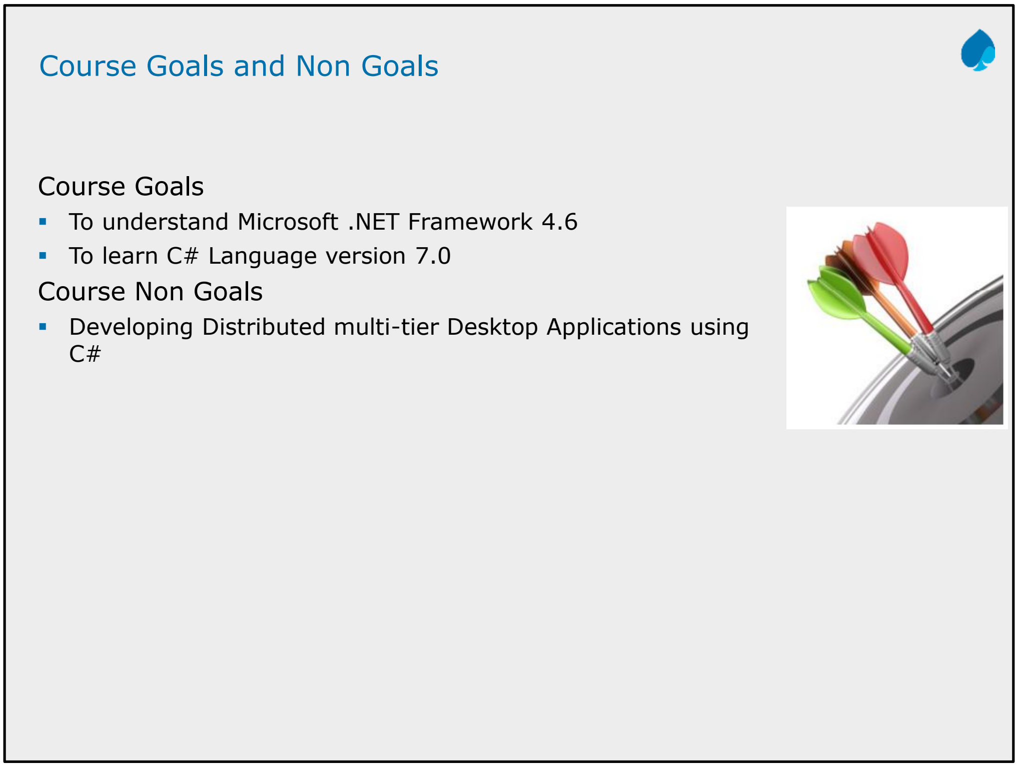
C# 7.0

Lesson 00:



