.Net Programming

- **Theory**: 45 Lessons
- **Practice**: 30 Lessons
- **Lecturer**: MSc. Vo Tan Toan
- **Email**: vttoan@hcmuaf.edu.vn

Subject assessment

- Lab assignments (submitted via email): 5%
- Midterm exam or seminar 30%
- Seminar on topics (5 members), up to 08 groups
- Quick test in class: 5%
- Final Exam: oral examination 60% 5 members

Subject goals

- Master the basics of programming (variables, functions, branching structures, data types, etc.)
- Master the basics of object-oriented programming
- Mastering ASP.NET MVC web programming knowledge
- Learn to expand technologies such as WPF, Xamarin (MAUI), SOAP, ...

References

- Lecture slides
- Books:
  - Joyce Farrell "*Microsoft® Visual C# 2010: An Introduction to Object-Oriented Programming*"
  - John Sharp, *"Microsoft Visual C# 2008 Step by Step"*
  - Rob Miles, "*C# Programming*"
  - Phạm Hữu Khang, "*C# 2005 - Tập 3: Lập Trình Hướng Đối Tượng*"
- Internet

.NET History

- In July 2000, Microsoft began to release the .NET Framework platform. A new language was also introduced by them, COOL - C-like Object Oriented Language - named C# was born.
- On February 12, 2002, .NET Framework 1.0 was released with Visual Studio .NET 2002.
- .NET is a platform that defines the basic foundations for programmers to rely on and develop applications on.
- C# is Microsoft's strategic and foundational language.

C#

- C# is an object-oriented programming language developed by Microsoft
- C# is developed on basic of C and C++ languages.
- It is a simple object-oriented programming language.
- C# is used for many projects such as: text processing, graphics applications, spreadsheet processing; even create compilers for other languages.

C#

- C#'s structure is quite close to traditional high-level languages such as C and C++, and is an object-oriented programming language. It has a strong similarity to Java, which is popular with programmers around the world.
- Some important features of C#:
  - Automatic garbage collection by Garbage-Collector(GC)
  - Standard Library
  - Multithreading made easy (Multithreading)
  - Integration with Windows

C# Example:

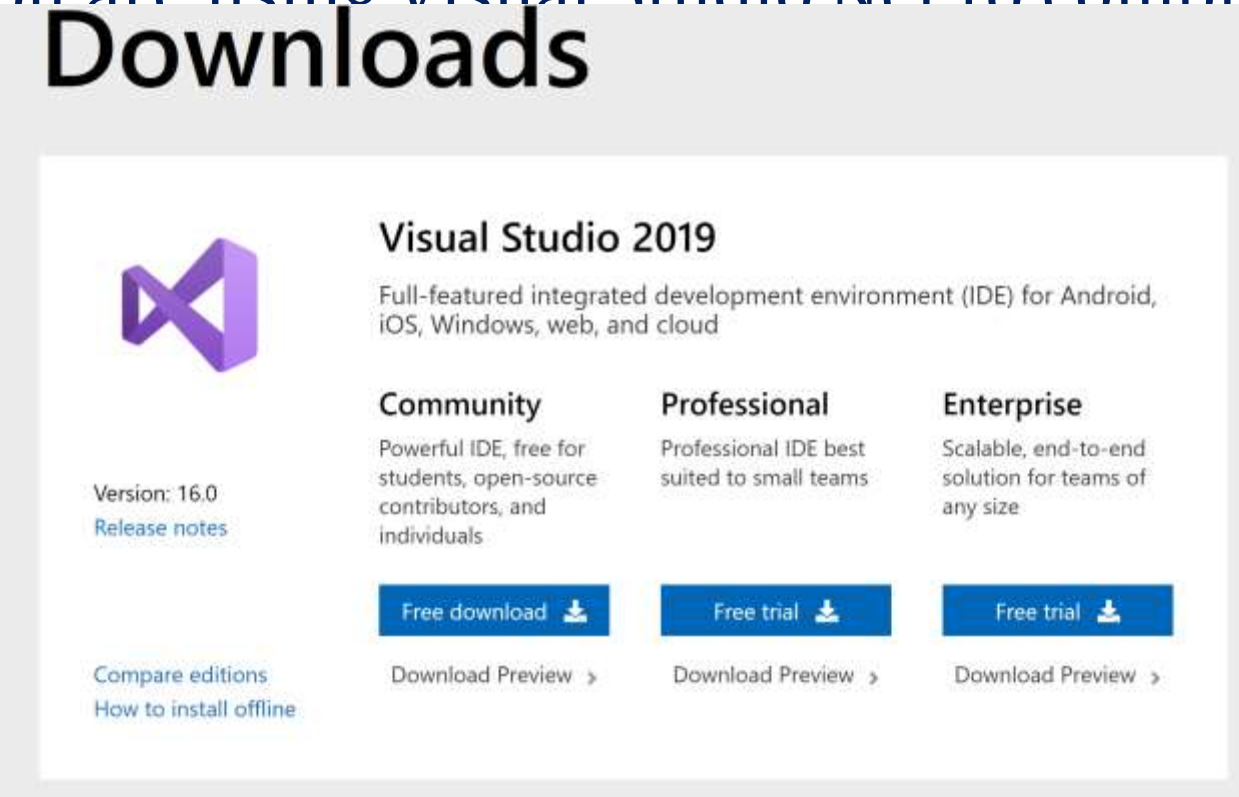
- Basic example:

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

}
```

Compile and execute C# Program

- If you are using Visual Studio Net to compile and exe
- Wri ...utton or p
- If t ...mand Pro ... Hello Wo



Compile and execute C# Program

- You can compile a C# program using the command-line instead of the Visual Studio IDE:
- Open a Text Editor and add the above code.
- Save the file as helloworld.cs
- Open the Command Prompt tool and go to the folder where you saved the file.
- Type csc helloworld.cs and press Enter to compile your code.
- If there are no errors in the code, the Command prompt takes you to the next line and creates the executable helloworld.exe file.
- You can see the output as Hello World printed on the screen.

C# Example

```
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# Example

```
using System;
namespace ITNongLam
{ class ExecuteRectangle
{ static void Main(string[] args)
{ Console.WriteLine("Chuong trinh minh hoa tinh huong doi
tuong 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(); } }
```

C# Basic syntax

- The using keyword in C#
  - The first command in any C# program would be:
  - using System;
  - The using keyword is used to contain the namespace in the program.
  - One C# program can contain many using statements.
- The class keyword in C#
  - The class keyword is used to declare a class in C#.

C# Basic syntax

Comments in C#

- Compiler ignores comments.
- Multiline comments in C# programs begin with
- /* and end with */ as follows:

```
/* This line illustrates multiple
line comments in C#.
...*/
```

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

```
// vi du comment don dong in C#
```

C# Basic syntax

Identifier in C#

- An identifier is a name used to identify a class, variable, function, or any user-defined item.
- A name must begin with a letter that can be followed by a sequence of letters, numbers (0-9) or underscores (_). 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#.

C# Basic syntax

Keywords in C#

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

Compile and execute C# Program

- C# is a programming language that is case sensitive.
- All commands and expressions must end with a semicolon (;).
- Program execution begins at the Main method.
- Unlike Java, program file names can be different from class names.

Datatypes

- Variables in C# are divided into the following types:
  - Value type
  - Reference type

Data type (value type)

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	-	-

Reference types in C#

- Reference types do not contain actual data stored in a variable, but they do contain a reference to the variables.
- In other words, they refer to a memory location.
- Examples of available reference types in C# are:
  - object, dynamic, and string…

Dynamic Type in C#

- You can store any type of value in a dynamic data type variable.
- Checking for these variable types takes place at run time.
- The syntax to declare a dynamic type in C# is:
  - dynamic <variable_name> = value;
- For example:
  - dynamic d = 20;
- Dynamic types are similar to object types

Convert data type

- Implicit (ngầm định) type casting:
  - This type casting is performed automatically by the C# compiler. When casting implicitly, the target data type must have a larger range of values than the source value, this is to ensure that there will be no data loss after casting..
- Ex:

```
int valueOne = 34;
 //ép kiểu từ int --> float
 //Compiler sẽ tự động ép kiểu
float valueTwo = valueOne;
```

Convert data type

- Explicit (tường minh) type casting: Changing a data type from a large value range to a data type with a smaller value range is called explicit casting.
- Syntax:
  - **datatype** name = (dataType) oldVariable;
- Note:
  - When using explicit type casting, it may lead to data loss.

Parse data from String

- To get the value of data in a string we use the Parse method, each of the basic data types of C# has this method.
- Parse is used to convert from string to basic data type.

```
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);
```

C# Structure

- Comparison operator

Tên toán tử	Ký hiệu	Biểu thức logic	Giá trị	Logic
and	&&	(x == 3) && (y ==7)	false	Cả hai điều kiệnphải đúng
or	||	(x == 3) || (y == 7)	true	Chỉ cần một điềukiện đúng
not	!	! (x == 3)	true	Biểu thức trongngoặc phải sai.

C# Structure

Comparison operator

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

If - Else Statement

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 bao va khoi tao bien
                int bienDem = 9 ;
                // Xuat ra man hinh
                if (bienDem % 2 == 0)
                        Console.WriteLine("{0} la so chan", bienDem) ;
                else    Console.WriteLine("{0} la so le", bienDem) ;
        }
}
```

If - Else Statement (cont…)

Cú pháp:

```
if (Điều_Kiện_1)
    <Khối lệnh 1>
else if (Điều_Kiện_2)
        <Khối lệnh 2.1>
    else
        <Khối lệnh 2.2>
```

Ví dụ :

```
using System;
class Thu_Trong_Tuan
{
    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
        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 nhat") ;
    }
}
```

Switch case statement

Cú pháp:

```
switch (Biểu_Thức)
{
    case <giá_trị_1>:
        < Khối lệnh 1>
        <Lệnh Nhảy>
    case <giá_trị_2>:
        < Khối lệnh 2>
        <Lệnh Nhảy>
    ....
    [default:
        < Khối 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;
        }
    }
}
```

While loop Statement

Cú pháp:

while (Điều_Kiện)
    < Khối lệnh>

Ví dụ 2.4:

```
using System;
class UsingWhile
{
        static void Main()
        {
                // Khai bao va khoi tao bien dem
                int i = 1 ;
                // Xuat ra man hinh
                while (i<=10)  {
                        Console.WriteLine("i = {0}",i) ;
                        i++ ; // tang bien dem,
                }
        }
}
```

Do- while loop Statement

Cú pháp:

```
do
        < Khối lệnh>
while (Điều_Kiện) ;
```

```
using System;
class UsingDoWhile
{
        static void Main()
        {
                // Khai bao va khoi tao bien dem
                int i = 1 ;
                // Xuat ra man hinh
                do  {
                        Console.WriteLine("i = {0}",i) ;
                        i++ ; // tang bien dem
                 } while (i<= 10) ;
        }
}
```

For loop statement

Cú pháp:

for ([Khởi_tạo] ; [Điều_kiện] ; [Bước_lặp])
    < Khối 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 statement

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) ;
        }
}
```

Homework

- Write a Hello world program (compile using command line).

THE END!

Q & A