



**TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN TIN HỌC VÀ KTTT**

LẬP TRÌNH NÂNG CAO

Giới thiệu môn học và ngôn ngữ lập trình

Giảng viên: Nguyễn Thị Phương Dung

Email: dungntp@tlu.edu.vn



GIỚI THIỆU MÔN HỌC

- ❖ Số tín chỉ: 3 tín chỉ
- ❖ Số tiết học: 30 tiết LT + 15 tiết TH
- ❖ Điều kiện dự thi kết thúc học phần:
 - Nghỉ học không quá 20% số tiết
 - Tham gia ít nhất 50% số bài kiểm tra và đạt
- ❖ Trọng số điểm môn học:
 - Điểm quá trình: 50%
 - Điểm thi kết thúc học phần: 50%



GIỚI THIỆU MÔN HỌC

Phương pháp đánh giá:

- ❖ Tổng điểm quá trình: 50%, trong đó:
 - Chuyên cần: 10%
 - ✓ Dựa vào số buổi học và ý thức của SV trong giờ học
 - Kiểm tra: 40%
 - ✓ 2 bài kiểm tra 60' khi kết thúc chương 2 và chương 4
- ❖ Thi cuối kỳ: 50%
 - Thi trên máy 60 -> 90' khi kết thúc môn học 1->2 tuần



GIỚI THIỆU MÔN HỌC

Nội dung chính của môn học:

- Chương 1: Kiểu dữ liệu và Cấu trúc lập trình trong C++
- Chương 2: Hàm
- Chương 3: Kiểu mảng, chuỗi ký tự và con trỏ
- Chương 4: Kiểu cấu trúc và hợp
- Chương 5: Vào ra dữ liệu