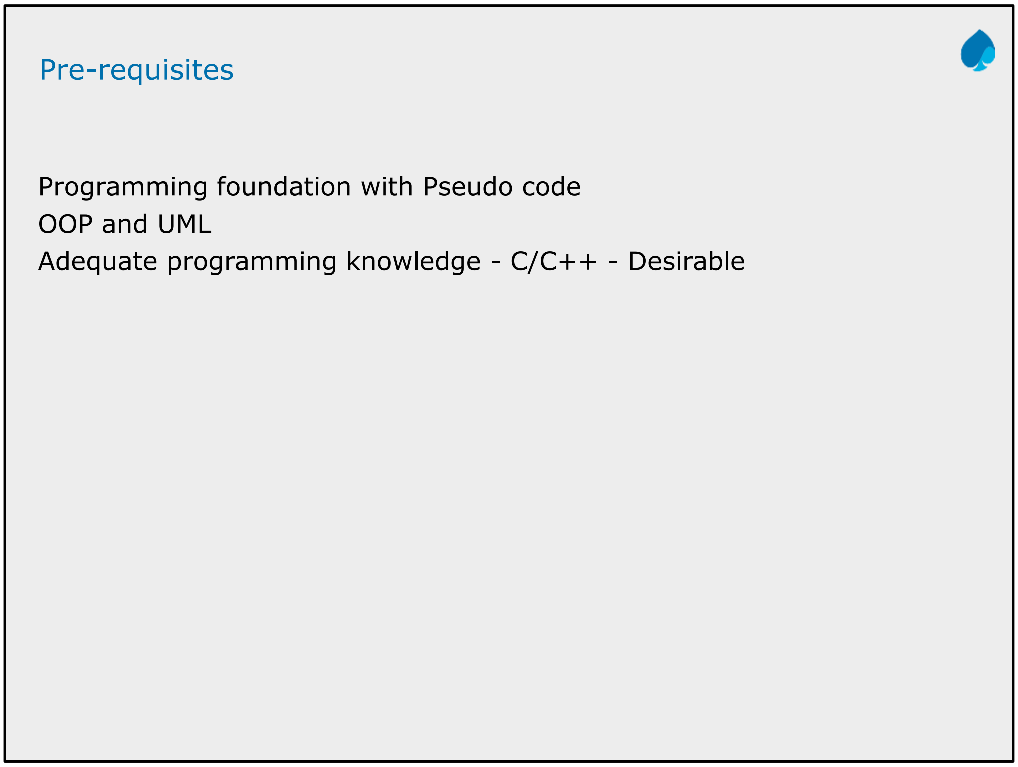
Course Goals and Non Goals

Course Goals

- To understand Microsoft .NET Framework 4.6
- To learn C# Language version 7.0

Course Non Goals

- Developing Distributed multi-tier Desktop Applications using C#

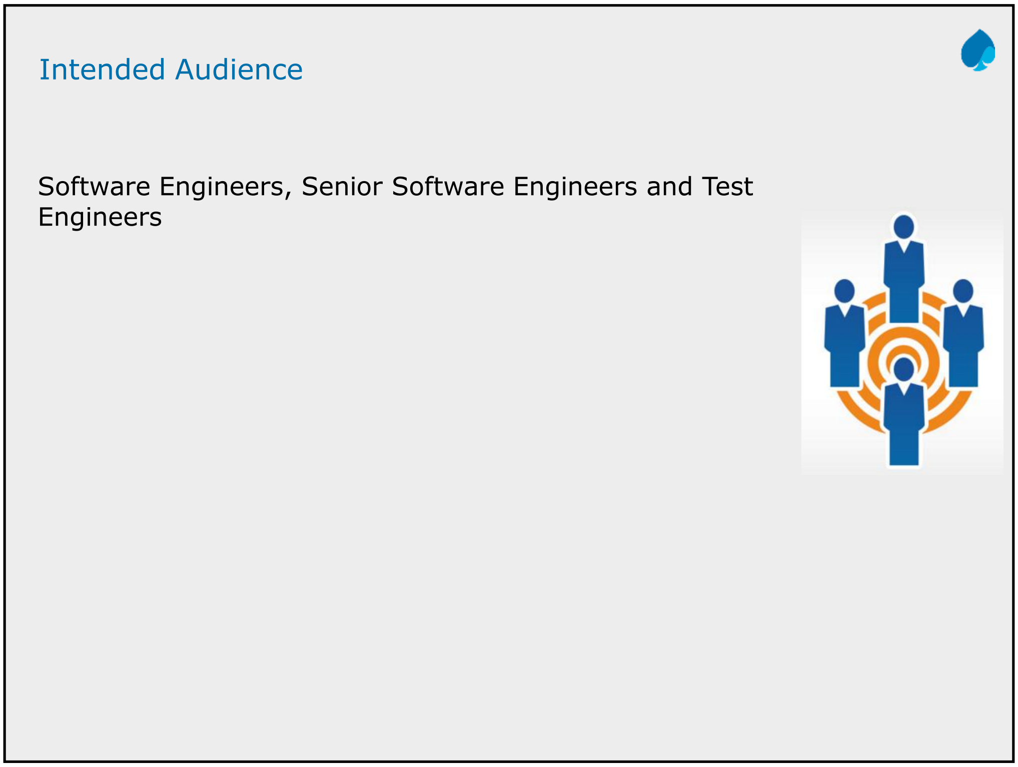


Pre-requisites

Programming foundation with Pseudo code

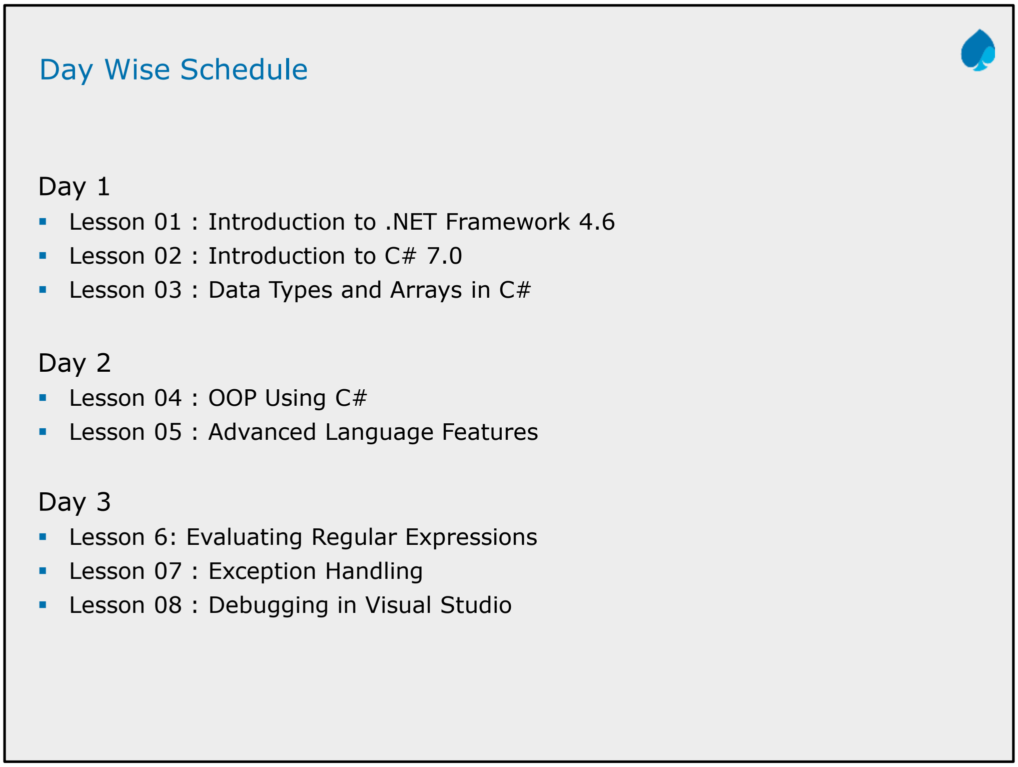
OOP and UML

Adequate programming knowledge - C/C++ - Desirable



Intended Audience

Software Engineers, Senior Software Engineers and Test Engineers



Day Wise Schedule

Day 1

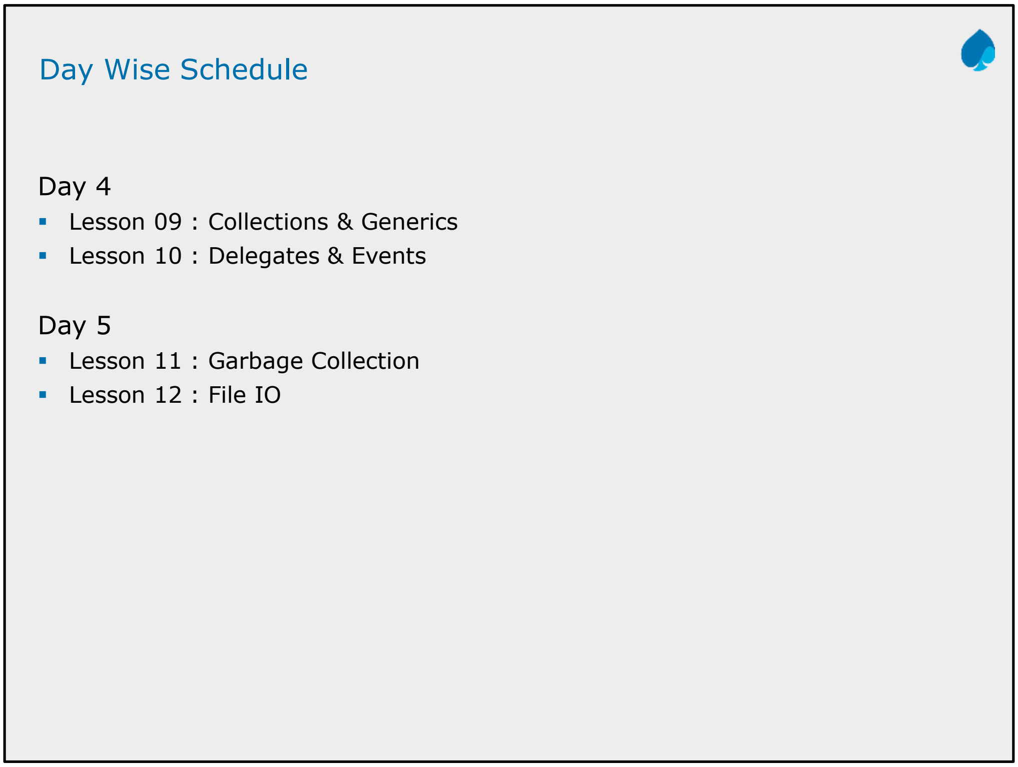
- Lesson 01 : Introduction to .NET Framework 4.6
- Lesson 02 : Introduction to C# 7.0
- Lesson 03 : Data Types and Arrays in C#

Day 2

- Lesson 04 : OOP Using C#
- Lesson 05 : Advanced Language Features

Day 3

- Lesson 6: Evaluating Regular Expressions
- Lesson 07 : Exception Handling
- Lesson 08 : Debugging in Visual Studio



Day Wise Schedule

Day 4

- Lesson 09 : Collections & Generics
- Lesson 10 : Delegates & Events

Day 5

- Lesson 11 : Garbage Collection
- Lesson 12 : File IO

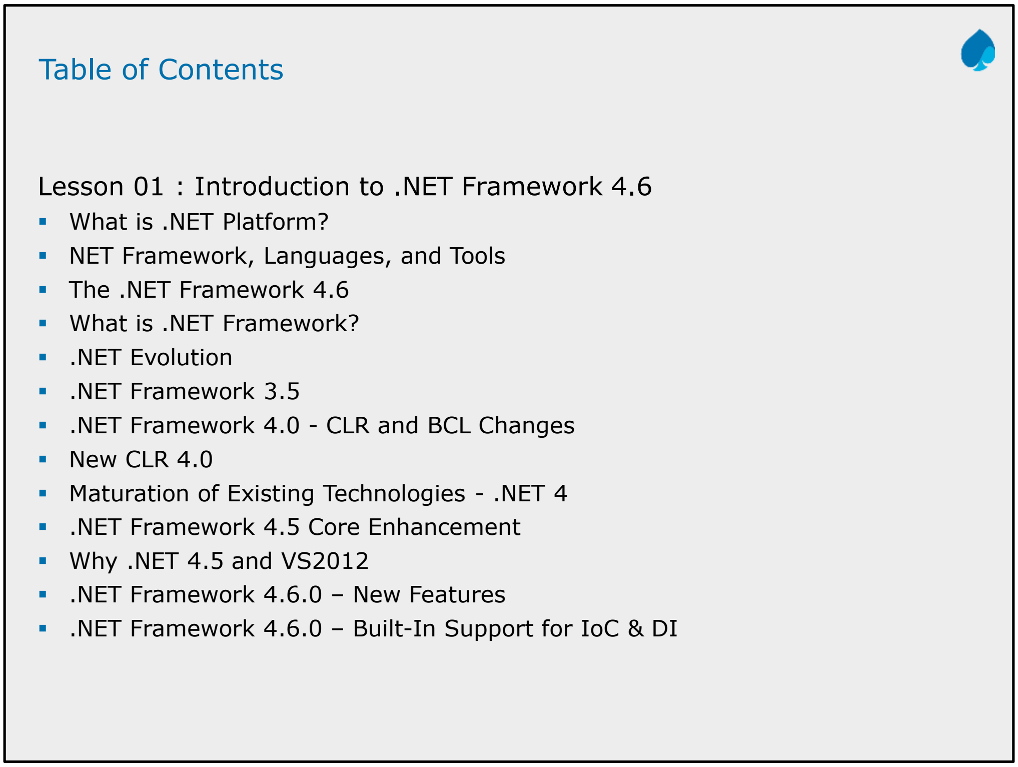


Table of Contents

Lesson 01 : Introduction to .NET Framework 4.6

- What is .NET Platform?
- NET Framework, Languages, and Tools
- The .NET Framework 4.6
- What is .NET Framework?
- .NET Evolution
- .NET Framework 3.5
- .NET Framework 4.0 - CLR and BCL Changes
- New CLR 4.0
- Maturation of Existing Technologies - .NET 4
- .NET Framework 4.5 Core Enhancement
- Why .NET 4.5 and VS2012
- .NET Framework 4.6.0 – New Features
- .NET Framework 4.6.0 – Built-In Support for IoC & DI

