



Lập trình .Net

- ☐ Lý thuyết: 45 Tiết
- ☐ Thực hành: 30 Tiết
- ☐ Giảng viên: Võ Tấn Toàn
- ☐ Email: vttoan@hcmuaf.edu.vn



Đánh giá môn học



- ❖ Bài tập lab (nộp bài qua email): 10%
- ❖ Thi giữa kỳ hoặc seminar 30%
 - Giữa kỳ: Thi thực hành online tại google classroom
 - Seminar về các chủ đề (5 thành viên), tối đa 5 nhóm đăng ký cho **Lớp trưởng**
- ❖ Bài kiểm tra nhanh tại lớp: 10%
- ❖ Cuối kỳ thi vấn đáp project (4 thành viên) 50%





Sự ra đời của .NET

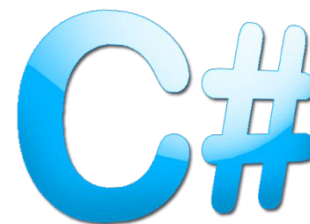


- ❖ Tháng 7 năm 2000, Microsoft đã bắt đầu cho ra mắt về nền tảng .NET Framework. Một ngôn ngữ mới cũng được họ giới thiệu là COOL – C-like Object Oriented Language – trước khi được lấy tên là C# thì nó còn có một cái tên nữa là COM3.
- ❖ Ngày 12/2/2002, .NET Framework 1.0 ra đời đi cùng bộ Visual Studio.NET 2002.
- ❖ .NET là một platform định nghĩa những nền tảng, căn bản để cho các lập trình viên dựa vào đó và phát triển ứng dụng.
- ❖ C# là ngôn ngữ nền tảng, chiến lược của Microsoft.





- ❖ C# là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft
- ❖ C# được phát triển trên nền tảng ngôn ngữ C và C++.
- ❖ Là ngôn ngữ lập trình đơn giản hướng đối tượng.
- ❖ C# được sử dụng cho nhiều dự án như: xử lý văn bản, ứng dụng đồ họa, xử lý bảng tính; thậm chí tạo ra những trình biên dịch cho các ngôn ngữ khác.





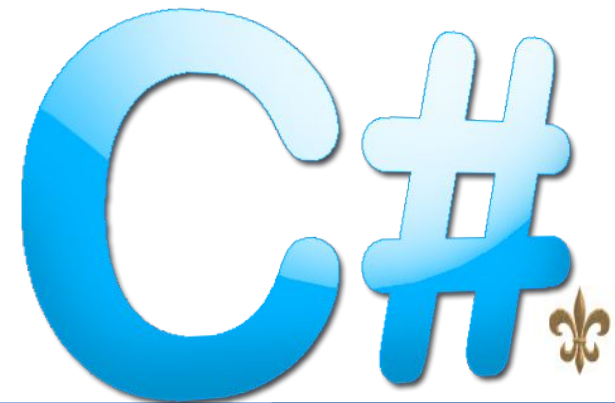
Điểm mạnh mẽ của C#



❖ Cấu trúc C# khá gần với các ngôn ngữ high-level truyền thống, C và C++, và là một ngôn ngữ lập trình hướng đối tượng. Nó **có sự giống nhau mạnh mẽ với Java**, được lập trình viên trên toàn thế giới ưa chuộng.

❖ Một số đặc điểm quan trọng của C#:

- Tự động dọn rác bởi Garbage-Collector (GC)
- Thư viện chuẩn (Standard Library)
- Đa luồng dễ dàng (Multithreading)
- Tích hợp với Windows





C# HelloWorld



❖ Vd cơ bản:

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






Biên dịch và thực thi Chương trình C#



- ❖ Nếu bạn đang sử dụng Visual Studio.Net để biên dịch và thực thi các chương trình C#:




Downloads

Visual Studio 2019
Full-featured integrated development environment (IDE) for Android, iOS, Windows, web, and cloud



Version: 16.0
[Release notes](#)

[Compare editions](#)
[How to install offline](#)

Community	Professional	Enterprise
Powerful IDE, free for students, open-source contributors, and individuals	Professional IDE best suited to small teams	Scalable, end-to-end solution for teams of any size
Free download 	Free trial 	Free trial 
Download Preview >	Download Preview >	Download Preview >

thực
có lỗi,
t hiện
màn





Biên dịch và thực thi Chương trình C#



- ❖ Bạn có thể biên dịch một chương trình C# bởi sử dụng command-line thay cho Visual Studio IDE:
- ❖ Mở một Text Editor và thêm code trên vào.
- ❖ Lưu tệp dưới dạng **helloworld.cs**
- ❖ Mở công cụ Command Prompt tool và tới thư mục nơi bạn lưu file đó.
- ❖ Soạn **csc helloworld.cs** và nhấn Enter để biên dịch code của bạn.
- ❖ Nếu không có lỗi nào trong code, thì Command prompt đưa bạn tới dòng tiếp theo và tạo tệp **helloworld.exe** có thể thực thi (executable).
- ❖ Soạn **helloworld** để thực thi chương trình của bạn.
- ❖ Bạn có thể thấy output là Hello World được in trên màn hình.





