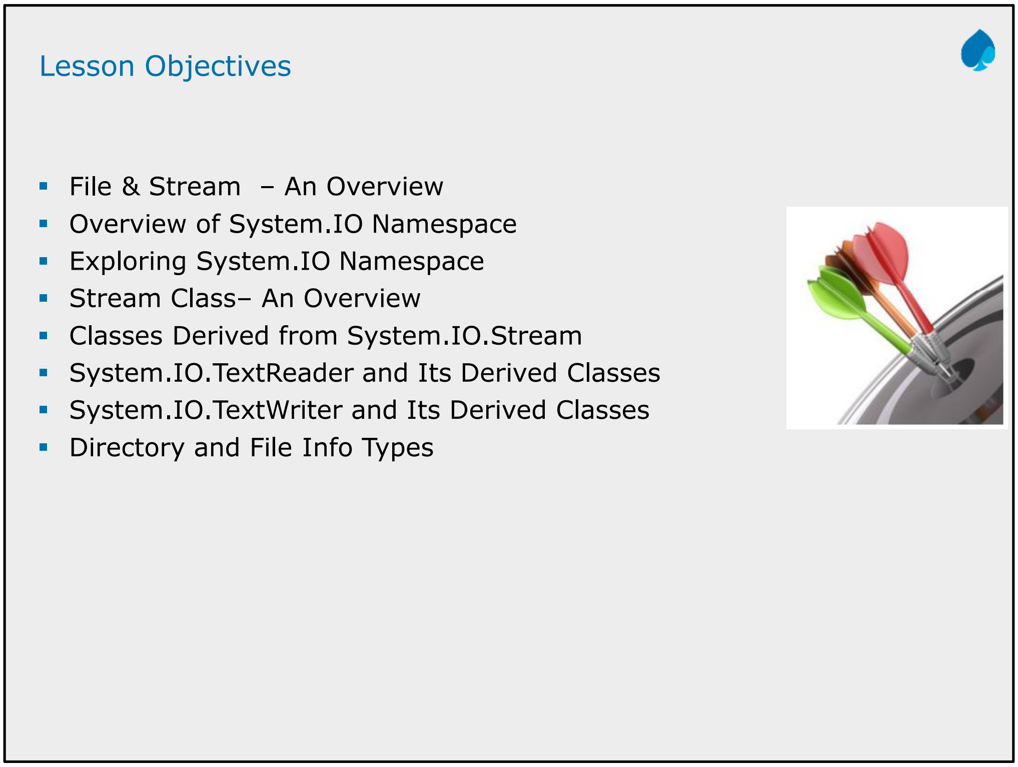
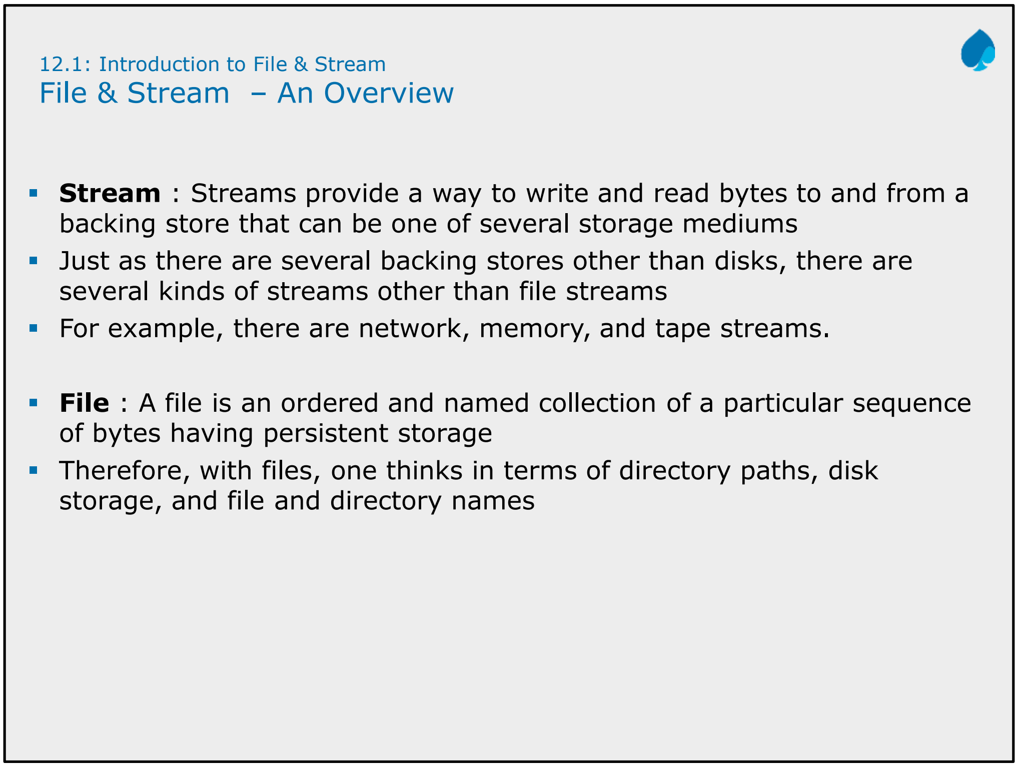
C# 7.0

Lesson 12 : File IO



Lesson Objectives

- File & Stream – An Overview
- Overview of System.IO Namespace
- Exploring System.IO Namespace
- Stream Class– An Overview
- Classes Derived from System.IO.Stream
- System.IO.TextReader and Its Derived Classes
- System.IO.TextWriter and Its Derived Classes
- Directory and File Info Types