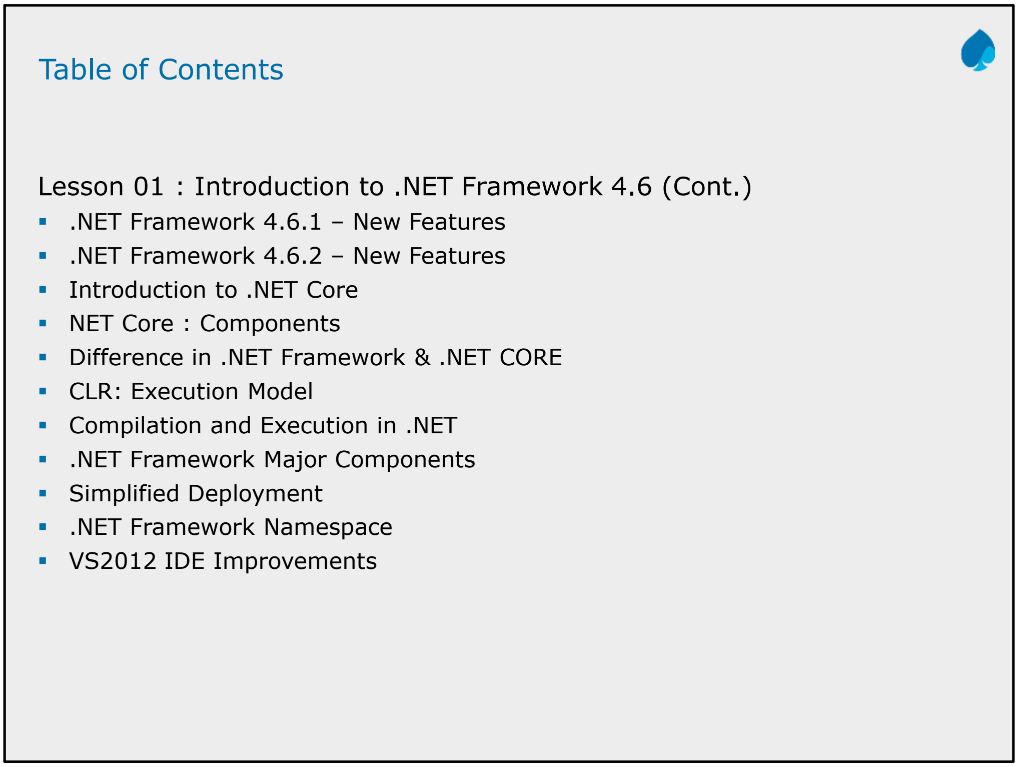


Table of Contents

Lesson 01 : Introduction to .NET Framework 4.6 (Cont.)

- .NET Framework 4.6.1 – New Features
- .NET Framework 4.6.2 – New Features
- Introduction to .NET Core
- NET Core : Components
- Difference in .NET Framework & .NET CORE
- CLR: Execution Model
- Compilation and Execution in .NET
- .NET Framework Major Components
- Simplified Deployment
- .NET Framework Namespace
- VS2012 IDE Improvements

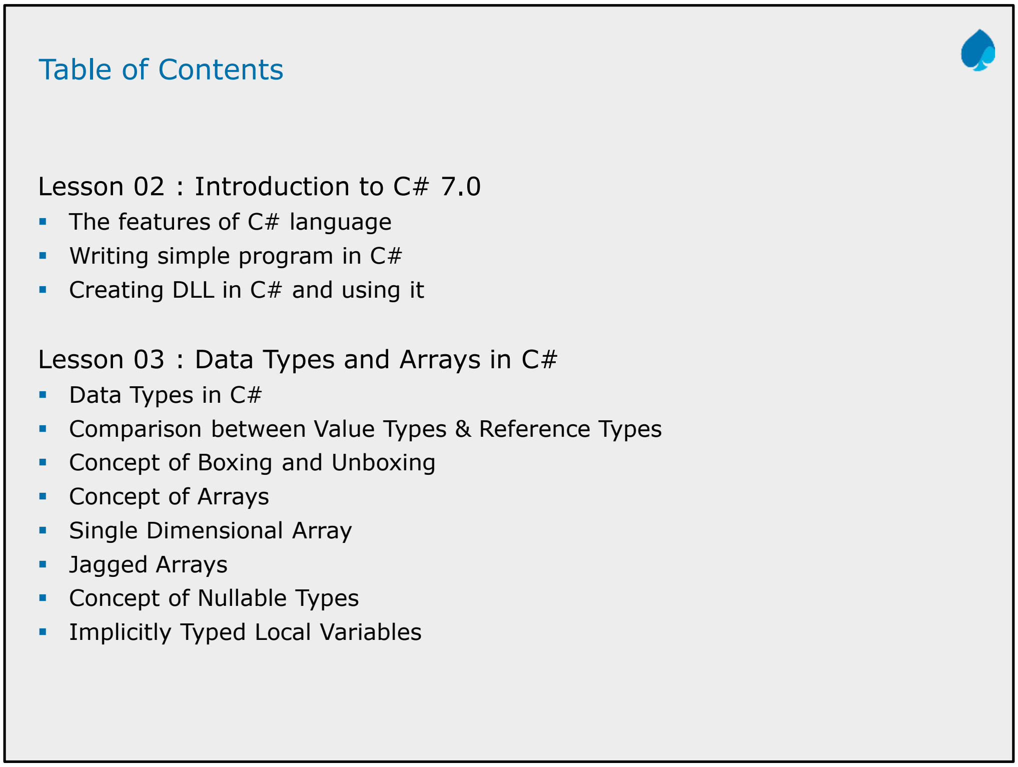


Table of Contents

Lesson 02 : Introduction to C# 7.0

- The features of C# language
- Writing simple program in C#
- Creating DLL in C# and using it

Lesson 03 : Data Types and Arrays in C#

- Data Types in C#
- Comparison between Value Types & Reference Types
- Concept of Boxing and Unboxing
- Concept of Arrays
- Single Dimensional Array
- Jagged Arrays
- Concept of Nullable Types
- Implicitly Typed Local Variables

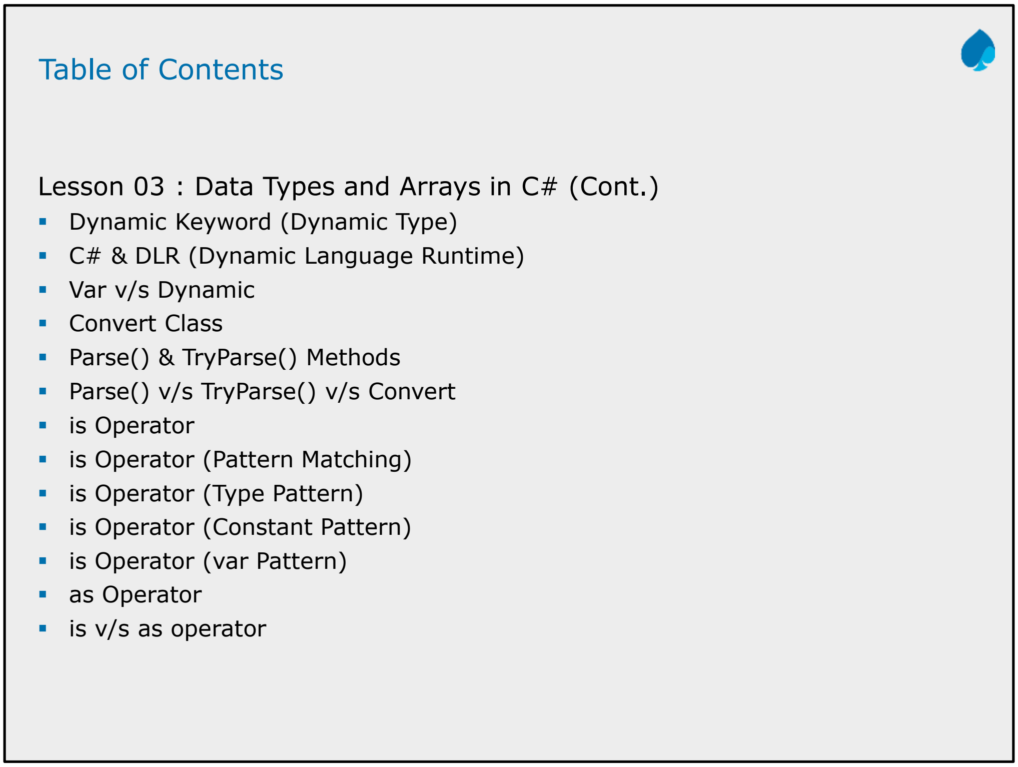


Table of Contents

Lesson 03 : Data Types and Arrays in C# (Cont.)

