



# Programming Language - C#

## Introduction to C#



# Agenda



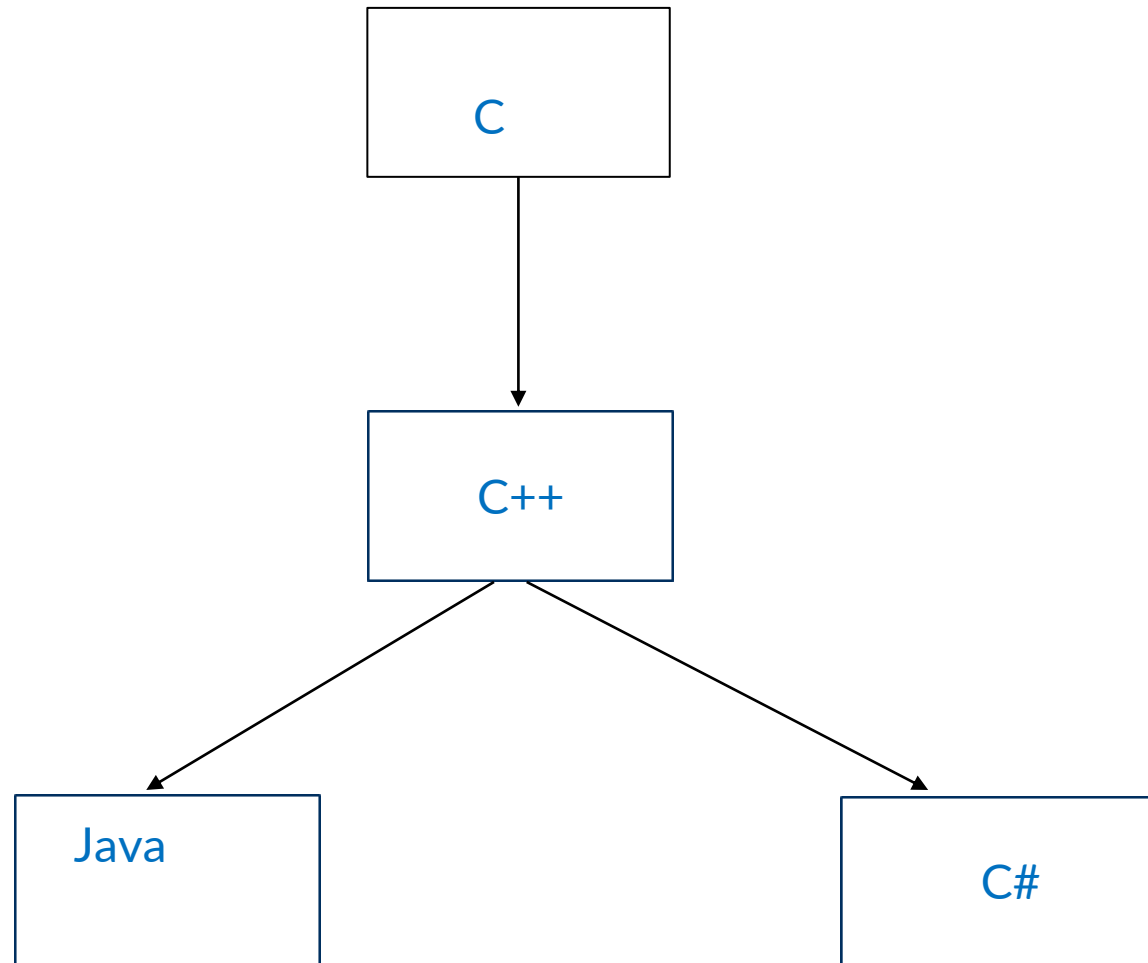
- What is C#?
- C# Version History
- Key Features of C#
- Why Choose C#?
- Introduction to .NET Framework
- Common Language Runtime (CLR)
- Common Type System (CTS)
- Sample Program
- Datatypes
- Operators

# What is C#?



- Definition: C# (pronounced C-sharp) is a modern, object-oriented programming language developed by Microsoft.
- Characteristics:
  - General-purpose language
  - Part of the .NET framework
  - Syntax like C++ and Java
  - Strongly typed language