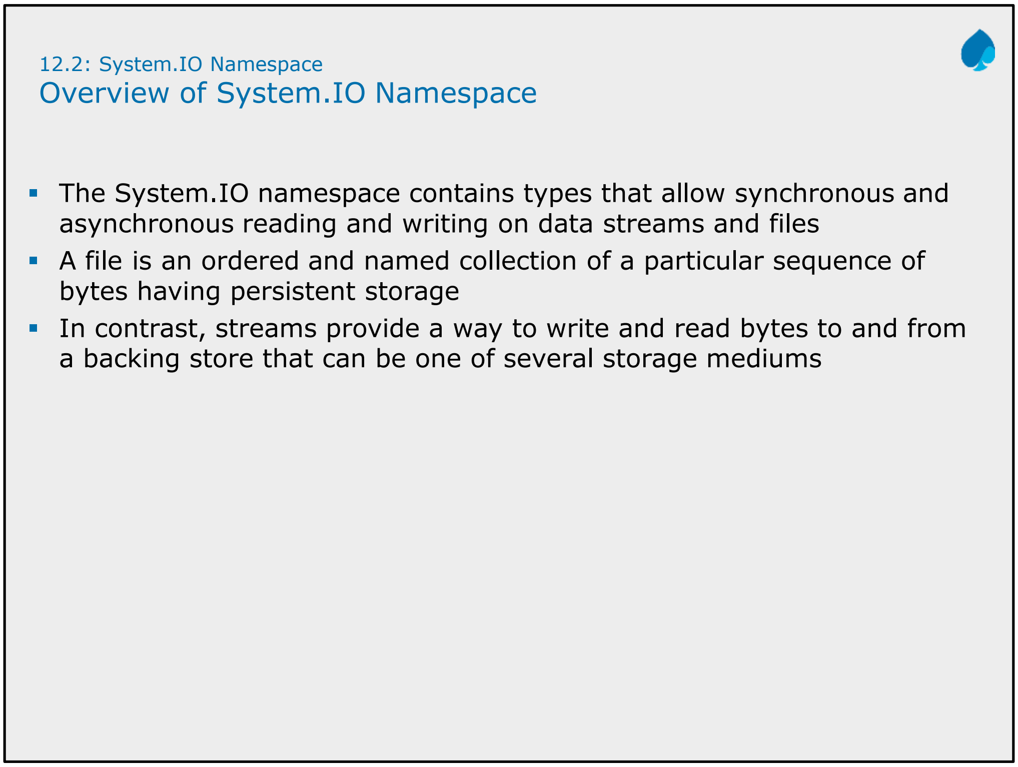


12.1: Introduction to File & Stream

File & Stream – An Overview

- **Stream** : Streams provide a way to write and read bytes to and from a backing store that can be one of several storage mediums
- Just as there are several backing stores other than disks, there are several kinds of streams other than file streams
- For example, there are network, memory, and tape streams.

- **File** : A file is an ordered and named collection of a particular sequence of bytes having persistent storage
- Therefore, with files, one thinks in terms of directory paths, disk storage, and file and directory names