- Dynamic Keyword (Dynamic Type)
- C# & DLR (Dynamic Language Runtime)
- Var v/s Dynamic
- Convert Class
- Parse() & TryParse() Methods
- Parse() v/s TryParse() v/s Convert
- is Operator
- is Operator (Pattern Matching)
- is Operator (Type Pattern)
- is Operator (Constant Pattern)
- is Operator (var Pattern)
- as Operator
- is v/s as operator

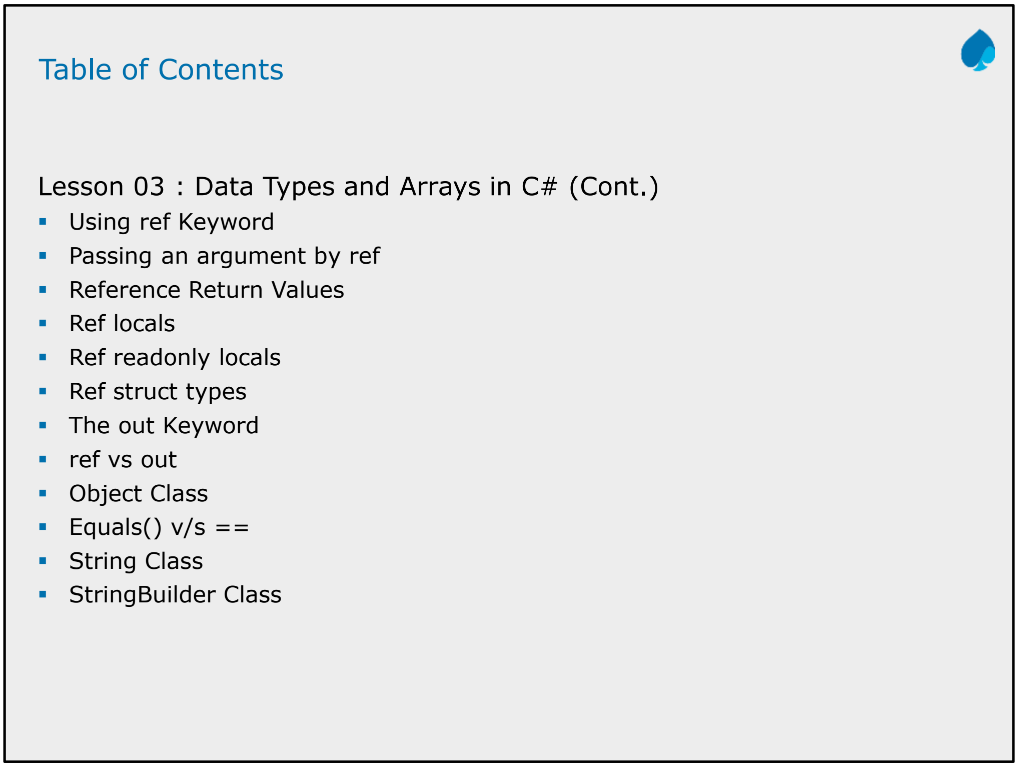


Table of Contents

Lesson 03 : Data Types and Arrays in C# (Cont.)

- Using ref Keyword
- Passing an argument by ref
- Reference Return Values
- Ref locals
- Ref readonly locals
- Ref struct types
- The out Keyword
- ref vs out
- Object Class
- Equals() v/s ==
- String Class
- StringBuilder Class

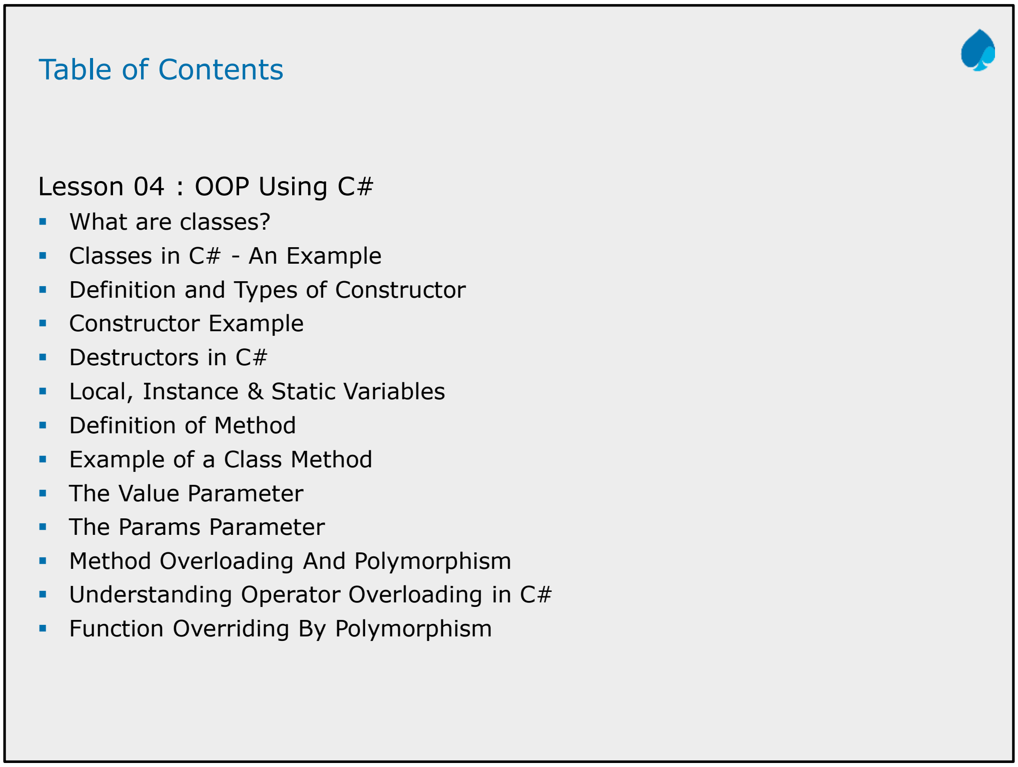


Table of Contents

Lesson 04 : OOP Using C#

- What are classes?
- Classes in C# - An Example
- Definition and Types of Constructor
- Constructor Example
- Destructors in C#
- Local, Instance & Static Variables
- Definition of Method
- Example of a Class Method
- The Value Parameter
- The Params Parameter
- Method Overloading And Polymorphism
- Understanding Operator Overloading in C#
- Function Overriding By Polymorphism

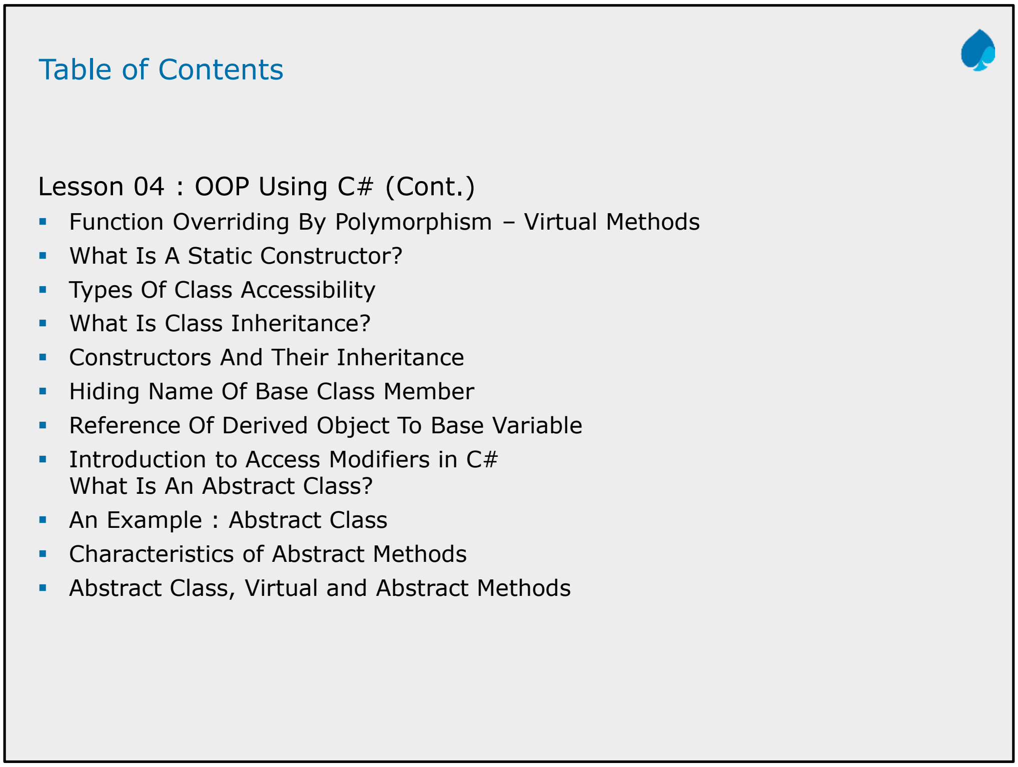


Table of Contents

Lesson 04 : OOP Using C# (Cont.)