7/3/2023

Nguyễn Thị Phương Dung

Chương 1

Kiểu dữ liệu

và Cấu trúc lập trình trong C++

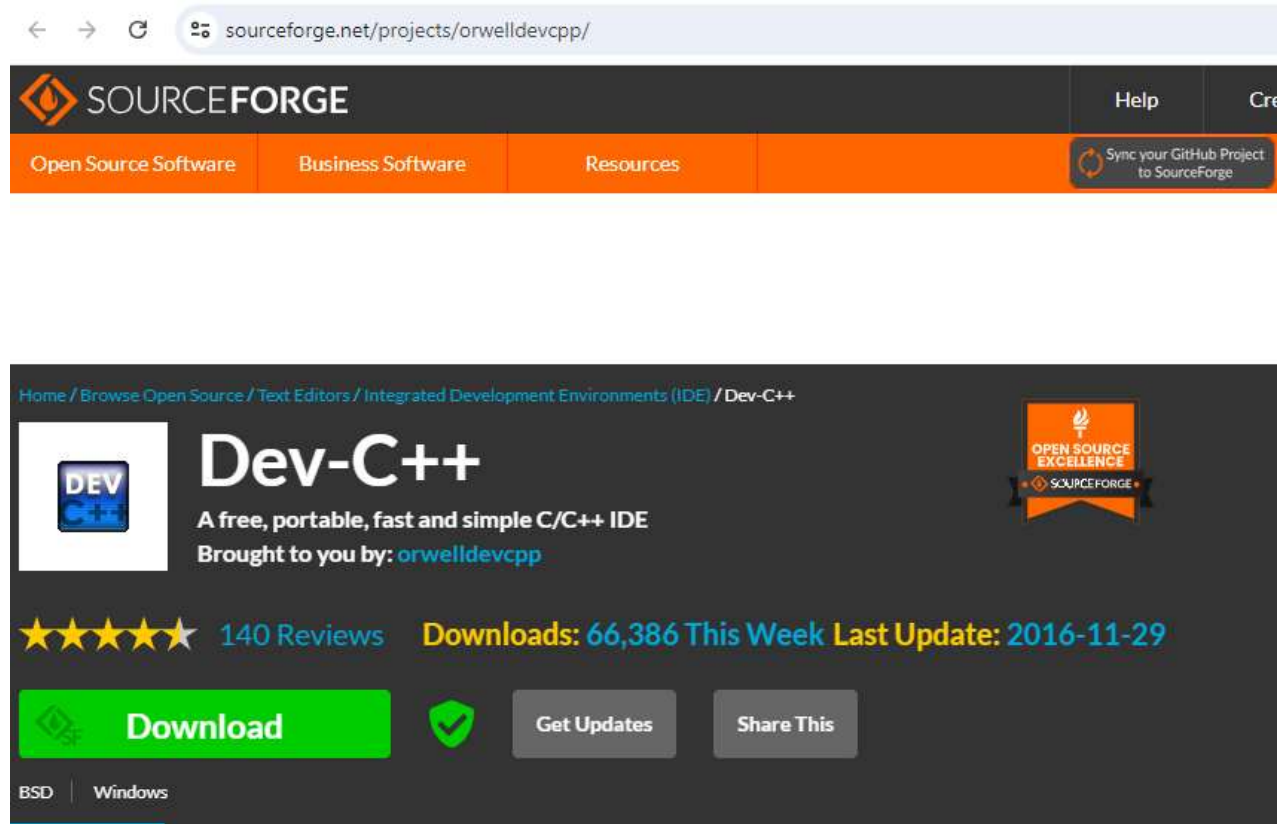


Ngôn ngữ lập trình C++

- C++ được phát triển vào năm 1979 bởi Bjarne Stroustrup tại Bell Labs nhằm mở rộng ngôn ngữ C
- Đạt chuẩn ISO vào năm 1998
- C++ là ngôn ngữ lập trình hướng đối tượng



Môi trường lập trình C++



The screenshot shows the SourceForge website for the Dev-C++ project. The browser address bar displays `sourceforge.net/projects/orwelldevcpp/`. The SourceForge logo is at the top left, with navigation links for "Open Source Software", "Business Software", and "Resources". A "Help" link and a "Create Account" button are on the right. A banner for "Sync your GitHub Project to SourceForge" is also present. The main content area features the Dev-C++ logo, the title "Dev-C++", and the description "A free, portable, fast and simple C/C++ IDE Brought to you by: [orwelldevcpp](#)". It includes a 5-star rating, "140 Reviews", and "Downloads: 66,386 This Week". The "Last Update" is listed as "2016-11-29". There are three buttons: a green "Download" button, a "Get Updates" button with a shield icon, and a "Share This" button. At the bottom left, it says "BSD | Windows".



7/3/2023

Nguyễn Thị Phương Dung

Môi trường lập trình C++

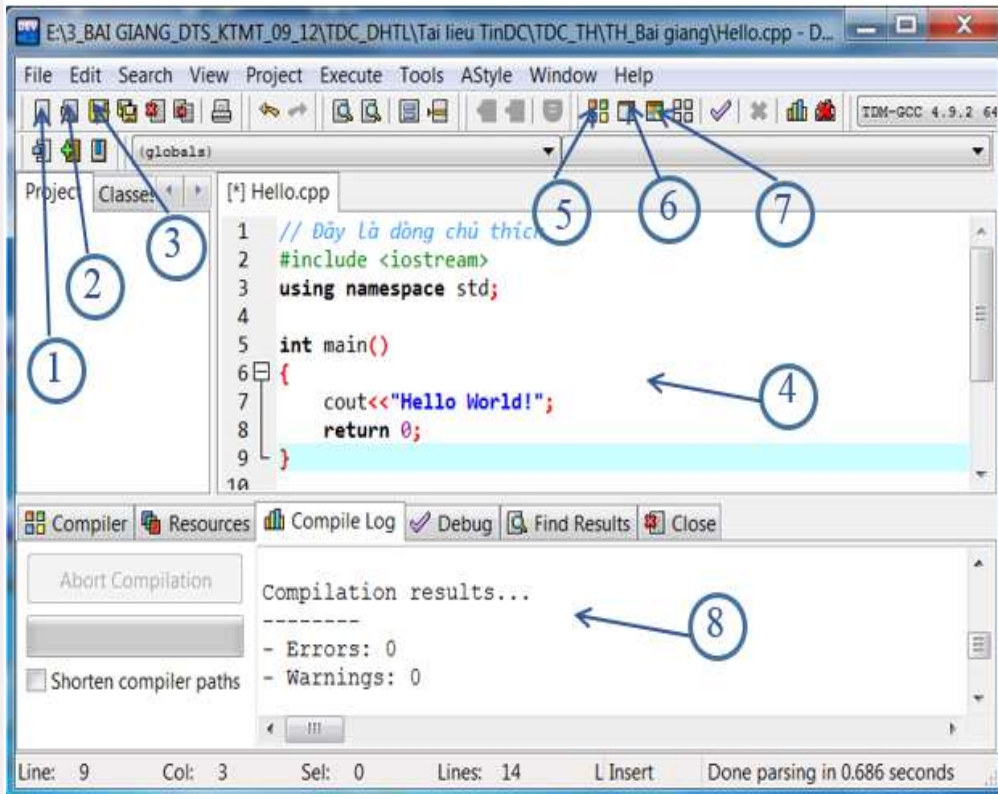
Dev-C++

- Hỗ trợ đầy đủ tính năng cho ngôn ngữ lập trình C/C++
- Hỗ trợ soạn thảo mã nguồn
- Sử dụng trình MinGW của GCC làm trình biên dịch
- Đường dẫn download
<https://sourceforge.net/projects/orwelldvcpp/files/latest/download>