# THE C# FAMILY TREE



# C# Version History



- C# 1.0 (2000):
  - Initial release
  - Basic language features
- C# 2.0 (2005):
  - Generics, nullable types, anonymous methods
  - Partial classes, iterators
- C# 3.0 (2007):
  - LINQ (Language Integrated Query)
  - Implicitly typed variables (var)
  - Extension methods
- C# 4.0 (2010):
  - Dynamic typing (dynamic keyword)
  - Named and optional arguments
  - Co- and contravariance
- C# 5.0 (2012):
  - Async and await keywords
  - Caller info attributes
- C# 6.0 (2015):
  - Null-conditional operators
  - Auto-property initializers
  - Expression-bodied members
- C# 7.0 (2017):
  - Tuples, pattern matching
  - Local functions, out variables
- C# 8.0 (2019):
  - Nullable reference types
  - Async streams, switch expressions
  - Ranges and indices
- C# 9.0 (2020):
  - Records, init-only properties
  - Improved pattern matching
- C# 10.0 (2021):
  - Interpolated string handlers
  - File-scoped namespaces



# Key Features of C#

- Object-oriented programming
- Strong type system
- Automatic memory management (Garbage Collection)
- Modern language features (LINQ, async/await, etc.)
- Cross-platform development (with .NET Core/5/6)
- Extensive standard library

# Why Choose C#?



- Reasons to Choose C#: Widely used in enterprise and game development
- Excellent tooling support (Visual Studio)
- Strong community and resources
- Versatility (desktop, web, mobile, IoT)
- Compatibility with .NET ecosystem
- Continuous language improvement

# Introduction to .NET Framework



- Definition: .NET Framework is a software framework developed by Microsoft for building and running Windows applications.
- Components:
  - CLR
  - Class Library
  - Tools
  - CTS and CLS



# Common Language Runtime (CLR)



- Explanation: CLR is the heart of the .NET Framework responsible for executing .NET applications.
- Functions:
  - Compilation to Intermediate Language (IL)
  - Just-In-Time (JIT) Compilation
  - Memory Management
  - Exception Handling
  - Security

# Common Language Specification (CLS)



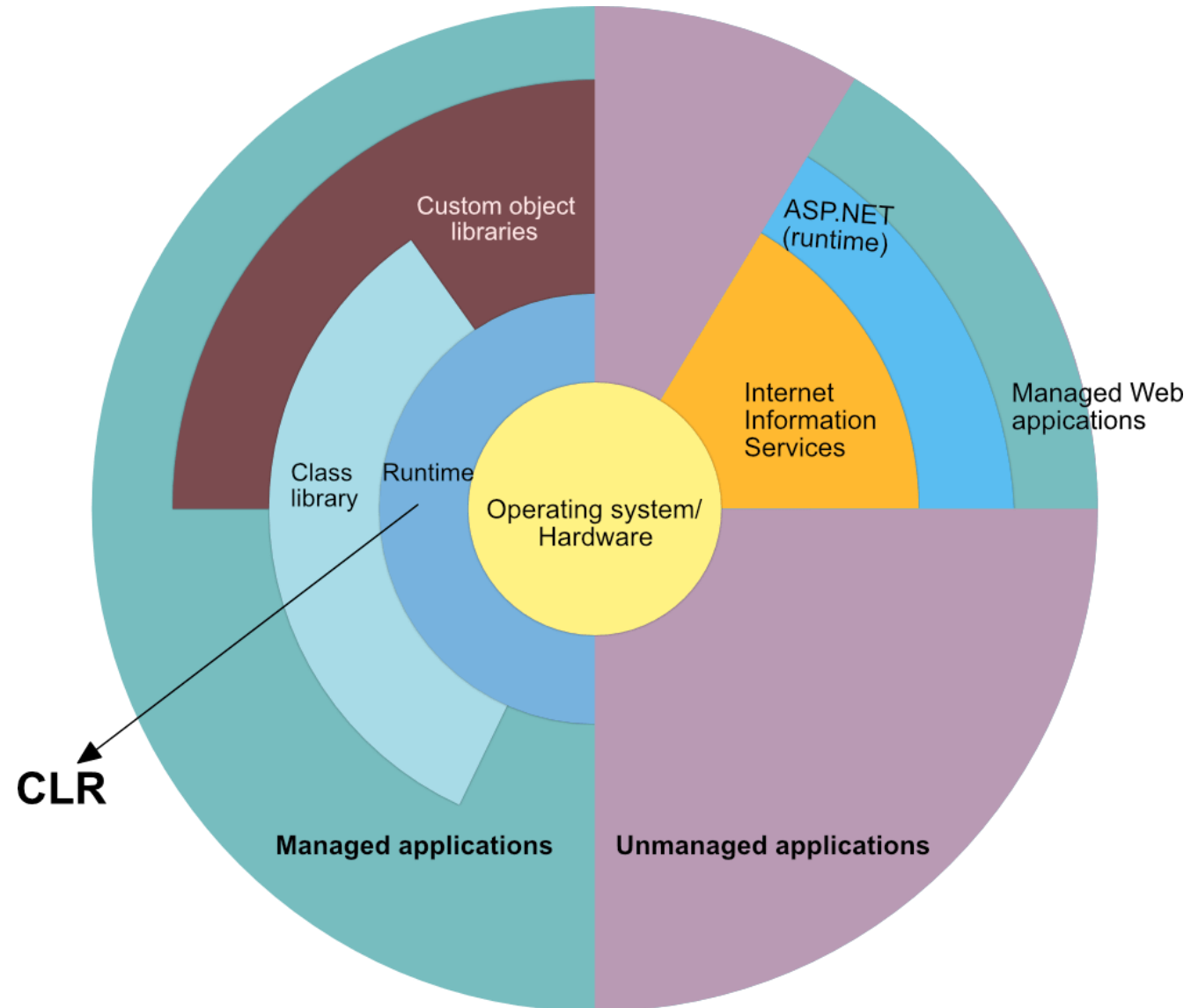
- CLS is a set of rules that .NET languages must adhere to for compatibility.
- Importance:
  - Ensures that code written in different .NET languages can interoperate seamlessly.