- Function Overriding By Polymorphism – Virtual Methods
- What Is A Static Constructor?
- Types Of Class Accessibility
- What Is Class Inheritance?
- Constructors And Their Inheritance
- Hiding Name Of Base Class Member
- Reference Of Derived Object To Base Variable
- Introduction to Access Modifiers in C#
  What Is An Abstract Class?
- An Example : Abstract Class
- Characteristics of Abstract Methods
- Abstract Class, Virtual and Abstract Methods

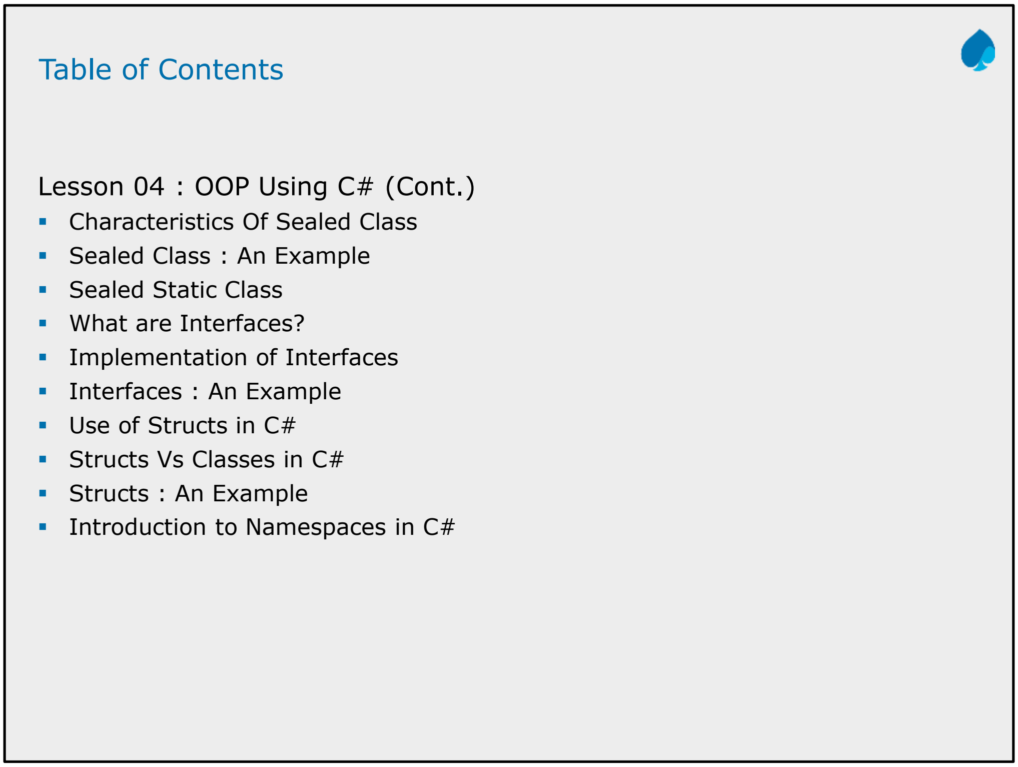


Table of Contents

Lesson 04 : OOP Using C# (Cont.)

- Characteristics Of Sealed Class
- Sealed Class : An Example
- Sealed Static Class
- What are Interfaces?
- Implementation of Interfaces
- Interfaces : An Example
- Use of Structs in C#
- Structs Vs Classes in C#
- Structs : An Example
- Introduction to Namespaces in C#

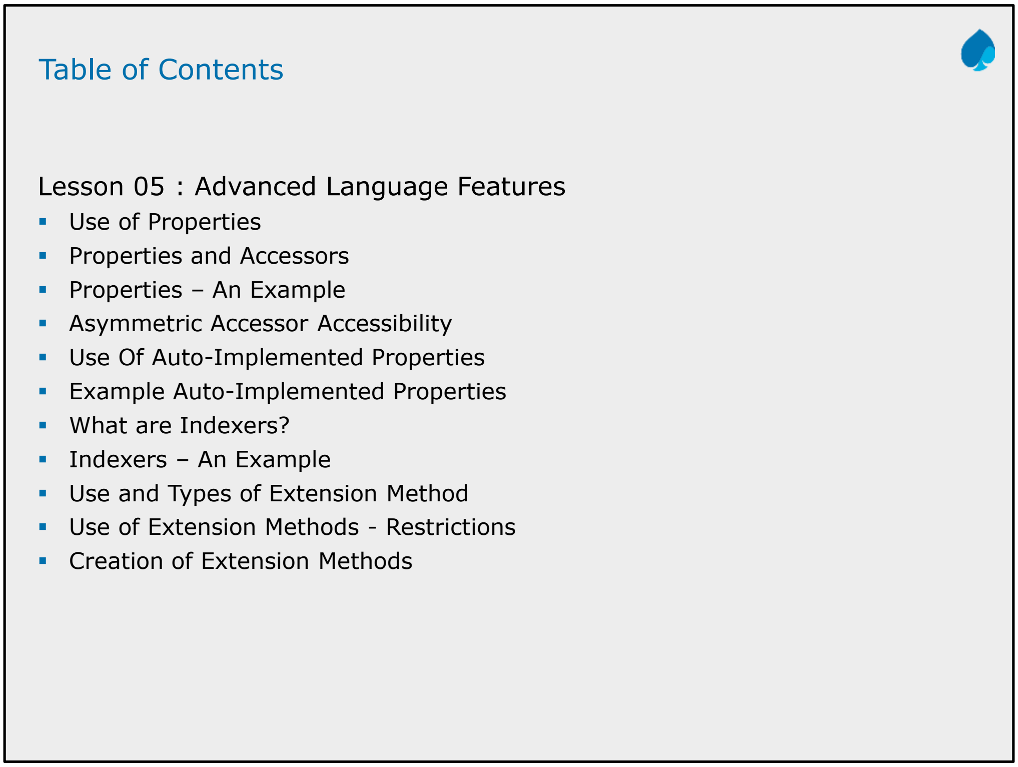


Table of Contents

Lesson 05 : Advanced Language Features

- Use of Properties
- Properties and Accessors
- Properties – An Example
- Asymmetric Accessor Accessibility
- Use Of Auto-Implemented Properties
- Example Auto-Implemented Properties
- What are Indexers?
- Indexers – An Example
- Use and Types of Extension Method
- Use of Extension Methods - Restrictions
- Creation of Extension Methods

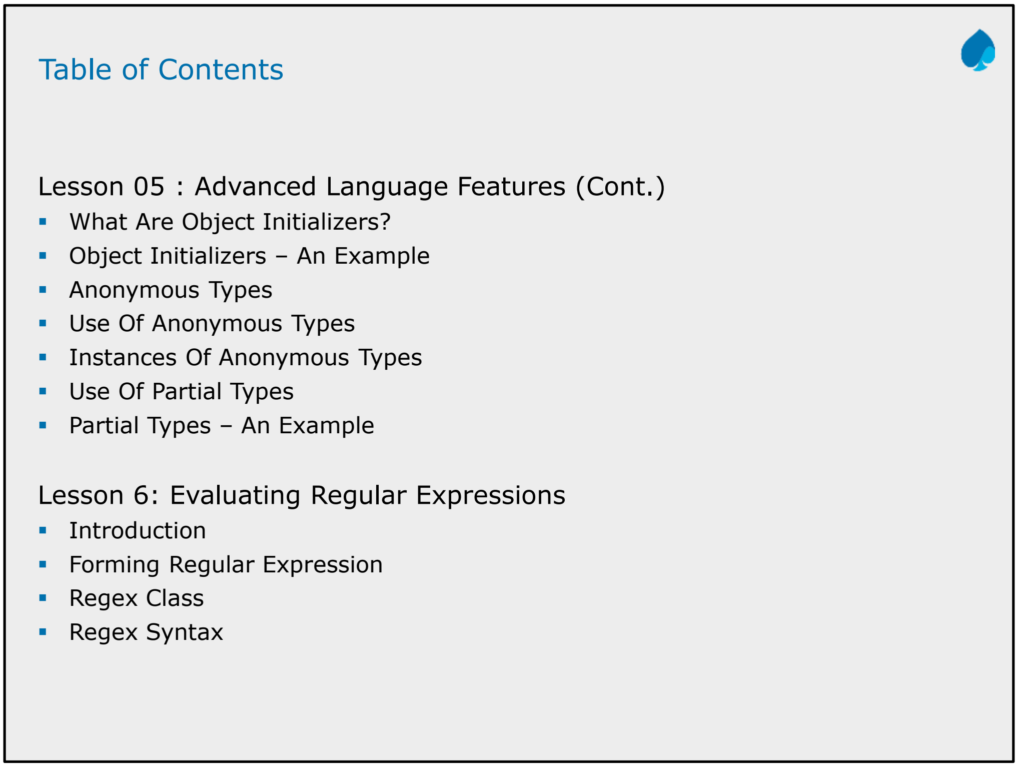


Table of Contents

Lesson 05 : Advanced Language Features (Cont.)