Môi trường lập trình C++

➤ Giao diện Dev-C++



1. Tạo mới 1 file mã nguồn:

File -> New -> Source File (Ctrl + N)

2. Mở file mã nguồn đã tồn tại:

File -> Open (Ctrl + O)

3. Lưu nội dung file mã nguồn:

File -> Save (Ctrl + S)

4. Vùng soạn thảo

5. Biên dịch chương trình:

Execute -> Compile (F9)

6. Chạy chương trình:

Execute -> Run (F10)

7. Biên dịch và chạy chương trình:

Execute -> Compile & Run (F11)

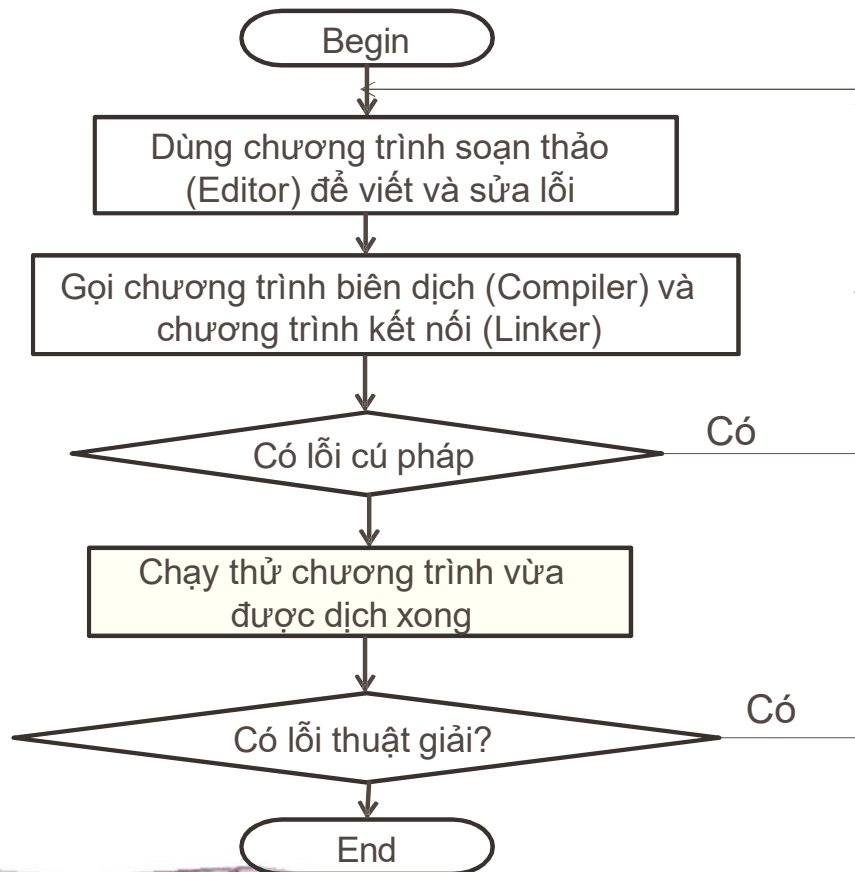
8. Vùng hiển thị kết quả biên dịch



7/3/2023

Nguyễn Thị Phương Dung

Các bước cơ bản để viết và chạy chương trình



- Editor là chương trình để viết mã nguồn
- Tên các tệp mã nguồn có dạng ***.cpp**



Hello.cpp

- Dịch mã nguồn ra tệp OBJ
- Liên kết các tệp OBJ qua Linker
- Kết quả trả ra file **.exe**



Hello.exe



7/3/2023

Nguyễn Thị Phương Dung

Cấu trúc chương trình

Hello.cpp

```
1 // Đây là dòng chú thích
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     cout<<"Hello World!";
8     return 0;
9 }
```



7/3/2023

Nguyễn Thị Phương Dung

Cấu trúc chương trình

- **Dòng 1:** Là dòng chú thích, bắt đầu bằng “//”
- **Dòng 2:** Khai báo sử dụng thư viện bắt đầu bằng dấu #
`#include <tênthưviện>`
- **Dòng 3:** Khai báo sử dụng không gian tên
`using namespace tên;`
- **Dòng 4:** Dòng trống
- **Dòng 5:** Khai báo hàm, *main* là hàm đặc biệt, gọi đầu tiên khi chương trình chạy
- **Dòng 6, 9:** Bắt đầu và kết thúc khối lệnh của hàm *main*
- **Dòng 7:** xuất xâu ký tự ra màn hình
- **Dòng 8:** Câu lệnh thoát khỏi hàm *main*