# Common Type System (CTS)



- CTS defines the data types that can be used in .NET applications.
- Benefits:
  - Type safety, language interoperability, and a consistent data model.

# How They Work Together



# Benefits and Use Cases



- Benefits:
  - Language Independence
  - Platform Independence
  - Code Reusability
- Use Cases:
  - Windows Applications
  - Web Applications
  - Mobile Applications
  - Gaming
  - Cloud Services

# Sample Program



```
using System;

class Hello
{
    static void Main()
    {
        Console.WriteLine("hello world");
    }
}
```

# Java Vs C#



DIFF	JAVA	C#
Platform	Platform independent	Platform dependent
Technicality	<ul style="list-style-type: none"><li>Doesn't support pointers for developer</li><li>Doesn't have operator Overloading apart from internal (+)</li><li>Doesn't have structure</li></ul>	<ul style="list-style-type: none"><li>supports pointer</li><li>It supports operator overloading</li><li>Does support structure</li></ul>
Ecosystem	Has a huge open-source ecosystem	Used to develop for Microsoft platforms
ORM	Java has more ORM support	C# has LINQ
Pure OOP	Partly OOP language	Complete OOP language

# CONTD...



Diff	Java	C#
Generics	Implemented using erasures and casts	Integrated into the cli and allows type info to be available during runtime
Delegates	Requires use of interface to achieve similar functionality	Which serves as methods that can be called without knowledge of target object
Checked exceptions	Checked and unchecked	Unchecked
Polymorphism	Polymorphism by default	Virtual keyword in base and override in derived
When to use	On unix,linux or other platforms outside of the Microsoft platform	On the .NET FW , mono and portable .NET