- What Are Object Initializers?
- Object Initializers – An Example
- Anonymous Types
- Use Of Anonymous Types
- Instances Of Anonymous Types
- Use Of Partial Types
- Partial Types – An Example

Lesson 6: Evaluating Regular Expressions

- Introduction
- Forming Regular Expression
- Regex Class
- Regex Syntax

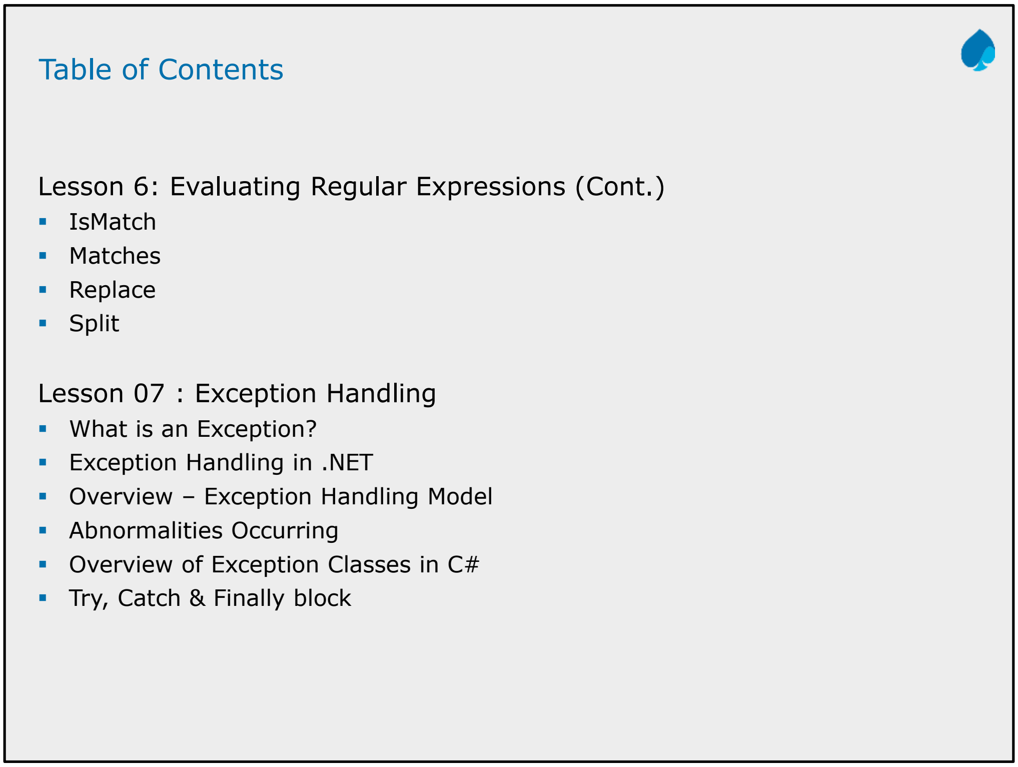


Table of Contents

Lesson 6: Evaluating Regular Expressions (Cont.)

- IsMatch
- Matches
- Replace
- Split

Lesson 07 : Exception Handling

- What is an Exception?
- Exception Handling in .NET
- Overview – Exception Handling Model
- Abnormalities Occurring
- Overview of Exception Classes in C#
- Try, Catch & Finally block

Table of Contents

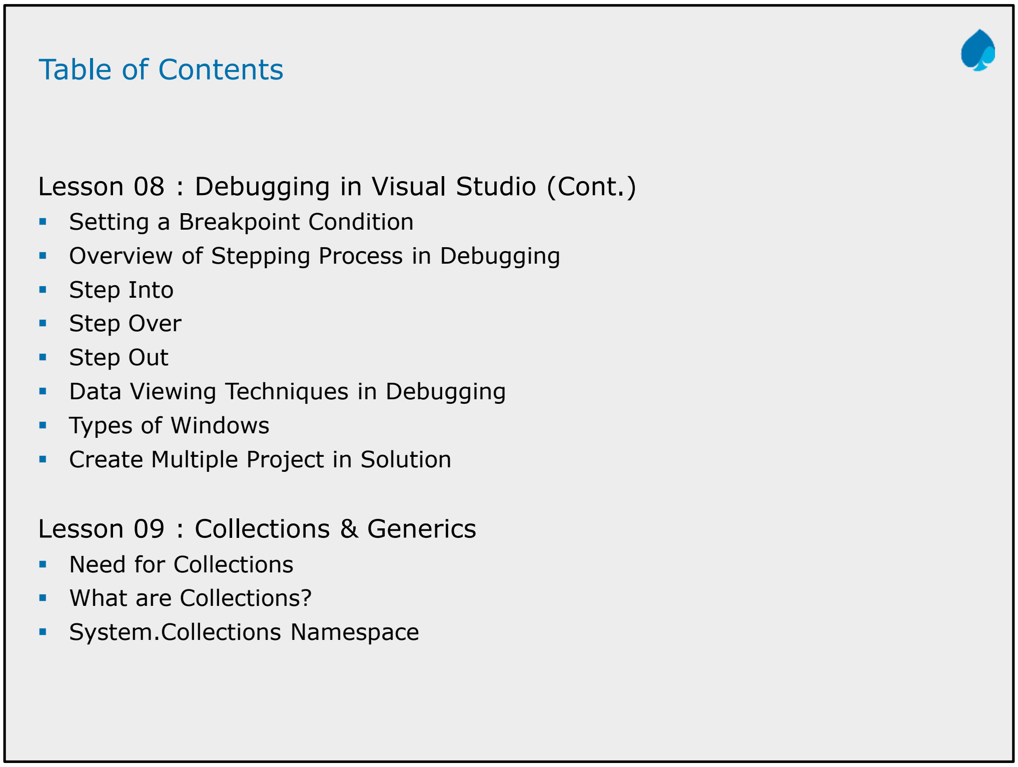


Table of Contents

Lesson 08 : Debugging in Visual Studio (Cont.)

- Setting a Breakpoint Condition
- Overview of Stepping Process in Debugging
- Step Into
- Step Over
- Step Out
- Data Viewing Techniques in Debugging
- Types of Windows
- Create Multiple Project in Solution

Lesson 09 : Collections & Generics

- Need for Collections
- What are Collections?
- System.Collections Namespace

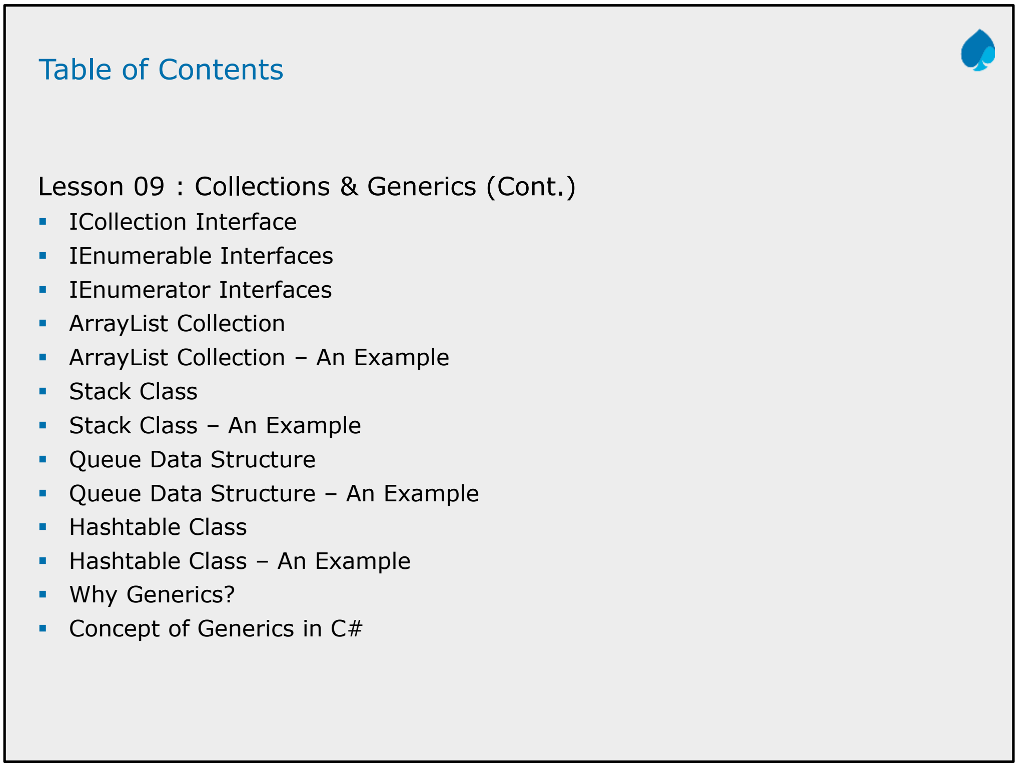


Table of Contents

Lesson 09 : Collections & Generics (Cont.)

- ICollection Interface
- IEnumerable Interfaces
- IEnumerator Interfaces
- ArrayList Collection
- ArrayList Collection – An Example
- Stack Class
- Stack Class – An Example
- Queue Data Structure
- Queue Data Structure – An Example
- Hashtable Class
- Hashtable Class – An Example
- Why Generics?
- Concept of Generics in C#