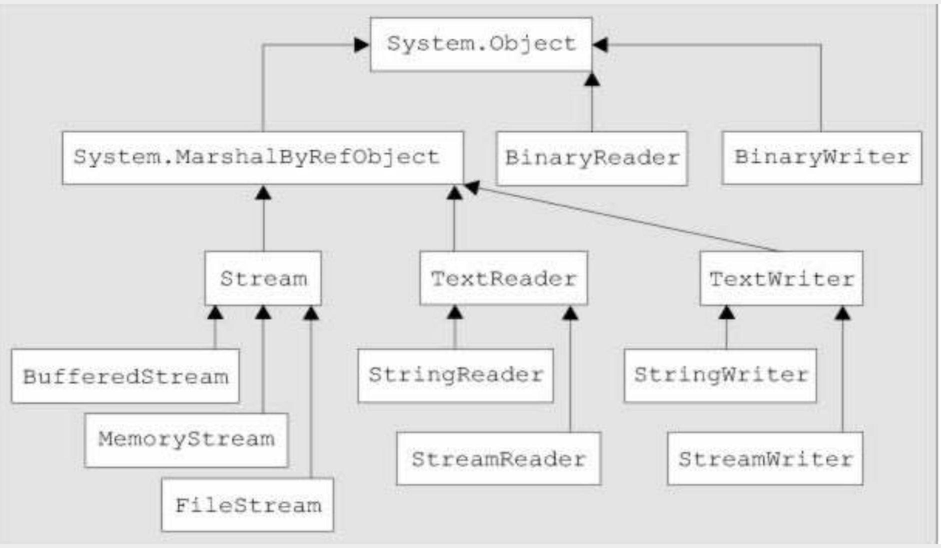
12.2: System.IO Namespace

Overview of System.IO Namespace

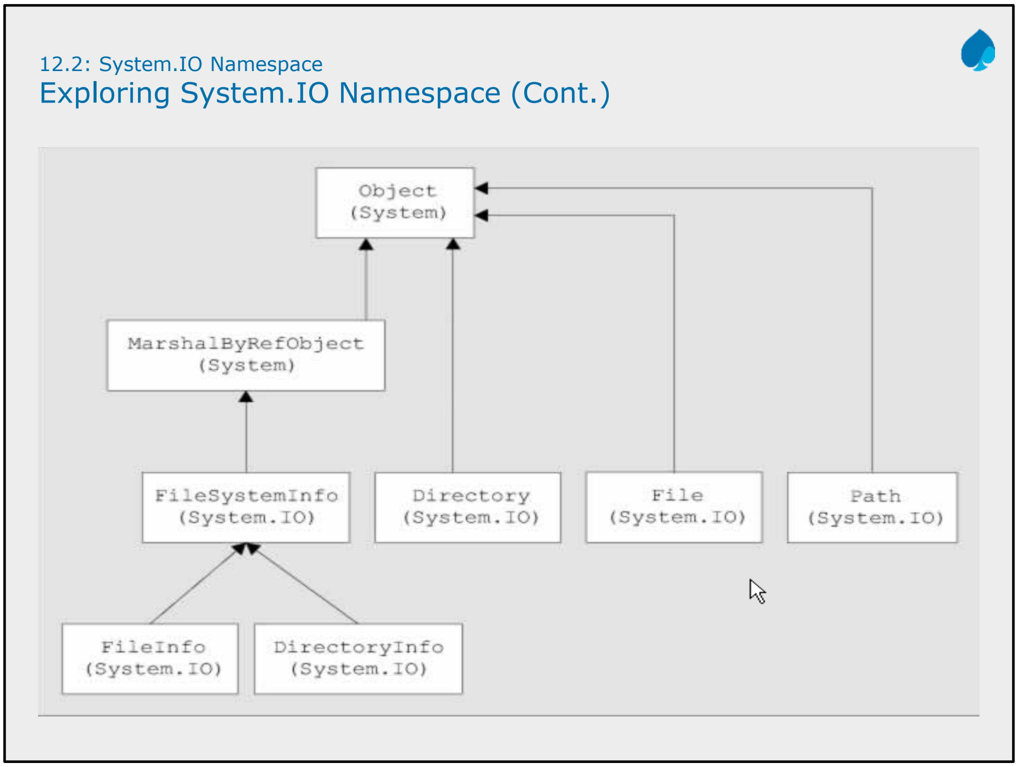
- The System.IO namespace contains types that allow synchronous and asynchronous reading and writing on data streams and files
- A file is an ordered and named collection of a particular sequence of bytes having persistent storage
- In contrast, streams provide a way to write and read bytes to and from a backing store that can be one of several storage mediums

12.2: System.IO Namespace

Exploring System.IO Namespace

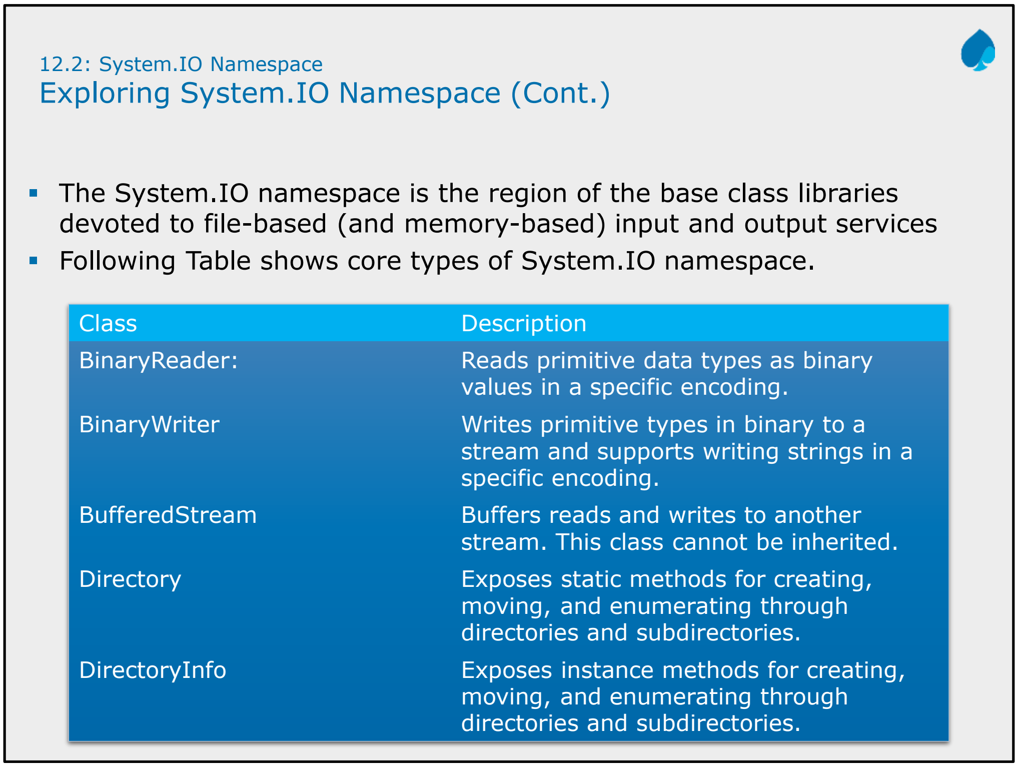


All classes that represent streams inherit from the Stream class. The Stream class and its derived classes provide a generic view of data sources and repositories, isolating the programmer from the specific details of the operating system and underlying devices.



12.2: System.IO Namespace

Exploring System.IO Namespace (Cont.)



