

C#

Detailed Study of C#

Yash Lathiya

|  |  |  |  |
| --- | --- | --- | --- |
| Sr No. | Topic | Date | Page No. |
|  |  |  |  |
| 1 | Introduction to C# | 17/08/2023 | 2 |
| 2 | Structure of C# | 18/08/2023 | 3 |
| 3 | Working with code files, projects & Solutions | 18/08/2023 | 4 |
| 4 | Datatypes & Variables | 18/08/2023 | 5 |
| 5 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Introduction to C#

* Object-Oriented Programming language
* Type-safe programming language
* Similar to C, C++, Java, Javascript
* Automatic garbage collection
* Exception handling
* Nullable types are assigned to variable which are not assigned values
* Support of asynchronous operations.
* Supports generic methods & types.

Execution of C#

* C# programs run on .NET.
* CLR (Common Language Runtime) is a virtual library which is implementation of CLI (Common Language Infrastructure)..
* Source code written in C# is compiled to IL (Intermediate Language) that conforms CLI specifications.
* IL code and other resources (bitmaps, strings,..) are stored in assembly with .dll extension.
* When C# is executed, assembly is loaded in CLR & CLR performs compilation (JIT – Just in Time) to convert IL code to native machine instructions.
* CLR also provides automatic garbage collection, exception handling, resource management, etc..
* IL code (Compiled code of C#) can interact with .NET, C++ and other languages for which CTS (Common Type Specifications) is allowed.

Structure of C#

// A skeleton of a C# program

using System; //System is namespace & Console is class of that namespace

// C# program starts here:

Console.WriteLine("Hello world!");

namespace YourNamespace //It can contain class, struct, interface, delegate, enum or nested namespace

{

class YourClass

{

}

struct YourStruct // encapsulate data and related functionality

{

}

interface IYourInterface //Can’t be instanstiated directly, It’s members are implemented by class or structs implements interface

{

}

delegate int YourDelegate();

enum YourEnum // constants which has integral numeric type

{

}

namespace YourNestedNamespace

{

struct YourStruct

{

}

}

}

Working with code files, projects & solutions



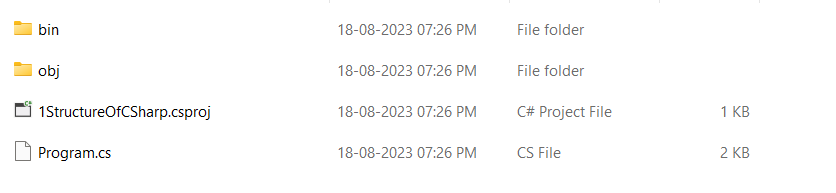
(Main Folder)

(Contains Solution Folder)



(Configuration file..& contains info about compilation & etc.)

(bin folder contains binary [machine-readable] data of the project)



Project File C# file

Datatypes & Variables

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Size** | **Description** |
| int | 4 bytes | Stores whole numbers from -2,147,483,648 to 2,147,483,647 |
| long | 8 bytes | Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| float | 4 bytes | Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits |
| double | 8 bytes | Stores fractional numbers. Sufficient for storing 15 decimal digits |
| bool | 1 bit | Stores true or false values |
| char | 2 bytes | Stores a single character/letter, surrounded by single quotes |
| string | 2 bytes per character | Stores a sequence of characters, surrounded by double quotes |

Type Casting

* **Implicit Casting** (automatically) - converting a smaller type to a larger type size  
  char -> int -> long -> float -> double
* **Explicit Casting** (manually) - converting a larger type to a smaller size type  
  double -> float -> long -> int -> char