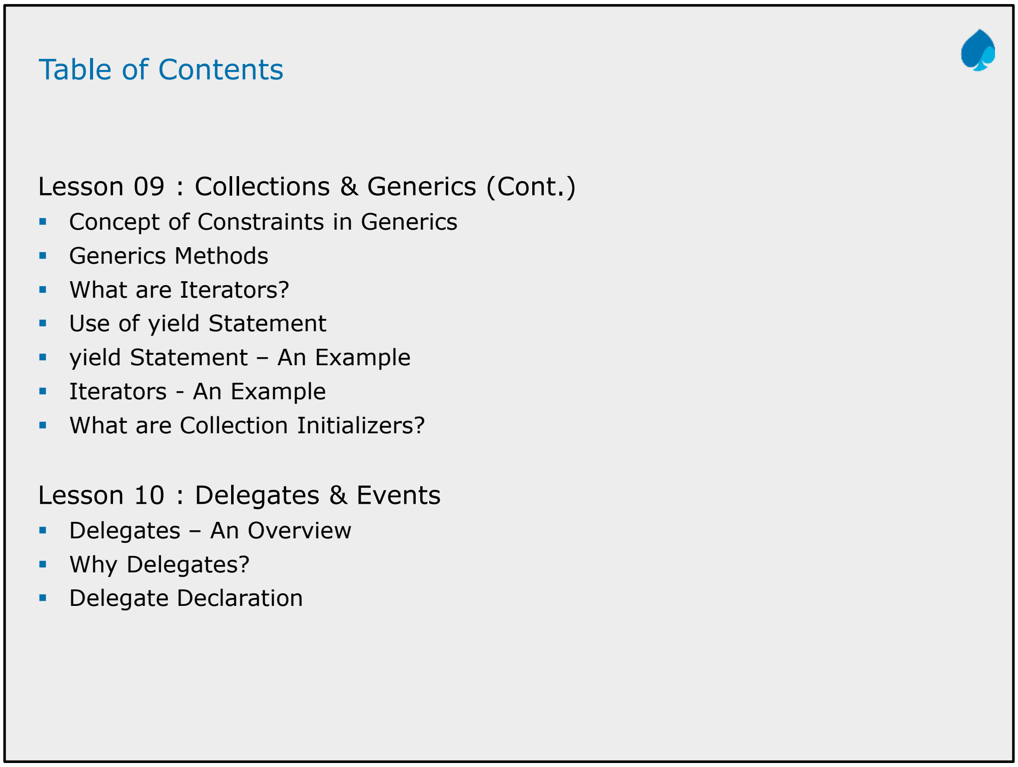


Table of Contents

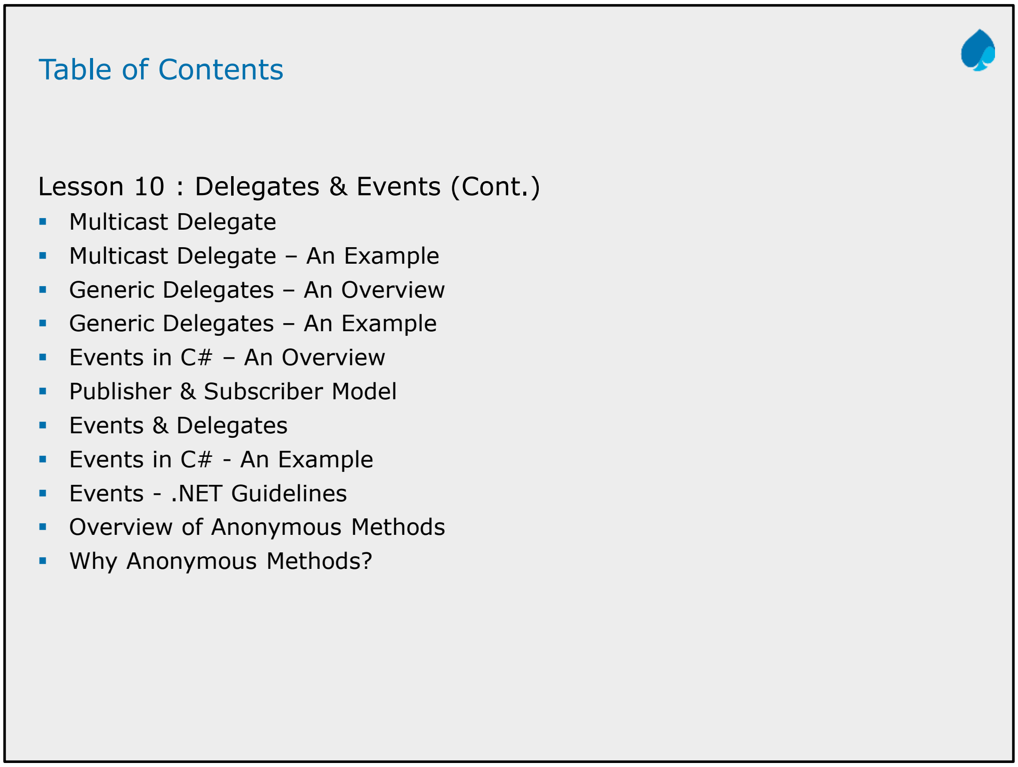


Table of Contents

Lesson 10 : Delegates & Events (Cont.)

- Multicast Delegate
- Multicast Delegate – An Example
- Generic Delegates – An Overview
- Generic Delegates – An Example
- Events in C# – An Overview
- Publisher & Subscriber Model
- Events & Delegates
- Events in C# - An Example
- Events - .NET Guidelines
- Overview of Anonymous Methods
- Why Anonymous Methods?