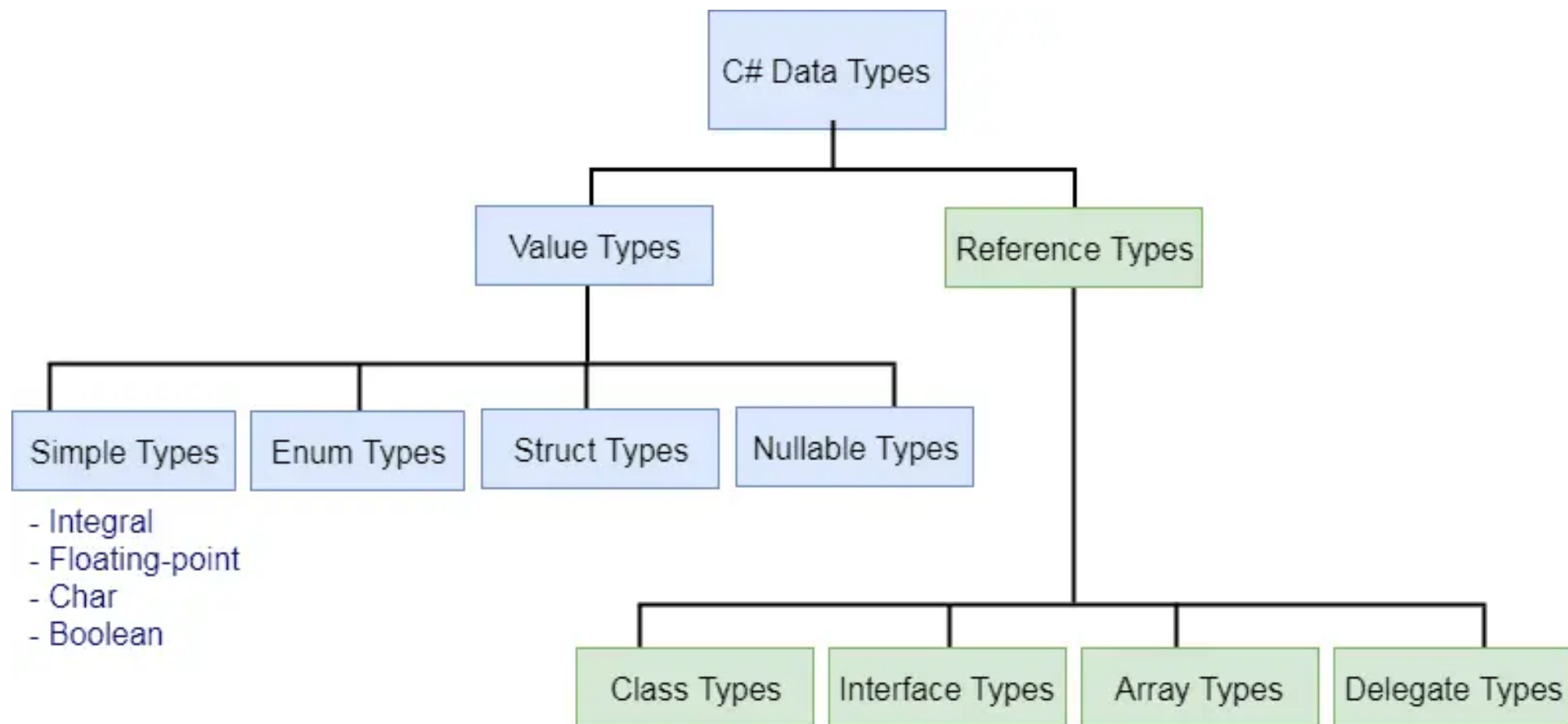




# Fundamentals

- **Data types**
  - Bool,byte, char ,decimal,double ,float,int ,long ,object,short,string
- **Type conversion methods**
  - ToBoolean,ToByte,ToChar,ToDateTime,ToDecimal,ToDouble,ToInt16
- **Operators**
  - Assignment,comparison,arithmetic ,logical & bitwise operator
- **Other operators**
  - Sizeof() , typeof() , & , \* , ?: , is , as
- Enum
- StringBuilder
- Nullable Types

# Data Types



# Operators



	Operator	Type
Binary Operator →	+, -, *, /, %	Arithmetic Operators
	<, <=, >, >=, ==, !=	Relational Operators
	&&,   , !	Logical Operators
	&,  , <<, >>, ~, ^	Bitwise Operators
	=, +=, -=, *=, /=, %=	Assignment Operators
Unary Operator →	++, --	Unary Operators
Ternary Operator →	?:	Ternary Operator or Conditional Operator

# ACCESS MODIFIERS



MODIFIER	EXPLANATION
PUBLIC	Accessible by any code inside or outside the assembly
Private	Only accessibly by code inside a class
Protected	Accessible in the same or derived class
Internal	Accessible in same assembly not from any other assembly
Protected internal	Accessible in same assembly or derived class from another assembly



# Naming conventions

- |                                   |              |
|-----------------------------------|--------------|
| • Class                           | CarClass     |
| • Method                          | GetDetails   |
| • Local variable/parameter        | carMileage   |
| • Private variable                | _vinNo       |
| • Constant                        | PI           |
| • Public/protected/internal field | PetrolRate   |
| • Interface                       | IBankService |



# Thank you

*Innovative Services*



*Passionate Employees*



*Delighted Customers*