CÁC KHÁI NIỆM CƠ BẢN TRONG C++



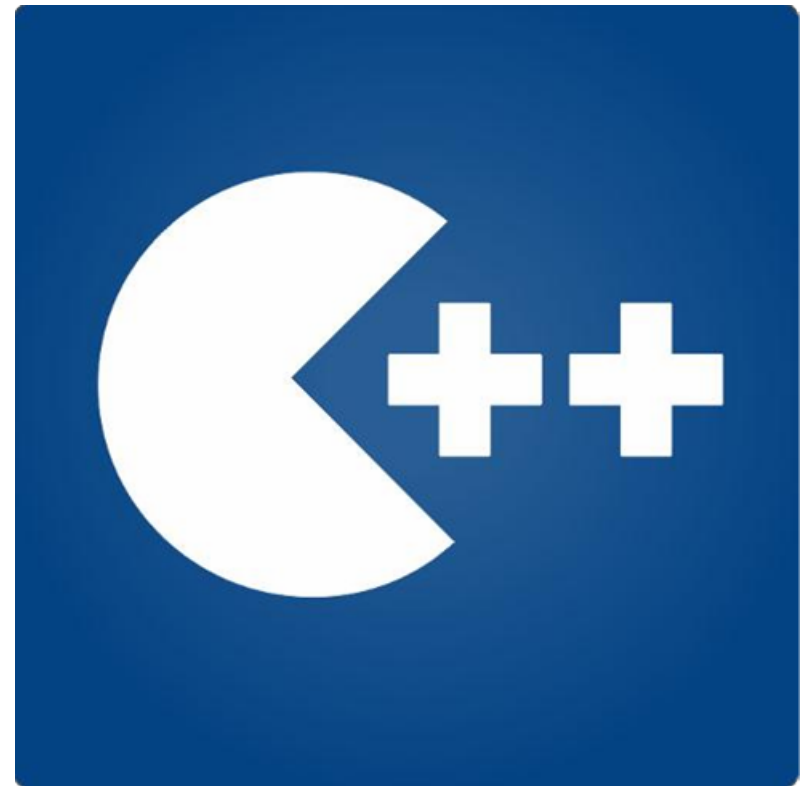
7/3/2023

Nguyễn Thị Phương Dung

13

CÁC KHÁI NIỆM CƠ BẢN TRONG C++

- Từ khóa
- Định danh
- Câu lệnh và khối lệnh
- Chú thích
- Các kiểu dữ liệu cơ sở
- Biến
- Hằng
- Toán tử
- Các hàm trong thư viện



TỪ KHÓA

- Các từ khóa cơ bản của ngôn ngữ, không thể định nghĩa lại

<code>alignas (since C++11)</code>	<code>enum</code>	<code>return</code>
<code>alignof (since C++11)</code>	<code>explicit</code>	<code>short</code>
<code>and</code>	<code>export(1)</code>	<code>signed</code>
<code>and_eq</code>	<code>extern</code>	<code>sizeof</code>
<code>asm</code>	<code>false</code>	<code>static</code>
<code>auto(1)</code>	<code>float</code>	<code>static_assert (since C++11)</code>
<code>bitand</code>	<code>for</code>	<code>static_cast</code>
<code>bitor</code>	<code>friend</code>	<code>struct</code>
<code>bool</code>	<code>goto</code>	<code>switch</code>
<code>break</code>	<code>if</code>	<code>template</code>
<code>case</code>	<code>inline</code>	<code>this</code>
<code>catch</code>	<code>int</code>	<code>thread_local (since C++11)</code>
<code>char</code>	<code>long</code>	<code>throw</code>
<code>char16_t (since C++11)</code>	<code>mutable</code>	<code>true</code>
<code>char32_t (since C++11)</code>	<code>namespace</code>	<code>try</code>
<code>class</code>	<code>new</code>	<code>typedef</code>
<code>compl</code>	<code>noexcept (since C++11)</code>	<code>typeid</code>
<code>const</code>	<code>not</code>	<code>typename</code>
<code>constexpr (since C++11)</code>	<code>not_eq</code>	<code>union</code>
<code>const_cast</code>	<code>nullptr (since C++11)</code>	<code>unsigned</code>
<code>continue</code>	<code>operator</code>	<code>using(1)</code>
<code>decltype (since C++11)</code>	<code>or</code>	<code>virtual</code>
<code>default(1)</code>	<code>or_eq</code>	<code>void</code>
<code>delete(1)</code>	<code>private</code>	<code>volatile</code>
<code>do</code>	<code>protected</code>	<code>wchar_t</code>
<code>double</code>	<code>public</code>	<code>while</code>
<code>dynamic_cast</code>	<code>register</code>	<code>xor</code>
<code>else</code>	<code>reinterpret_cast</code>	<code>xor_eq</code>