C# VD



```
using System;
namespace ITNongLam
{
    class Rectangle
    {
        // cac bien thanh vien
        double length;
        double width;
        //phuong thuc
        public void Acceptdetails() { length = 4.5; width = 3.5; }
        //phuong thuc
        public double GetArea() { return length * width; }
        //phuong thuc
        public void Display() { Console.WriteLine("Chieu dai: {0}", length);
        Console.WriteLine("Chieu rong: {0}", width);
        Console.WriteLine("Dien tich: {0}", GetArea()); } }
    }
```





C# VD



```
using System;
namespace ITNongLam
{
    class ExecuteRectangle
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Chương trình minh họa tình huống đối  
tượng trong C#");
            Console.WriteLine("-----\n");
            //tao doi tuong
            Rectangle Rectangle r = new Rectangle();
            //goi cac phuong thuc cua doi tuong nay
            r.Acceptdetails();
            r.Display();
            Console.ReadLine();
            Console.ReadKey(); } }
```





❖ Từ khóa *using* trong C#

- Lệnh đầu tiên trong bất kỳ chương trình C# nào là:
- `using System;`
- Từ khóa *using* được sử dụng để chứa namespace trong chương trình.
- Một chương trình C# có thể chứa nhiều lệnh *using*.

❖ Từ khóa *class* trong C#

- Từ khóa *class* được sử dụng để khai báo một lớp trong C#.





❖ Comments trong C#

- Compiler bỏ qua các comment. Các comment đa dòng trong các chương trình C# bắt đầu với **/*** và kết thúc với ***/** như sau:
- **/* dòng này minh họa comment nhiều dòng trong C#. Cú pháp cơ bản C# Ngon ngu lập trình C# */**
- Comment đơn dòng được chỉ dẫn bởi ký hiệu **//**. Ví dụ:
- **// ví dụ comment đơn dòng trong C#**





❖ Định danh (Identifier) trong C#

- Một định danh là một tên được sử dụng để nhận diện một lớp, biến, hàm hoặc bất kỳ mục tự định nghĩa (user-defined).
- Một tên phải bắt đầu với một chữ cái mà có thể được theo sau bởi một dãy các chữ cái, chữ số (0-9) hoặc dấu gạch dưới (_). Ký tự đầu tiên của một định danh không thể là một chữ số.
- Nó phải không chứa bất kỳ khoảng trống hoặc ký tự như ? - + ! @ # % ^ & * () [] { } . ; : " ' / và \. Tuy nhiên, dấu gạch dưới có thể được sử dụng.
- Nó không nên là một từ khóa trong C#.





❖ Từ khóa trong C#

- Từ khóa là các từ dành riêng (*Reserved Keyword*) được định nghĩa trước cho C# compiler.
- Những từ khóa này không thể được sử dụng làm định danh.
- Tuy nhiên, nếu bạn muốn sử dụng các từ khóa này để làm định danh, bạn có thể đặt ký tự @ ở trước chúng.





Biên dịch và thực thi Chương trình C#



- ❖ C# là ngôn ngữ lập trình có sự phân biệt kiểu chữ hoa, thường (case sensitive).
- ❖ Tất cả lệnh và biểu thức phải kết thúc với một dấu chấm phẩy (;).
- ❖ Sự thực thi chương trình bắt đầu tại phương thức Main.
- ❖ Không giống Java, tên file chương trình có thể khác tên lớp.





❖ Các biến trong C# được phân chia thành các kiểu sau:

- Kiểu giá trị (Value type)
- Kiểu tham chiếu (Reference type)
- Kiểu con trỏ (Pointer type)





Kiểu dữ liệu (kiểu giá trị)

Data Types	Default Value	Minimum Value	Maximum Value
sbyte	0	-128	127
byte	0	0	255
short	0	-32768	32767
ushort	0	0	65535
int	0	-2147483648	2147483647
uint	0u	0	4294967295
long	0L	-9223372036854775808	9223372036854775807
ulong	0u	0	18446744073709551615
float	0.0f	$\pm 1.5 \times 10^{-45}$	$\pm 3.4 \times 10^{38}$
double	0.0d	$\pm 5.0 \times 10^{-324}$	$\pm 1.7 \times 10^{308}$
decimal	0.0m	$\pm 1.0 \times 10^{-28}$	$\pm 7.9 \times 10^{28}$
bool	false	Two possible values: true and false	
char	'\u0000'	'\u0000'	'\uffff'
object	null	-	-
string	null	-	-





