Unit-1

Introduce C#, its features and applications

C# is pronounced "C-Sharp".
It is an object-oriented programming language created by Microsoft that runs on the .NET Framework.
C# has roots from the C family, and the language is close to other popular languages like C++ and Java.
The first version was released in year 2002. The latest version, **C# 8**, was released in September 2019.

C# is used for:

- Mobile applications
- Desktop applications
- Web applications
- Web services
- Web sites
- Games
- VR
- Database applications
- And much, much more

Reasons to use c#

- It is one of the most popular programming language in the world
- It is easy to learn and simple to use
- It has a huge community support
- C# is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As C# is close to C, C++ and Java, it makes it easy for programmers to switch to C# or vice versa

Visual Studio Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality. The Visual Studio IDE is a creative launching pad that you can use to edit, debug, and build code, and then publish an app.

1.2 Structure of C#

```
using System;

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

Example explaination

1 using System means that we can use classes from the System namespace.

2: namespace is used to organize your code, and it is a container for classes and other namespaces.

3: The curly braces {} marks the beginning and the end of a block of code.

4: class is a container for data and methods. Every line of code that runs in C# must be inside a class. In our example, we named the class Program.

5: The Main method: Any code inside its curly brackets {} will be executed

6: Console is a class of the System namespace, which has a WriteLine() method that is used to print text. In our example it will output "Hello World!".

Note: Every C# statement ends with a semicolon ;.

Note: When saving the file, save it using a proper name and add ".cs" to the end of the filename.

1.3 C# Variables

A variable is nothing but a name given to a storage area that our programs can manipulate. Each variable in C# has a specific type, which determines the size and layout of the variable's memory the range of values that can be stored within that memory and the set of operations that can be applied to the variable.

Rules to create variable name:

- The name can contain letters, digits, and the underscore character (_).
- The first character of the name must be a letter or underscore. C# is case-sensitive;
- C# keywords can't be used as variable names.
- Syntax: *type variableName = value;*

Example **string name = "Rohit";**

1.4 C# identifier

In programming languages, identifiers are used for identification purposes. Or in other words, identifiers are the user-defined name of the program components. In C#, an identifier can be a class name, method name, variable name, or label.

Rules for defining identifiers in C#:

- The only allowed characters for identifiers are all alphanumeric characters(**[A-Z]**, **[a-z]**, **[0-9]**), '_' (underscore).
- Identifiers should not start with digits([0-9
- Identifiers should not contain white spaces.

- Identifiers are not allowed to use as keywords unless they include @ as a prefix. For example, **@as** is a valid identifier, but "**as**" is not because it is a keyword.
- C# identifiers allow Unicode Characters.
- C# identifiers are case-sensitive.
- C# identifiers cannot contain more than 512 characters.
- Identifiers do not contain two consecutive underscores in their name because such types of identifiers are used for the implementation.

C# Keywords

Keywords are predefined sets of reserved words that have special meaning in a program. The meaning of keywords can not be changed, neither can they be directly used as identifiers in a program. C# has a total of 79 keywords. All these keywords are in lowercase

Abstract	As	base	bool
Break	Byte	case	catch
Char	Checked	class	const
Continue	Decimal	default	delegate
Do	Double	else	enum
Event	Explicit	extern	false
Finally	Fixed	float	for
Foreach	Goto	if	implicit
In	in (generic modifier)	int	interface
Internal	Is	lock	long
Namespace	New	null	object
Operator	Out	out (generic modifier)	override
Params	Private	protected	public
Readonly	Ref	return	sbyte
Sealed	Short	sizeof	stackalloc
Static	String	struct	switch

This	Throw	true	try
Typeof	Uint	ulong	unchecked
Unsafe	Ushort	using	using static
Void	Volatile	while	

Keywords can be used as identifiers if `@` is added as prefix.