

Basic C # syntax

C # is an object-oriented programming language. In an object-oriented programming method, a program of diverse objects that interacts with the ways of action. Actions that an object can receive are called methods.

C # is an object-oriented programming language. In an object-oriented programming method, a program consists of diverse objects that interact with each other by action methods. Actions that an object can receive are called methods. Objects of the same type are considered as having the same type or, as if they were in the same class.

For example, consider the Rectangle object. It has attributes like length and width. Depends on the design above. It may need ways to grant these attribute values, calculate area and show details.

You follow the example after deploying a Rectangle class and the basic C # syntax of this program. To clearly illustrate the object-oriented nature of C #, I will create two classes in two files separately.

Rectangle : contains the main properties and methods:

```
using System ; namespace QTMsharp { class Rectangle { // các bi?
n thành viên double length ; double width ; // ph??ng th?
c public void Acceptdetails () { length = 4.5 ; width = 3.5 ; } // ph??
ng th?c public double GetArea () { return length * width ; } // ph??
ng th?
c public void Display () { Console . WriteLine ( "Chieu dai: {0}" , length ); C
```

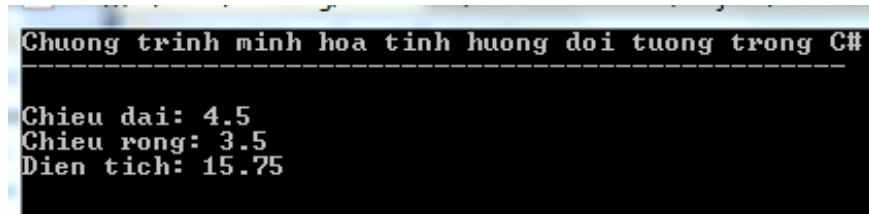
Class ExecuteRectangle : is a class containing main () method to conduct manipulation on Rectangle object of Rectangle class. Come here, maybe you do not understand anything, but it's okay because this is just a chapter to help you get acquainted with C #.

```
dùng System;
```

```
QTMsharp namespace
{
class ExecuteRectangle
{
static void Main (string [] args)
{
Console.WriteLine ("Full version in C #");
Console.WriteLine ("----- n ");
// tao doi tuong Rectangle
Rectangle r = new Rectangle ();
// Call the current version
r.Acceptdetails ();
r.Display ();
Console.ReadLine ();
```

```
Console.ReadKey ();  
}  
}
```

Pressing F5 to compile and run the above C # program will produce the following results:



```
Chương trình minh họa tính huống đối tượng trong C#  
-----  
Chieu dai: 4.5  
Chieu rong: 3.5  
Dien tich: 15.75
```

Using keyword in C

The first command in any C # program is:

```
using System ;
```

The using keyword is used to include namespaces in the program. A C # program can include many using statements.

Keyword class in C

The keyword class is used to declare a class in C #.

Comments in C

The comment is used to initialize the code. Compiler ignores comments. Multi-line comments in C # programs start with /* and end with */ as follows:

```
/* dòng này minh họa comment nhiều dòng trong C#. Cú pháp cơ bản C# Ngôn ngữ lập
```

Single-line comments are indicated by the symbol '//'. For example:

```
// ví dụ comment đơn dòng trong C#
```

Turning members in C

Variables are attributes or data members of a class, used to store data. In the previous program, Rectangle class has two member variables which are length and width.

Member functions in C

A function is a collection of commands that perform a specific task. The member functions of a class are declared inside that class. Rectangle class contains 3 member functions: AcceptDetails, GetArea and Display.

Explain a Class in C

In the above program, the ExecuteRectangle class contains the **Main () method** and initializes the Rectangle class.

Identifier in C

An identifier is a name used to identify a class, variable, function or any self-defined item (user-defined).

A name must start with a letter that can be followed by a sequence of letters, numbers (0-9) or underscore (_). The first character of an identifier cannot be a digit.

It must not contain any spaces or characters like? - +! @ #% ^ & * () [] { }. ; : "/ and. However, underscores can be used.

It should not be a keyword in C #.

Keywords in C

Keywords are predefined reserved words (Reserved Keyword) for C # compiler. These keywords cannot be used as identifiers. However, if you want to use these keywords as identifiers, you can set the @ character before them.

In C #, some identifiers that have special meanings in the context of the code, such as get and set, are called contextual keywords.

The following table lists reserved keywords and contextual keywords in C #:

Reserved Keyword abstract as base bool break byte case checked catch char class const continue decimal default delegate due to double else the event enum explicit extern false finally fixed float for foreach goto if implicit in (Generic modifier) int internal interface (lock) không có m?t tên c?a ???ng d?n không th? xác ??nh là m?t tên t?p tin, không ph?i là m?t s? th? t? ??i t??ng, không rõ ?ã k?t thúc v?i b? ??m chung (tùy ch?n chung ?ã xác ??nh) public readonly ref ?ã chuy?n l?i sealed Sbyte sizeof static stack string struct chuy?n ??i này không th? th?c hi?n typeof uint ulong unchecked unsafe ushort using virtual void volatile while Contextual keyword inh global group vào Join m?t let (không h?p l?) ph?n (ki?u) ph?n (method) remove select set

Follow tutorialspoint

Previous article: C # program structure

Next lesson: Data type in C #

You finished reading the article "**Basic C # syntax**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.