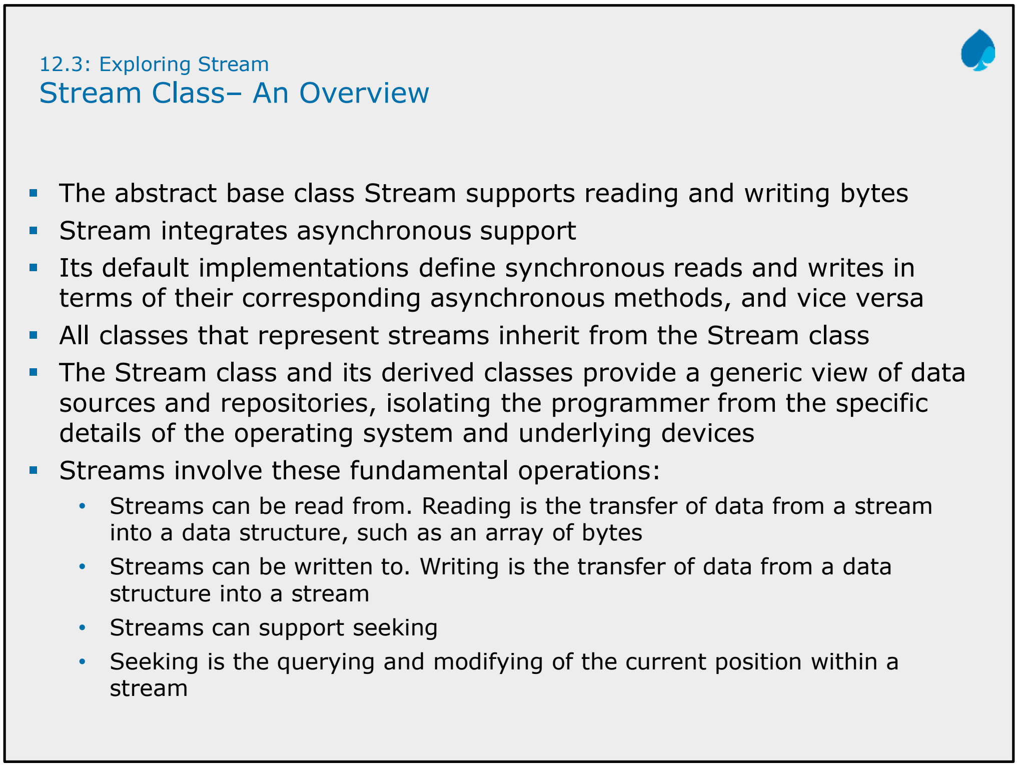
12.2: System.IO Namespace

Exploring System.IO Namespace (Cont.)

- The System.IO namespace is the region of the base class libraries devoted to file-based (and memory-based) input and output services
- Following Table shows core types of System.IO namespace.

Class	Description
BinaryReader:	Reads primitive data types as binary values in a specific encoding.
BinaryWriter	Writes primitive types in binary to a stream and supports writing strings in a specific encoding.
BufferedStream	Buffers reads and writes to another stream. This class cannot be inherited.
Directory	Exposes static methods for creating, moving, and enumerating through directories and subdirectories.
DirectoryInfo	Exposes instance methods for creating, moving, and enumerating through directories and subdirectories.

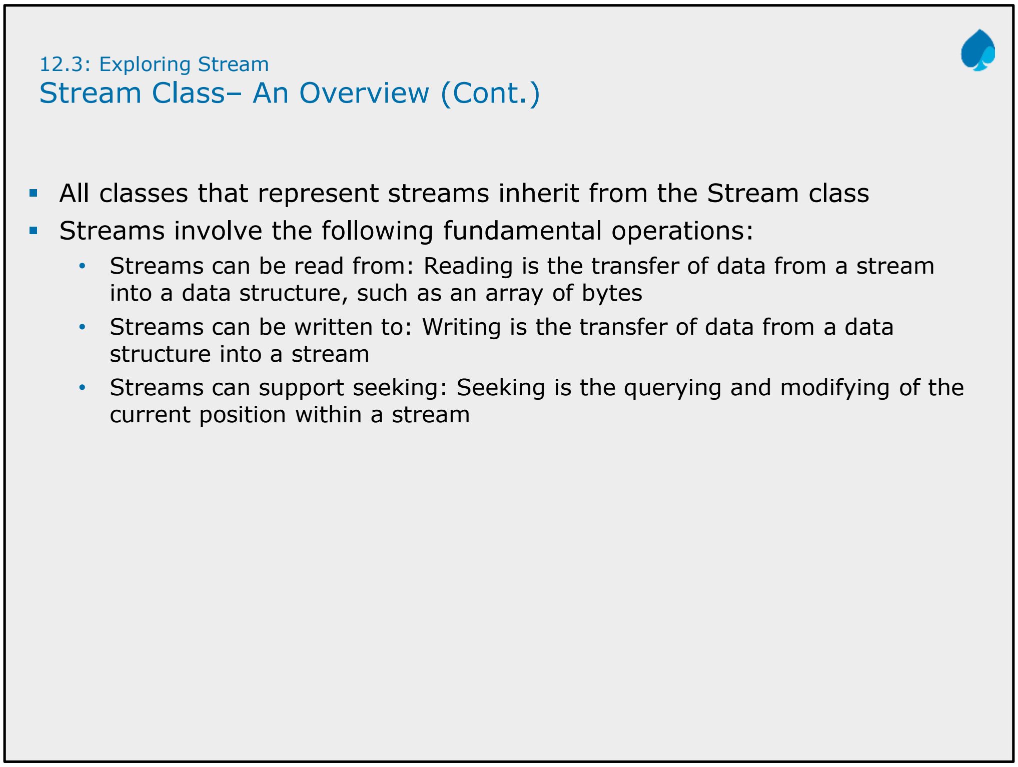
All classes that represent streams inherit from the Stream class. The Stream class and its derived classes provide a generic view of data sources and repositories, isolating the programmer from the specific details of the operating system and underlying devices.



