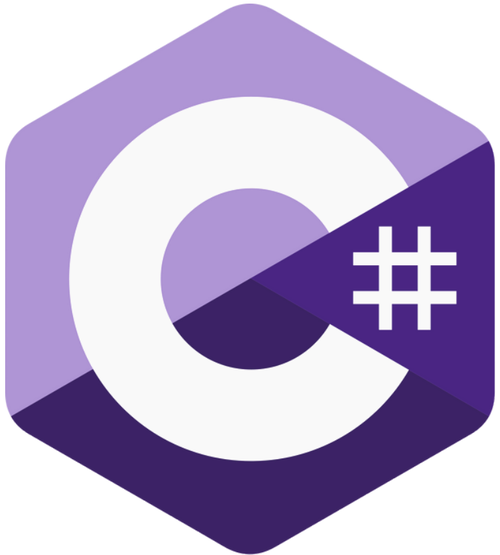
A LITTLE MORE INTO



ANASWARA SURESH . M . K

CB.EN.U4CYS22007

INTRODUCTION

C# (pronounced "C-sharp") is a modern, object-oriented programming language developed by Microsoft in 2000. It is based on the C and C++ programming languages, but includes features such as garbage collection and a simplified syntax. C# is most commonly used to develop Windows desktop and web applications, but it can also be used to develop mobile apps and games using frameworks like Xamarin and Unity.

C# is a statically-typed language, which means that variables must be declared with a specific data type (such as int, string, or bool) before they can be used. It also supports object-oriented programming concepts such as classes, objects, inheritance, and polymorphism.

WE ALWAYS START THINKING WHEN WE ARE PRESENTED WITH QUESTIONS SO I'M GOING TO LIST A BUNCH OF QUESTIONS WHICH WE ARE GOING TO ANSWER AS WE MOVE THROUGH THE POWERPOINT

WHAT IS C# ?

WHEN AND WHO CREATED C# ?

WHAT ARE THE IMPORTANT FEATURES OF C# ?

WHAT PARADIGM IS C# ?

WHAT LANGUAGES INFLUENCED C# ?

WHAT LANGUAGES DID C# INFLUENCE ?

WHICH OPERATING SYSTEMS SUPPORT C# ?

WHAT ARE THE REAL LIFE APPLICATIONS OF C# ?

HOW TO DO " Hello world " PROGRAMMING ?

WHAT IS C# ?

C# is an object-oriented programming language developed by Microsoft in 2000. It is based on C and C++ and is used for developing Windows desktop and web applications, mobile apps and games. C# is statically-typed, supports object-oriented concepts, and is typically written using the Visual Studio development environment. It has a large developer community and many open-source libraries and frameworks are available to help developers write code more efficiently. C# also can be used to develop cross-platform application using .NET Core.

WHEN AND WHO CREATED C# ?

C# was created by Anders Hejlsberg and his team at Microsoft in 2000, as part of the company's .NET initiative. The first version of the language, C# 1.0, was released in 2002 as part of Visual Studio .NET. C# has since undergone several major updates, with the latest version being C# 9.0, which was released in November 2020 as part of .NET 5.0.

WHAT ARE THE IMPORTANT FEATURES OF C# ?

C# is an important programming language for several reasons:

- Widely used: C# is widely used for developing Windows desktop and web applications, as well as mobile apps and games using frameworks like Xamarin and Unity. This makes it a valuable skill for developers to have, as there is a high demand for C# developers in the job market.

- Powerful and versatile: C# is a powerful and versatile language that supports object-oriented programming concepts and has a large set of built-in features, making it well-suited for developing robust, scalable software.

- Cross-platform: C# can be used to develop cross-platform applications using .NET Core, which enables developers to write code once and run it on Windows, Linux, and MacOS.

- Large developer community: C# has a large and active developer community, which means there are many open-source libraries and frameworks available to help developers write code more efficiently. This also makes it easier to find help and resources when working with the language.

- **Strong tooling: C# is typically written using the Visual Studio development environment, which provides a code editor, debugging tools, and a visual designer for building user interfaces. This makes it easy to develop, test, and deploy C# applications.**

In summary, C# is an important programming language with a wide range of uses and a large community of developers. Its versatility, power, and support for object-oriented programming makes it a valuable skill for any developer, and the ability to develop cross-platform applications make it even more appealing.

WHAT PARADIGM IS C# ?

C# is a multi-paradigm programming language, which means it supports multiple programming paradigms such as object-oriented programming (OOP), imperative programming, and functional programming.

Object-oriented programming (OOP) is the primary paradigm used in C#. OOP is a programming model that allows you to organize code into objects that represent real-world entities, and define their behavior and properties. C# supports OOP concepts like classes, objects, inheritance, and polymorphism, which makes it well-suited for developing large, complex software systems.

Imperative programming is also supported by C#, which is a style of programming that describes how a program should change the state of the computer. C# has a wide range of built-in statements and operators that allow you to write imperative code.

Functional programming is another paradigm that is supported by C#, it is a style of programming that emphasizes immutability, referential transparency and the use of first-class functions. C# has introduced functional features like lambda expressions, local functions, and pattern matching to make it easier for developers to write functional code.