Kiểu tham chiếu trong C#



- ❖ Kiểu tham chiếu không chứa dữ liệu thực sự được lưu giữ trong một biến, nhưng chúng chứa một tham chiếu tới các biến.
- ❖ Nói cách khác, chúng tham chiếu tới một vị trí bộ nhớ.
- ❖ Ví dụ các kiểu tham chiếu **có sẵn** trong C# là: **object**, **dynamic**, và **string...**





Kiểu Dynamic trong C#



- ❖ Bạn có thể lưu giữ bất kỳ kiểu giá trị nào trong biến kiểu dữ liệu **dynamic**. Việc kiểm tra các kiểu biến này diễn ra tại run time.
- ❖ **Cú pháp để khai báo một kiểu dynamic trong C# là:**
 - `dynamic <tên_biến> = giá_trị;`
 - Ví dụ : `dynamic d = 20;`
- ❖ Kiểu dynamic là tương tự với kiểu object





Kiểu con trỏ trong C#



- ❖ Các biến kiểu con trỏ lưu giữ địa chỉ bộ nhớ của kiểu khác. Các con trỏ trong C# có khả năng như con trỏ trong C hoặc C++.
- ❖ Cú pháp để khai báo một kiểu con trỏ trong C# là:
- ❖ `type* identifier;`

Ví dụ:

```
char* cptr;
```

```
int* iptr;
```





Chuyển đổi kiểu dữ liệu

- ❖ Ép kiểu ngầm định: Việc ép kiểu này được C# compiler thực hiện tự động. Khi ép kiểu ngầm định thì kiểu dữ liệu đích phải có dãy giá trị lớn hơn dãy giá trị nguồn, điều này nhằm đảm bảo rằng sẽ không mất mát dữ liệu sau khi ép kiểu.

❖ Ex:

```
int valueOne = 34;
```

```
//ép kiểu từ int --> float
```

```
//Compiler sẽ tự động ép kiểu
```

```
float valueTwo = valueOne;
```





Ép kiểu tường minh.



- ❖ Việc thay đổi kiểu dữ liệu từ dãy giá trị lớn sang kiểu dữ liệu có dãy giá trị nhỏ hơn gọi là ép kiểu tường minh.
- ❖ Cú pháp:
 - **datatype** name = (dataType) oldVariable;
- ❖ Lưu ý:
 - Khi sử dụng ép kiểu tường minh sẽ có thể dẫn tới mất dữ liệu.





Parse dữ liệu từ String

- ❖ Để lấy được giá trị của dữ liệu trong một chuỗi chúng ta sử dụng phương thức Parse, mỗi kiểu dữ liệu cơ bản của C# đều có phương thức này. Parse được sử dụng để chuyển từ kiểu string sang kiểu dữ liệu cơ bản.

```
static void Main(string[] args)
{
    //Chuyển từ kiểu chuỗi sang kiểu số tự nhiên.
    string myString = "12345";
    int myInt = int.Parse(myString);

    //Chuyển từ kiểu chuỗi sang kiểu số thực
    float myFloat = float.Parse(myString);
}
```





Phép toán

❖ Toán tử so sánh

Tên toán tử	Ký hiệu	Biểu thức logic	Giá trị	Logic
and	&&	$(x == 3) \&\& (y == 7)$	false	Cả hai điều kiện phải đúng
or		$(x == 3) (y == 7)$	true	Chỉ cần một điều kiện đúng
not	!	$!(x == 3)$	true	Biểu thức trong ngoặc phải sai.





Phép toán



❖ Toán tử so sánh

Toán tử	Ý nghĩa
+=	Cộng thêm giá trị toán hạng bên phải vào giá trị toán hạng bên trái
-=	Toán hạng bên trái được trừ bớt đi một lượng bằng giá trị của toán hạng bên phải
*=	Toán hạng bên trái được nhân với một lượng bằng giá trị của toán hạng bên phải.
/=	Toán hạng bên trái được chia với một lượng bằng giá trị của toán hạng bên phải.
%=	Toán hạng bên trái được chia lấy dư với một lượng bằng giá trị của toán hạng bên phải.