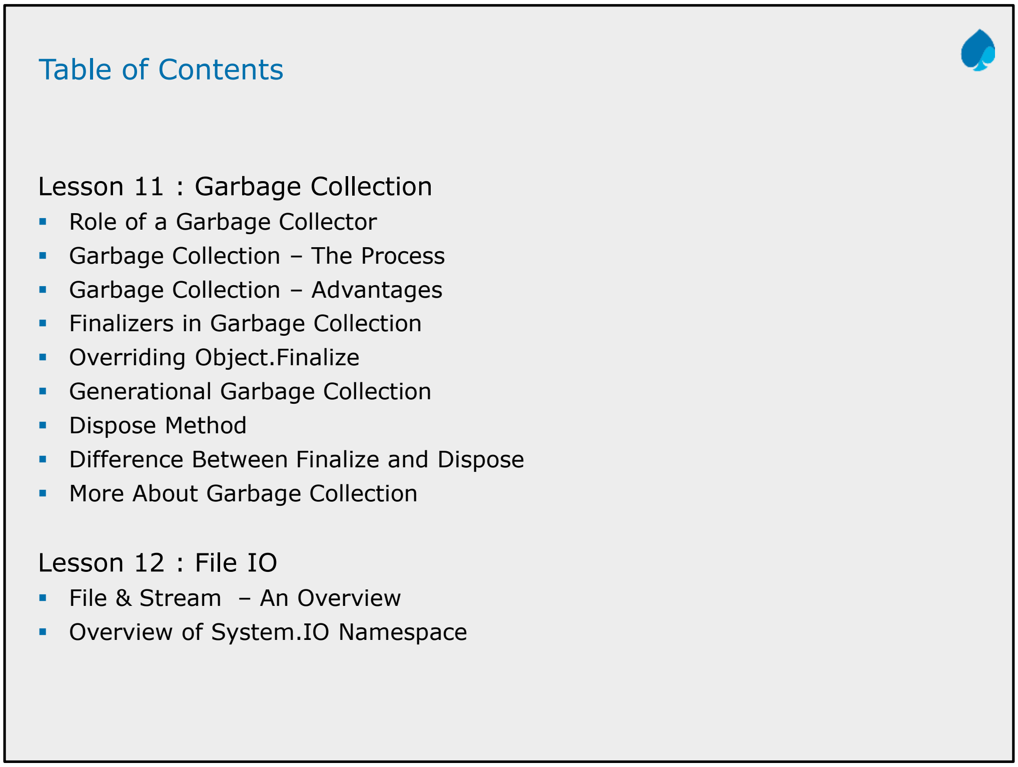


Table of Contents

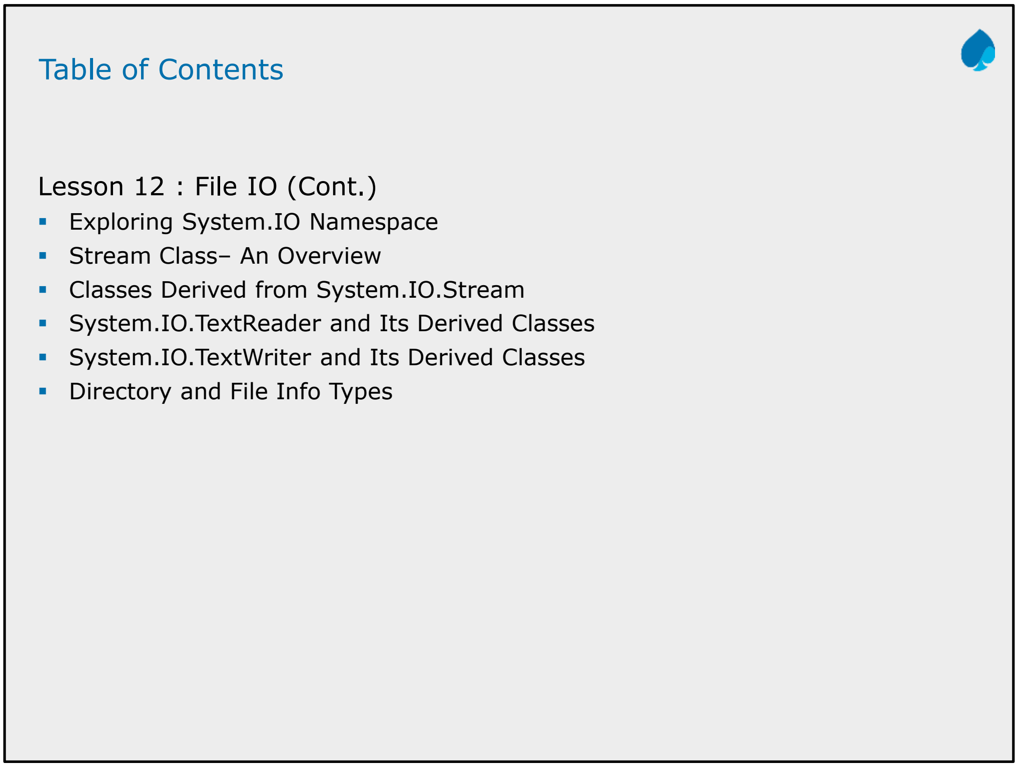
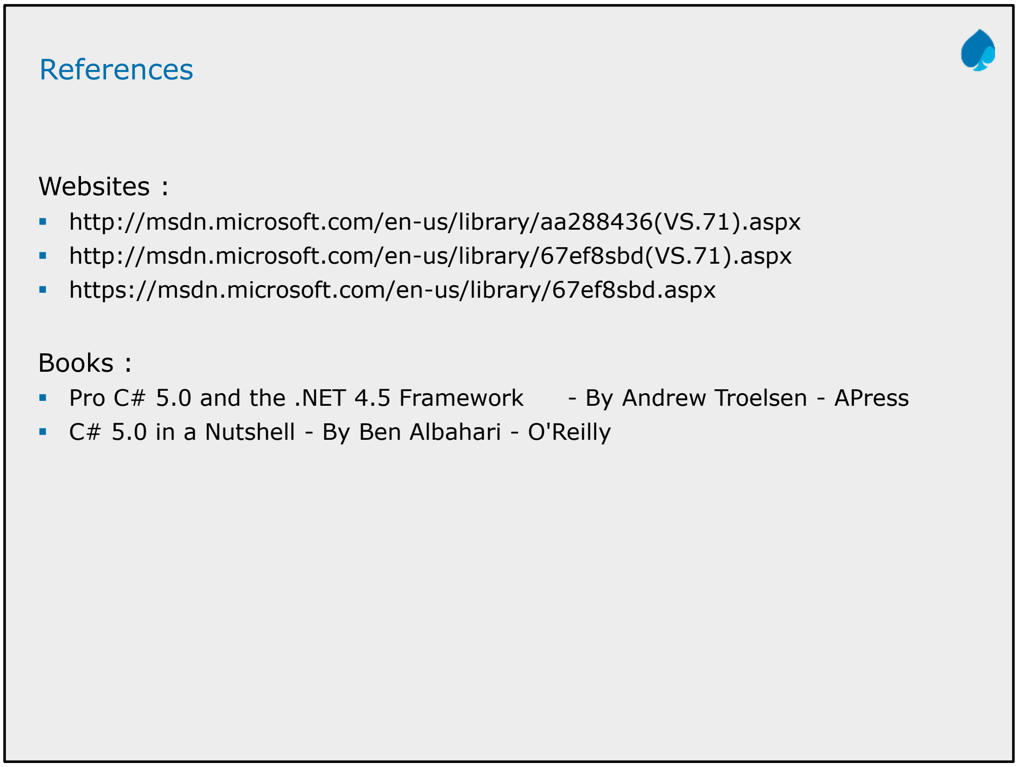


Table of Contents

Lesson 12 : File IO (Cont.)

- 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



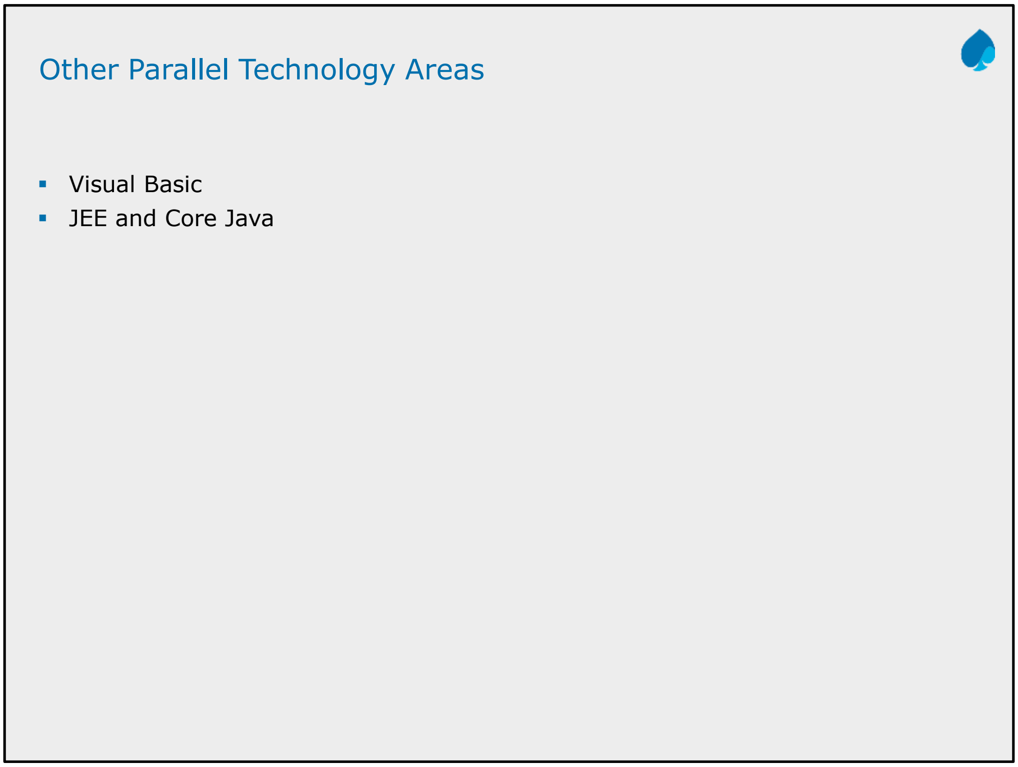
References

Websites :

- http://msdn.microsoft.com/en-us/library/aa288436(VS.71).aspx
- http://msdn.microsoft.com/en-us/library/67ef8sbd(VS.71).aspx
- https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx

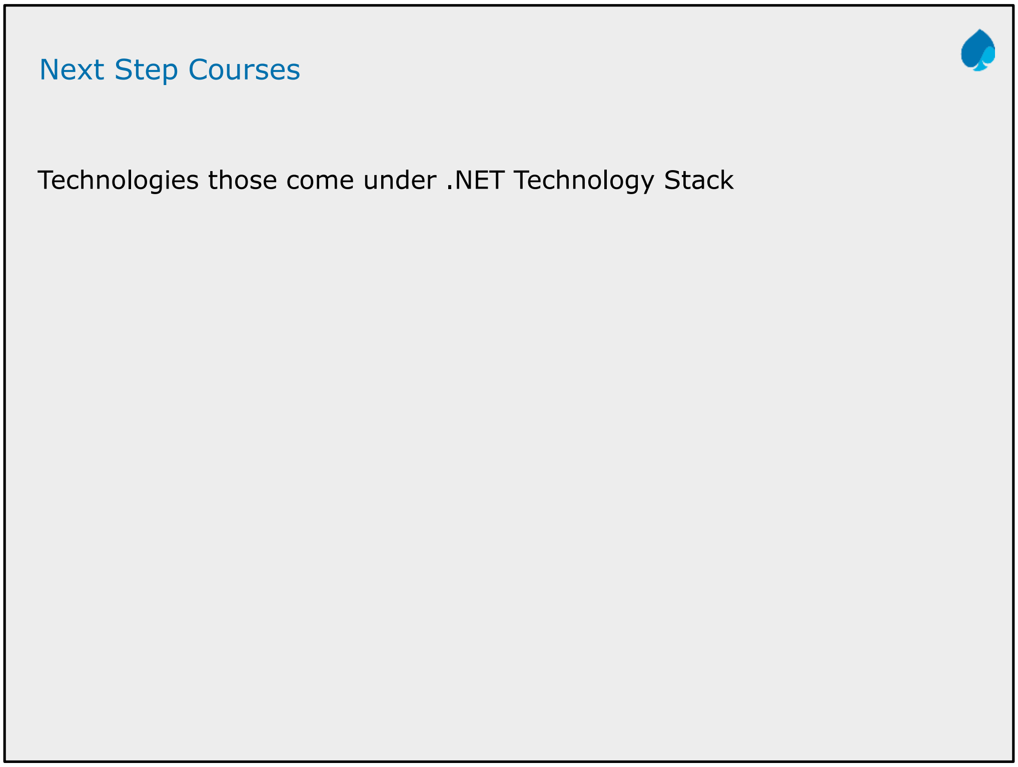
Books :

- Pro C# 5.0 and the .NET 4.5 Framework - By Andrew Troelsen - APress
- C# 5.0 in a Nutshell - By Ben Albahari - O'Reilly



Other Parallel Technology Areas

- Visual Basic
- JEE and Core Java



Next Step Courses

Technologies those come under .NET Technology Stack