C# - <https://www.tutorialspoint.com/csharp>

**Overview**:

* General-purpose, object oriented programming language developed by Microsoft
* Designed for Common Language Infrastructure (CLI)
* Consists of the Executable Code & the Runtime Environment that allows use of various high-level languages on different computer platforms and architectures

Strong Features:

* Automatic Garbage Collection
* Conditional Compilation
* Simple Multithreading
* LINQ & Lambda Expressions
* Integration w/ Windows

**Environment**:

.NET Framework (relation to C#):

* Applications written w/ .NET:
* Multi-platform applications – all .NET languages access the framework & communicate
* Windows Applications, Web Applications, & Web Services
* Components of .NET:
* Common Language Runtime (CLR)
* .NET Framework Class Library
* Common Language Specification, Common Type System
* Windows Forms, Windows Workflow Foundation (WF)
* Windows Presentation Foundation (WPF), Windows Communication Foundation (WCF)
* Metadata & Assemblies, LINQ

IDE’s for C#:

* Visual Studio
* Visual C# Express
* Visual Web Developer
* \*Can also write C# source code files w/ text editor
* compile the code into assemblies using the command-line compiler (part of .NET)

**Program Structure**: [Basic ‘Hello World’ C# Application]

Basic Program:

* Using Statements
* Namespace Declaration
* Class Declarations
* Class Methods
* Class Attributes
* Main Method Declaration
* Statements
* Expressions
* Comments

Compiling & Executing Programs:

* Visual Studio:
* File 🡪 New 🡪 Project 🡪 Visual C# 🡪 Windows 🡪 Console Application
* Command-Line:
* Write code in text editor 🡪 save as .cs file 🡪 command prompt 🡪

Basic Syntax:

* Using – for including namespaces
* Class – for declaring a class
* Comments - /\* \*/
* Member Variables – attributes/ data members of a class used for storing data
* Member Functions – set of statements that performs a specific task
* Identifiers – names used to identify a class, variable, function, or any other user defined item
* Instantiating a Class – the class containing the Main() method ???

**Data Types**:

Value Types:

* Assigned values directly; directly contain data

Reference Types:

* Do not contain the actual data stored in a var – contains a reference to the variable(s)
* They refer to a memory location
* User-Defined Reference Types – Classes, Interfaces, Delegates (covered later in tutorial)
* Built-In Reference Types – object, dynamic, & string
* \*Object Type:
* Ultimate base class for all data types in C# Common Type System (CTS)
* Object Types can be assigned values of any other types (requires type conversion)
* Boxing (value type 🡪 object type); Unboxing (object type 🡪 value type)
* \*Dynamic Type:
* Similar to object types – can store any type of value
* Different from object types – type checking takes place at runtime (vs compile time)
* \*String Type:
* Allows you to assign any string value to a variable
* Can be assigned using “” OR @ “”

Pointer Types:

* These variables store the memory address of another type (see Unsafe Codes)

Type Conversion:

* AKA Type Casting – converting one type of data to another type
* Implicit Type Conversion –
* Performed by C# in a type-safe manner
* Conversions from derived classes to base classes, etc.
* Explicit Type Conversion –
* Performed explicitly by users using the predefined functions
* Requires a Cast Operator
* C# provides built-in type conversion (toString(), toInt(), etc.)

**Declaring Variables** –

Lexicon: group of words / tokens that the language understands

Syntax: the rules / grammar for using those words (tokens) together