C# Keywords (Reserved, Contextual)

In c#, **Keywords** are the predefined set of reserved words that have special meaning for the compiler. So the keywords in c# cannot be used as identifiers such as variable name, class name, etc., in our applications.

Use Keywords as Variable Names

In c#, if you want to use Keywords as variable names (identifiers), you need to include `@` as a prefix for your variable names. For example, **@switch** is a valid identifier, but the **switch** is not because it's a keyword and having a special meaning for the compiler.

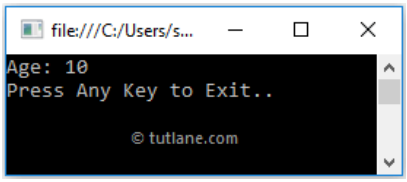
Following is the example of using the reserved keywords as variable names by including `@` as a prefix in c# programming language.

```
using System;

namespace CsharpExamples
{
    public class @class
    {
        public int age;
    }
    class Program
    {
        static void Main(string[] args)
        {
            @class p1 = new @class();
            p1.age = 10;
            Console.WriteLine("Age: "+p1.age);
            Console.WriteLine("Press Enter Key to Exit..");
            Console.ReadLine();
        }
    }
}
```

If you observe the above c# example, we used a **class** keyword as a variable name (**@class**) by including `@` as a prefix.

When you execute the above program, you will get the result as shown below.



We can use keywords as variable names in the c# programming language based on our requirements.

Different Types of Keywords

In c#, Keywords are differentiated into two types those are

- Reserve Keywords
- Contextual Keywords

Reserved keywords

The following table lists the available reserved keywords in the c# programming language.

abstract	bool	continue	decimal	default
event	explicit	extern	char	checked

class	const	break	as	base
delegate	is	lock	long	num
byte	case	catch	false	finally
fixed	float	for	as	foreach
goto	if	implicit	in	int
interface	internal	do	double	else
namespace	new	null	object	operator
out	override	params	private	protected
public	readonly	sealed	short	sizeof
ref	return	sbyte	stackalloc	static
string	struct	void	volatile	while
true	try	switch	this	throw
unchecked	unsafe	ushort	using	using static
virtual	typeof	uint	ulong	out (generic modifier)

Contextual keywords

In c#, **Contextual** keywords can be used as an identifier in a limited program context, which can be outside of the context.

Generally, whenever the new keywords are added to the C# language, those are treated as **Contextual** keywords to avoid breaking c# programs that we wrote in older versions.

The following table lists the available Contextual Keywords in the c# programming language.

add	alias	async	await	dynamic
from	get	orderby	ascending	descending
group	into	join	let	nameof
global	partial	set	remove	select
value	var	when	Where	yield

These are the keywords available in the c# programming language, and we can use them in our applications based on our requirements.

CONTACT US

Address: No.1-93, Pochamma Colony, Manikonda, Hyderabad, Telangana - 500089

Email: support@tutlane.com (mailto:support@tutlane.com)