12.3: Exploring Stream

Stream Class– An Overview

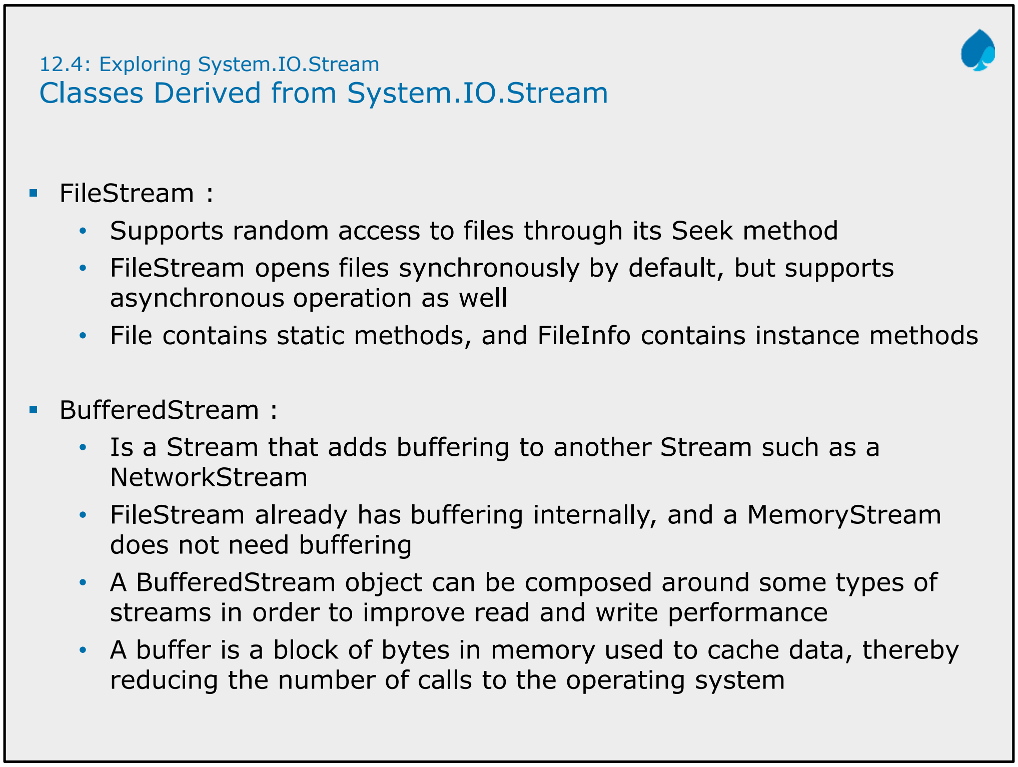
- The abstract base class Stream supports reading and writing bytes
- Stream integrates asynchronous support
- Its default implementations define synchronous reads and writes in terms of their corresponding asynchronous methods, and vice versa
- All classes that represent streams inherit from the Stream class
- The Stream class and its derived classes provide a generic view of data sources and repositories, isolating the programmer from the specific details of the operating system and underlying devices
- Streams involve these fundamental operations:
  - Streams can be read from. Reading is the transfer of data from a stream into a data structure, such as an array of bytes
  - Streams can be written to. Writing is the transfer of data from a data structure into a stream
  - Streams can support seeking
  - Seeking is the querying and modifying of the current position within a stream



12.3: Exploring Stream

Stream Class– An Overview (Cont.)

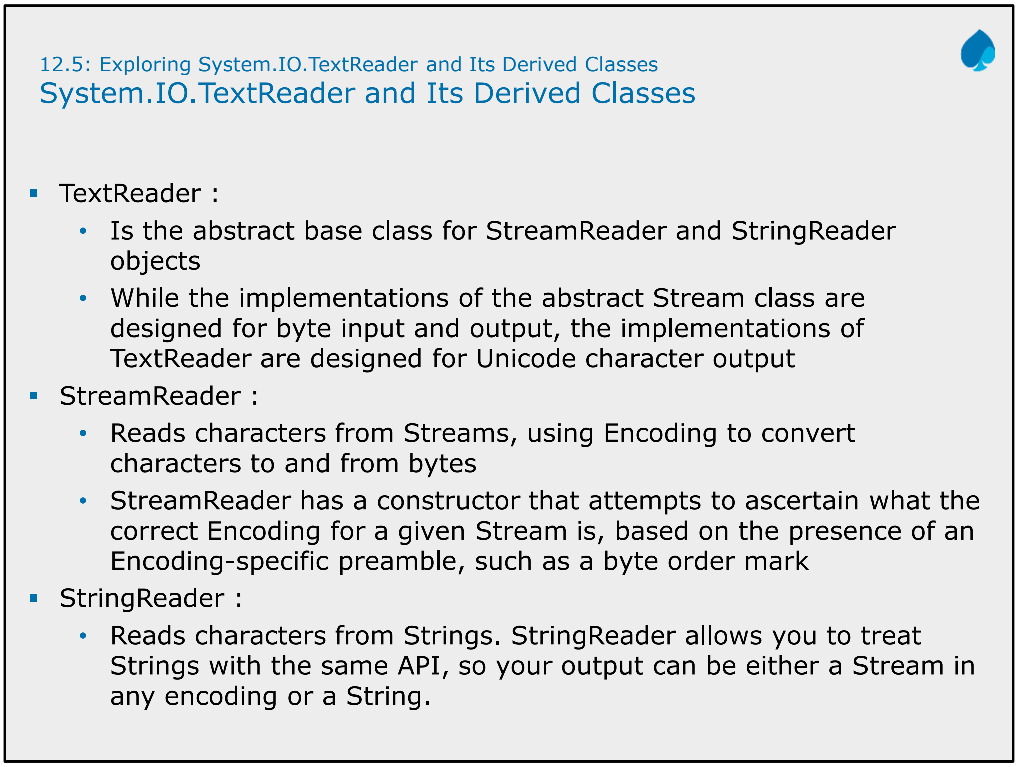
- All classes that represent streams inherit from the Stream class
- Streams involve the following fundamental operations:
  - Streams can be read from: Reading is the transfer of data from a stream into a data structure, such as an array of bytes
  - Streams can be written to: Writing is the transfer of data from a data structure into a stream
  - Streams can support seeking: Seeking is the querying and modifying of the current position within a stream



