C# Introduction - What I Learned

About C#

I learned that C# is a high-level, object-oriented programming language developed by Microsoft. It is mainly used for building applications within the .NET ecosystem. It is simple, type-safe, and well-integrated with Windows applications.

What is .NET Framework?

I now understand that the .NET Framework is a software development platform that provides tools, libraries, and a runtime environment to develop applications. It supports multiple programming languages and ensures smooth execution through the Common Language Runtime (CLR).

Architecture of .NET Framework

I learned that the .NET Framework is built on three main components. The Common Language Runtime (CLR) handles memory management and executes the code. The Base Class Library (BCL) provides reusable classes for file handling, networking, and other functionalities. Additionally, the Application Frameworks offer extra tools for web, desktop, and mobile app development.

Running a Hello World Program

I wrote my first C# program, which prints 'Hello World' using the Main() function inside a class. I used Console.WriteLine() to display output, which helped me understand how a basic program runs in C#.

Printing Statements (Write() vs WriteLine())

I learned that Console.Write() prints output on the same line, while Console.WriteLine() moves the cursor to the next line after printing. This difference is useful when formatting console output.

Basic Structure of a C# Program

I now understand that every C# program follows a basic structure. It starts with a namespace to organize code, followed by a class, which defines the program. The Main() function serves as the entry point where the execution begins.

Comments in C#

I learned how to use comments in C#. Single-line comments start with //, while multi-line comments are enclosed within /* */. Comments help in making the code more readable.

Variables and Data Types

I explored different data types in C#. I now know that:
- int is used for whole numbers.
- float and double store decimal values.
- char holds a single character.
- string is used for text.

Taking Inputs in C#

I learned that I can take user input using Console.ReadLine(). However, since it returns a string, I need to convert it to other data types when required.

Size of Data Types

I now know that each data type has a specific memory size. For example, an int takes 4 bytes, a float takes 4 bytes, and a double takes 8 bytes. Understanding this helps in managing memory efficiently.

Typecasting in C#

I learned that typecasting is used to convert one data type into another.
- Implicit casting happens automatically (e.g., converting int to double).
- Explicit casting requires manual conversion using (type) (e.g., double to float).

Character and Strings in C#

I now understand that C# differentiates between characters (char) and strings (string). Strings are sequences of characters, and I can manipulate them using built-in functions.

Conclusion

Overall, I gained a solid understanding of C# basics, including writing simple programs, handling input/output, and working with data types.