A tour of the C# Language

At its core, C# is an ***object-oriented*** language. You define types and their behavior.

Several C# features help create robust and durable applications:

* **Garbage collection** automatically reclaims memory occupied by unreachable unused objects.
* **Nullable types** guard against variables that don't refer to allocated objects.
* **Exception handling** provides a structured and extensible approach to error detection and recovery.
* **Lambda expressions** support functional programming techniques.
* **Language Integrated Query (LINQ)** syntax creates a common pattern for working with data from any source.
* Language support for **asynchronous operations** provides syntax for building distributed systems.
* C# has a **unified type system**. All C# types, including primitive types such as int and double, inherit from a single root object type.

## .NET architecture

C# programs run on .NET, a virtual execution system called the common language runtime (CLR) and a set of class libraries.

The CLR provides other services related to automatic garbage collection, exception handling, and resource management.

## Types and variables

A *type* defines the structure and behavior of any data in C#. The declaration of a type may include its members, base type, interfaces it implements, and operations permitted for that type.

A *variable* is a label that refers to an instance of a specific type.