12.4: Exploring System.IO.Stream

Classes Derived from System.IO.Stream

- FileStream :
  - Supports random access to files through its Seek method
  - FileStream opens files synchronously by default, but supports asynchronous operation as well
  - File contains static methods, and FileInfo contains instance methods
- BufferedStream :
  - Is a Stream that adds buffering to another Stream such as a NetworkStream
  - FileStream already has buffering internally, and a MemoryStream does not need buffering
  - A BufferedStream object can be composed around some types of streams in order to improve read and write performance
  - A buffer is a block of bytes in memory used to cache data, thereby reducing the number of calls to the operating system



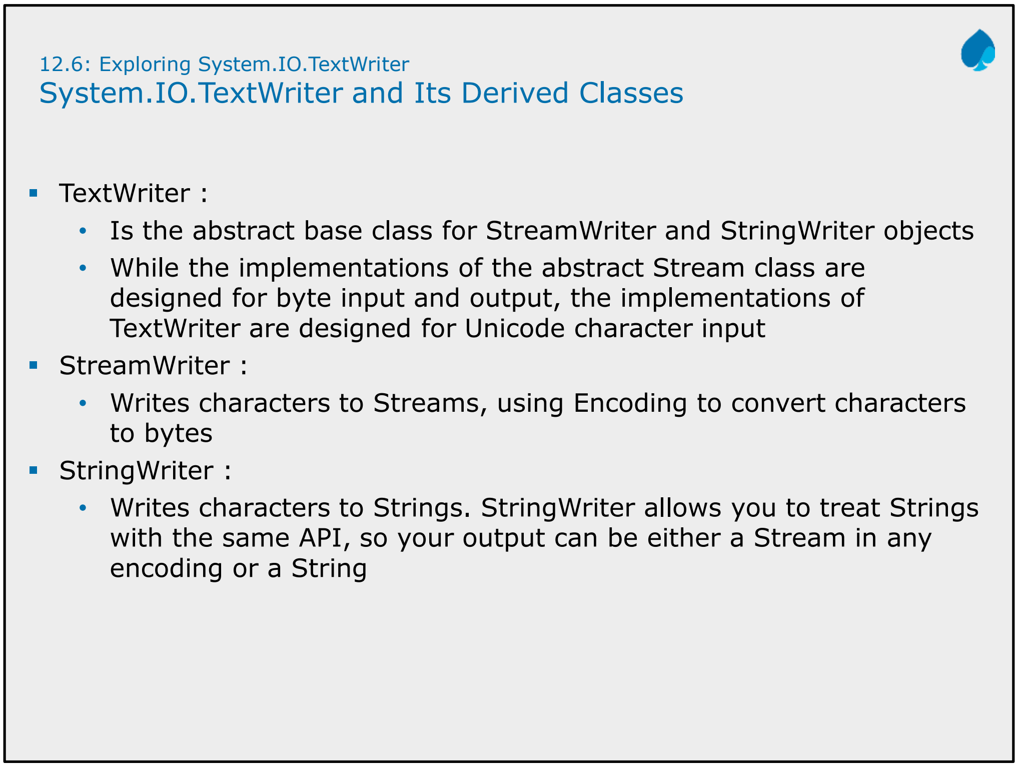
12.5: Exploring System.IO.TextReader and Its Derived Classes

System.IO.TextReader and Its Derived Classes

- TextReader :
  - Is the abstract base class for StreamReader and StringReader objects
  - While the implementations of the abstract Stream class are designed for byte input and output, the implementations of TextReader are designed for Unicode character output
- StreamReader :
  - Reads characters from Streams, using Encoding to convert characters to and from bytes
  - StreamReader has a constructor that attempts to ascertain what the correct Encoding for a given Stream is, based on the presence of an Encoding-specific preamble, such as a byte order mark
- StringReader :
  - Reads characters from Strings. StringReader allows you to treat Strings with the same API, so your output can be either a Stream in any encoding or a String.

Depending on the underlying data source or repository, streams might support only some of these capabilities.