Câu lệnh điều khiển if

Cú pháp:

if (Điều_Kiện)

<Khối lệnh Điều_Kiện đúng>

[else

<Khối lệnh Điều_Kiện sai>]

Ví dụ Dùng câu lệnh điều kiện if ... else

```
using System;
class Chan_Le
{
    static void Main()
    {
        // Khai báo và khởi tạo biến
        int bienDem = 9 ;
        // Xuất ra màn hình
        if (bienDem % 2 == 0)
            Console.WriteLine("{0} là số chẵn", bienDem) ;
        else
            Console.WriteLine("{0} là số lẻ", bienDem) ;
    }
}
```



Câu lệnh điều khiển if (lồng nhau)



Cú pháp:

```
if (Điều_Kiện_1)
    <Khởi_lệnh_1>
else if (Điều_Kiện_2)
    <Khởi_lệnh_2.1>
else
    <Khởi_lệnh_2.2>
```

Ví dụ :

```
using System;
class Thu_Trong_Tuan
{
    static void Main()
    {
        // Khai báo và khởi tạo biến
        int thu = 5 ; // 0: Chu nhật, 1: Thu hai, 2: Thu ba, 3: Thu tu,
                     // 4: Thu nam, 5: Thu sau, 6: Thu bay
        // Xuất ra màn hình
        if ((thu == 1) || (thu == 3) || (thu == 5))
            Console.WriteLine("Day la ngay 2-4-6") ;
        else if ((thu == 2) || (thu == 4) || (thu == 6))
            Console.WriteLine("Day la ngay 3-5-7") ;
        else Console.WriteLine("Day la ngay chu nhật") ;
    }
}
```





Cấu trúc switch case

Cú pháp:

switch (Biểu_Thức)

```
{  
    case <giá_trị_1>:  
        <Khởi_lệnh_1>  
        <Lệnh_Nhảy>  
    case <giá_trị_2>:  
        <Khởi_lệnh_2>  
        <Lệnh_Nhảy>  
    ....  
    [default:  
        <Khởi_lệnh_khác>]  
}
```





```
using System;
class Thu
{
    static void Main()
    {
        // Khai bao va khoi tao bien
        int thu = 5 ; // 0: Chu nhat, 1: Thu hai, 2: Thu ba, 3: Thu tu,
                     // 4: Thu nam, 5: Thu sau, 6: Thu bay
        // Xuat ra man hinh
        switch (thu)
        {
            case 0:
                Console.WriteLine("Chu nhat") ;
                break;
            case 1:
                Console.WriteLine("Thu hai") ;
                break;
            case 2:
```





```
        Console.WriteLine("Thu ba");  
        break;  
    case 3:  
        Console.WriteLine("Thu tu");  
        break;  
    case 4:  
        Console.WriteLine("Thu nam");  
        break;  
    case 5:  
        Console.WriteLine("Thu sau");  
        break;  
    case 6:  
        Console.WriteLine("Thu bay");  
        break;  
    default:  
        Console.WriteLine("Khong phai la thu trong tuan");  
        break;  
    }  
}  
}
```





Cấu trúc while

Cú pháp:

while (Điều_Kiện)
 < Khối_lệnh >

Ví dụ 2.4:

```
using System;
class UsingWhile
{
    static void Main()
    {
        // Khai báo và khởi tạo biến đếm
        int i = 1 ;
        // Xuất ra màn hình
        while (i<=10) {
            Console.WriteLine("i = {0}",i) ;
            i++ ; // tăng biến đếm,
        }
    }
}
```





Cấu trúc do while

Cú pháp:

do

< Khối lệnh >

while (Điều_Kiện) ;

```
using System;
class UsingDoWhile
{
    static void Main()
    {
        // Khai báo và khởi tạo biến đếm
        int i = 1 ;
        // Xuất ra màn hình
        do {
            Console.WriteLine("i = {0}",i) ;
            i++ ; // tăng biến đếm
        } while (i <= 10) ;
    }
}
```




Cấu trúc for

Cú pháp:

```
for ([Khởi_tạo] ; [Điều_kiện] ; [Bước_lặp])  
    <Khởi_lệnh>
```

Ví dụ 2.6:

```
using System;  
class UsingFor  
{  
    static void Main()  
    {  
        for (int i=1 ; i<=30 ; i++)  
            if (i % 10 ==0)  
                Console.Write("{0} \n\r",i) ;  
            else  
                Console.Write("{0} ",i) ;  
    }  
}
```





For each

Cú pháp:

foreach (<Kiểu_tập_hợp> <Tên_truy_cập_thành_phần> in <Tên_tập_hợp>)
< Khối lệnh>

Ví dụ 2.7:

```
using System;
public class UsingForeach
{
    public static void Main()
    {
        int[] intArray = {1,2,3,4,5,6,7,8,9,10};
        foreach (int item in intArray)
            Console.WriteLine("i = {0} ",item) ;
    }
}
```





Bài tập về nhà



- ❖ Cài đặt Visual Studio
- ❖ Viết chương trình Hello world (biên dịch bằng 2 cách, sử dụng IDE và sử dụng command line
- ❖ Viết chương trình giải phương trình bậc 2 bằng ngôn ngữ C#





THE END!

Q & A