C# being multi-paradigm provides the developer with the ability to choose the best approach for a given problem, which can lead to more elegant and maintainable code.

WHAT LANGUAGES INFLUENCED C# ?

C# is heavily influenced by two programming languages in particular: C and C++.

C is a simple programming language that was created in the 1970s. C# borrows many features from C, such as syntax, pointer support, and imperative programming support. C# has a type system similar to C, with data types such as int and double, as well as a memory model.

C++ is a C language extension and one of the most widely used programming languages for system programming and embedded systems. C# borrows many features from C++, such as object-oriented programming support, templates, and exception handling.

Furthermore, C# was influenced by other languages such as Java, which has similar syntax and supports the concept of garbage collection. C# also borrowed some features from Delphi and Object Pascal, such as properties and events.

To summarise, C# is heavily influenced by C and C++, but it also incorporates features and concepts from other languages such as Java, Delphi, and Object Pascal, making it a powerful and versatile programming language.

WHAT LANGUAGES DID C# INFLUENCE ?

C# has influenced a number of other programming languages, including:

- Java: C# and Java are both object-oriented programming languages that are widely used for developing web and enterprise applications. Many of the features and concepts in C#, such as the use of garbage collection and the syntax for defining classes and methods, are similar to those found in Java.

- F#: F# is a functional-first programming language that was developed by Microsoft. It is heavily influenced by C#, and it shares a similar syntax and many of the same features, such as garbage collection, properties, and events.

- Visual Basic .NET: Visual Basic .NET is a programming language that was developed by Microsoft as a successor to Visual Basic 6. It is heavily influenced by C#, and it shares a similar syntax and many of the same features.

- D: D is a systems programming language with C-like syntax and similar features like garbage collection, and it has been influenced by C# and C++.

- Kotlin: Kotlin is a statically typed programming language that runs on the JVM, and it is heavily influenced by C#. It has similar syntax and features like null safety, type inference, and extension functions.

WHICH OPERATING SYSTEMS SUPPORT C# ?

C# can be used to develop applications for several operating systems, including:

Windows: C# is primarily used to develop Windows desktop and web applications, and it is fully supported on all versions of Windows, from Windows XP to the latest Windows version. C# can be used with the .NET Framework or .NET Core to develop applications for Windows.

Linux: C# can also be used to develop applications for Linux using .NET Core. .NET Core is a cross-platform implementation of the .NET Framework that allows developers to write code once and run it on Windows, Linux, and macOS.

macOS: C# can also be used to develop applications for macOS using .NET Core.

Other platforms: In addition to the above platforms, C# can also be used to develop mobile apps and games using frameworks like Xamarin and Unity. Xamarin allows developers to create iOS and Android apps using C# and .NET, while Unity is a game engine that supports C# scripting.

WHAT ARE THE REAL LIFE APPLICATIONS OF C# ?

C# is a versatile programming language that can be used for a wide range of real-life applications, including:

Windows desktop applications: C# can be used to develop a wide range of Windows desktop applications, such as productivity tools, games, and utilities.

Web applications: C# can be used to develop web applications using the ASP.NET framework, which allows developers to create dynamic, interactive web pages and web services.

Mobile applications: C# can be used to develop mobile applications using the Xamarin framework, which allows developers to create iOS and Android apps using C# and the .NET framework.

Games: C# can be used to develop games using the Unity game engine, which supports C# scripting. This allows developers to create 2D and 3D games for multiple platforms, including Windows, iOS, and Android.

R

Enterprise applications: C# can be used to develop enterprise applications, such as inventory management systems, CRM systems, and ERP systems, using frameworks like the .NET framework and the Entity Framework.

IoT: C# can be used to develop applications for Internet of Things (IoT) devices, such as smart home systems, connected cars, and industrial automation systems.

Machine Learning and AI: C# can be used to develop machine learning and AI applications using frameworks like TensorFlow.NET and ML.NET.

In summary, C# is a versatile programming language that can be used to develop a wide range of applications, including Windows desktop applications, web applications, mobile applications, games, enterprise applications, IoT, and Machine Learning and AI applications.

HOW TO DO " Hello world " PROGRAMMING ?

"Hello World!" in C#

```
// Hello World! program
namespace HelloWorld
{
    class Hello {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World!");
        }
    }
}
```

When you run the program, the output will be :

```
Hello World!
```

- ## // Hello World! Program

/ / indicates the beginning of a comment in C#. Comments are not executed by the C# compiler.

They are intended for the developers to better understand a piece of code.

- ## namespace HelloWorld{...}

The namespace keyword is used to define our own namespace. Here we are creating a namespace called HelloWorld.

Just think of namespace as a container which consists of classes, methods and other namespaces.

- ## class Hello{...}

The above statement creates a class named - Helloin C#. Since, C# is an object-oriented programming language, creating a class is mandatory for the program's execution.

- ## static void Main(string[] args){...}

Main() is a method of class Hello. The execution of every C# program starts from the Main() method. So it is mandatory for a C# program to have a Main() method.

- ## System.Console.WriteLine("Hello World!");

For now, just remember that this is the piece of code that prints Hello World! to the output screen.