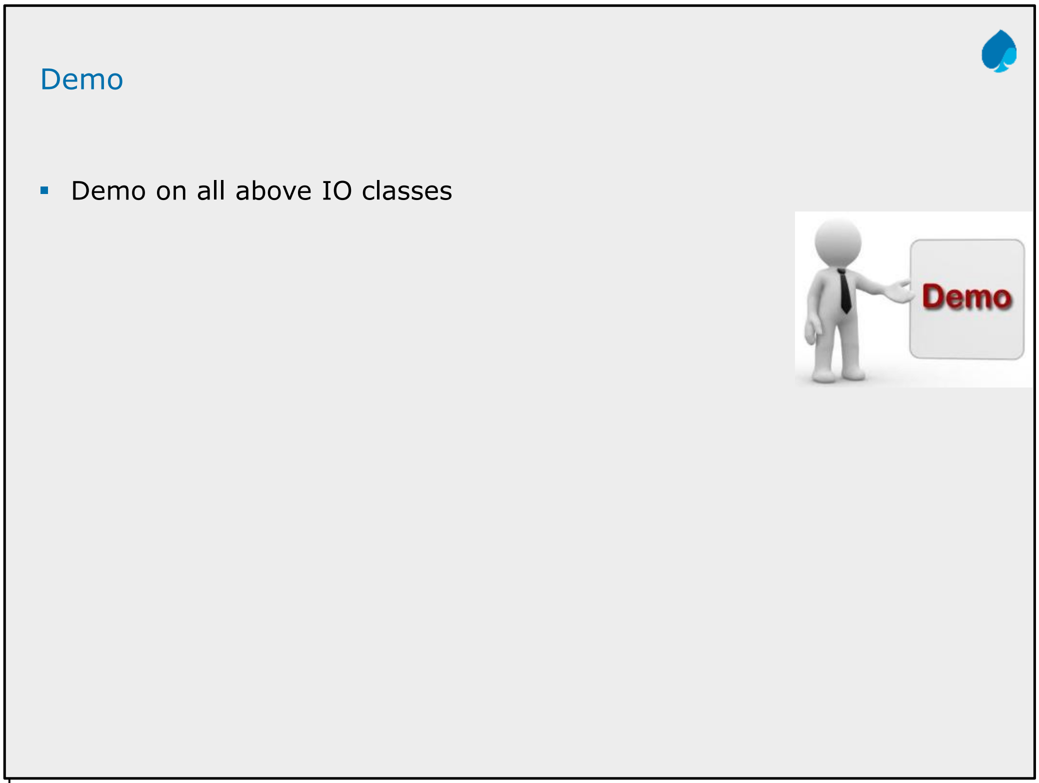
For example, NetworkStreams do not support seeking. The CanRead, CanWrite, and CanSeek properties of Stream and its derived classes determine the operations that various streams support.



12.6: Exploring System.IO.TextWriter

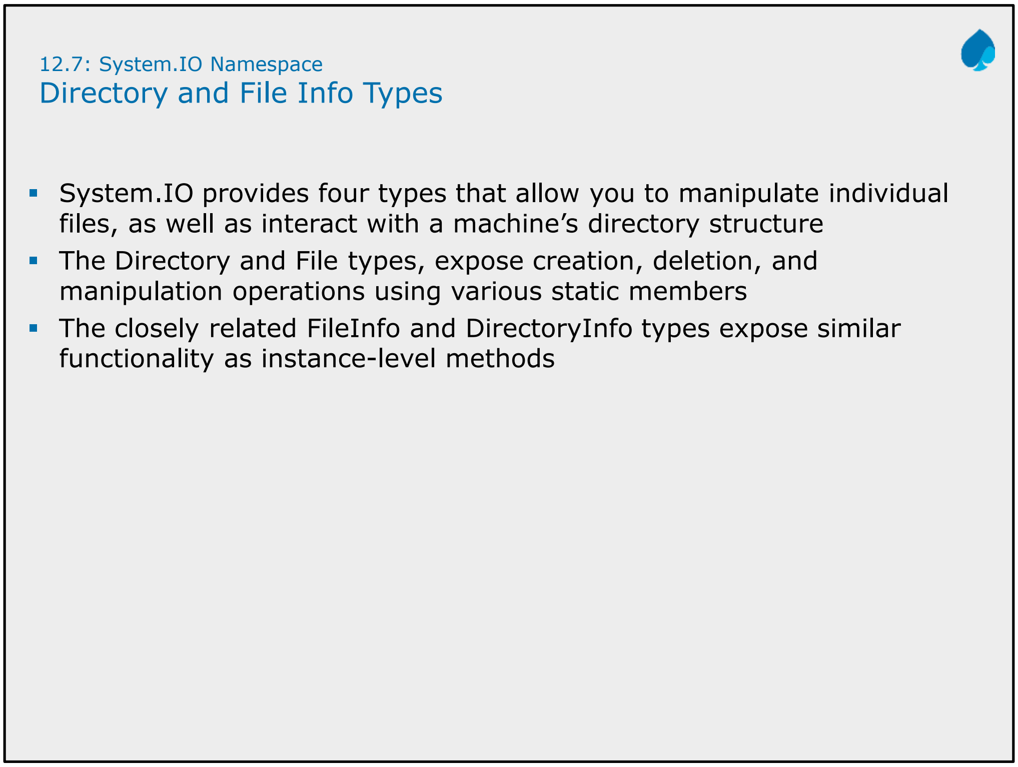
System.IO.TextWriter and Its Derived Classes

- TextWriter :
  - Is the abstract base class for StreamWriter and StringWriter objects
  - While the implementations of the abstract Stream class are designed for byte input and output, the implementations of TextWriter are designed for Unicode character input
- StreamWriter :
  - Writes characters to Streams, using Encoding to convert characters to bytes
- StringWriter :
  - Writes characters to Strings. StringWriter allows you to treat Strings with the same API, so your output can be either a Stream in any encoding or a String



Demo

- Demo on all above IO classes



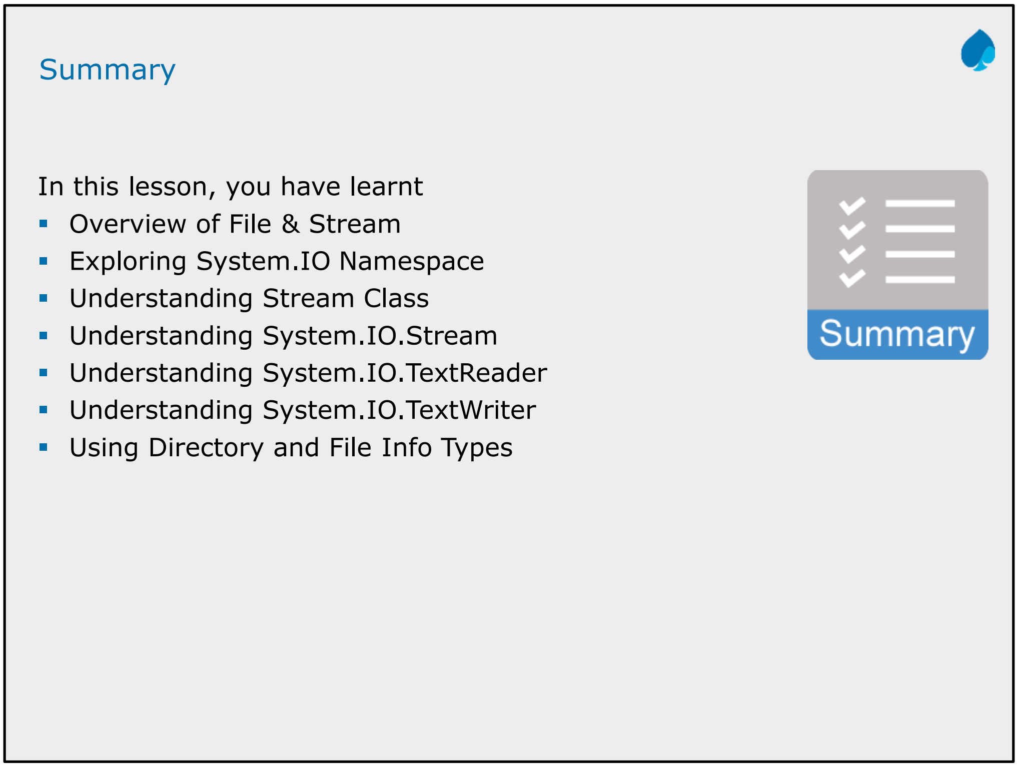
12.7: System.IO Namespace

Directory and File Info Types

- System.IO provides four types that allow you to manipulate individual files, as well as interact with a machine’s directory structure
- The Directory and File types, expose creation, deletion, and manipulation operations using various static members
- The closely related FileInfo and DirectoryInfo types expose similar functionality as instance-level methods

Demo

- Demo on Using File, FileInfo, Directory & DirectoryInfo in C#

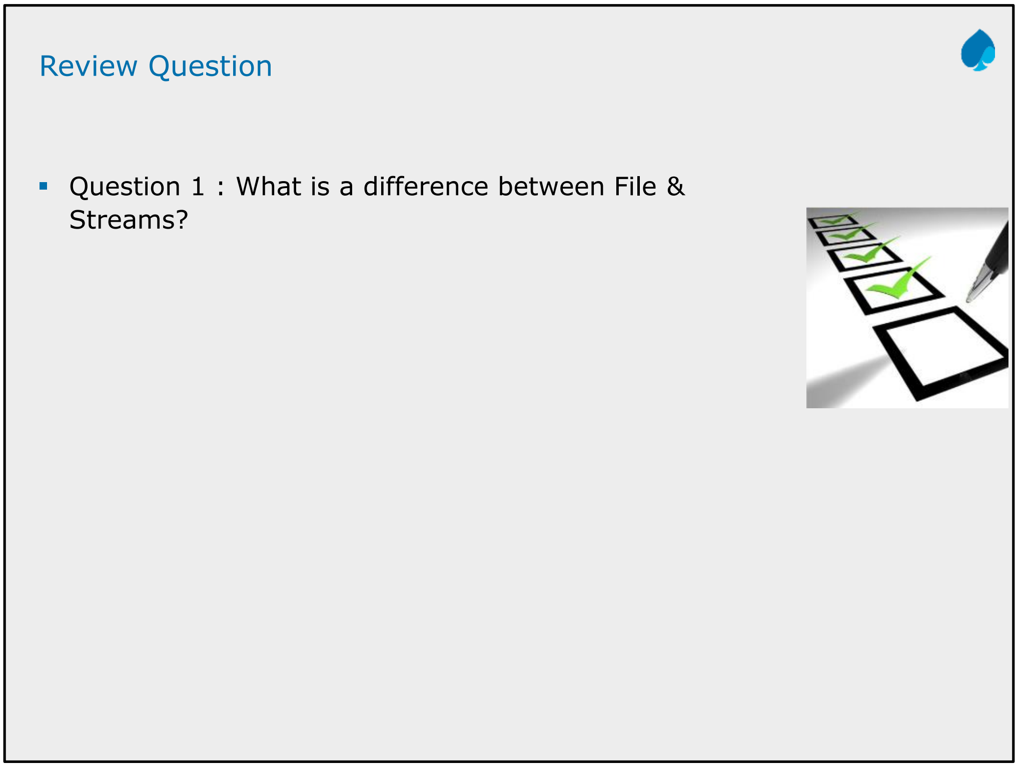


Summary

In this lesson, you have learnt

- Overview of File & Stream
- Exploring System.IO Namespace
- Understanding Stream Class
- Understanding System.IO.Stream
- Understanding System.IO.TextReader
- Understanding System.IO.TextWriter
- Using Directory and File Info Types

Add the notes here.



Review Question

- Question 1 : What is a difference between File & Streams?