ĐỊNH DANH

- Là đặt tên cho các đối tượng trong chương trình
- Các đối tượng:
 - Biến
 - Hằng số
 - Hàm
 - Kiểu dữ liệu
- Quy tắc đặt tên định danh:
 - ✓ Bắt đầu bằng các chữ cái (**a..z, A..Z**) hoặc “_”
 - ✓ Có thể chứa chữ số (**0..9**) hoặc kí tự “_”
 - ✓ C++ phân biệt chữ hoa, chữ thường
 - ✓ **Không trùng với các từ khóa C++**
 - ✓ Không chứa khoảng trống ở giữa
 - ✓ Không chứa các ký tự đặc biệt

(Xem mục 2.2 trong giáo trình)





Định danh nào sau đây là đúng/sai? Vì sao?

1. CamelCase
2. camelCase
3. PI
4. 20_tuoi
5. return
6. nghiem1
7. m&m
8. chieu cao



ĐỊNH DANH

- Quy tắc con lặc đà:

- **Tên biến, hàm:** viết thường từ đầu tiên, viết hoa chữ cái đầu từ tiếp theo
- **Tên hằng số:** viết hoa toàn bộ, sử dụng dấu gạch dưới để phân cách
- **Kiểu dữ liệu mới:** viết hoa chữ cái đầu tiên của tất cả các từ

Định danh	Loại đối tượng
HANG_SO	Hằng số
a,b, i,j, chieuDai, chieuRong	Biến
tinhTong, tinhTrungBinh	Hàm
SinhVien, KhachHang	Kiểu dữ liệu mới



LỆNH, KHỐI LỆNH

▪ Câu lệnh:

- Câu lệnh là thực hiện một chỉ thị
- Kết thúc bằng dấu chấm phẩy “;”

```
// Đây là dòng chú thích
#include <iostream>
using namespace
std;

int main ()
{
    cout << "Đây là chương trình
    C++"; return 0;
}
```

▪ Khối lệnh:

- Tập hợp nhiều câu lệnh
- Nhóm trong cặp dấu ngoặc {}
- Các câu lệnh có thể viết trên cùng một dòng
- **Các câu lệnh thực hiện tuần tự từ trên xuống dưới. Từ trái qua phải**

```
// Đây là dòng chú thích
#include <iostream>
using namespace std;

int main ()
{
    cout << "Đây là chương trình
    C++"; return 0;
}
```



CHÚ THÍCH

- Là lời giải thích vắn tắt cho câu lệnh hay đoạn chương trình
- Không ảnh hưởng đến kết quả thực thi của chương trình

- **Cách viết chú thích:**

- Chú thích một dòng: sử dụng **//**
- Chú thích nhiều dòng: sử dụng **/*** và ***/**

- **Ví dụ:**

```
/* Chương trình tính diện tích tam giác
   Nhập ba giá trị cho ba cạnh tam giác
   Tính diện tích tam giác và hiển thị kết quả */

#include <iostream>
using namespace std;

int main ()
{
    //Các câu lệnh bắt đầu từ đây
}
```



CÁC KIỂU DỮ LIỆU CƠ SỞ

Kiểu dữ liệu	Mô tả	Cỡ	Dải dữ liệu
char	Kiểu kí tự	1byte	signed: -128 to 127 unsigned: 0 to 255
short int(short)	Kiểu số nguyên	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Kiểu số nguyên	4bytes	signed: -2147483648 -> 2147483647 unsigned: 0 to 4294967295
long int	Kiểu số nguyên	4bytes	signed: -2147483648 -> 2147483647 unsigned: 0 to 4294967295
bool	Kiểu logic	1byte	true or false
float	Kiểu số thực	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Kiểu số thực	8bytes	+/- 1.7e +/- 308 (~15 digits)

CÁC KIỂU DỮ LIỆU CƠ SỞ

▪ Ví dụ:

➤ Ngày, tháng, năm: **int, short int**

- ngày: 12, tháng: 1, năm: 2014

➤ Số sinh viên trong lớp: **int, short int**

- 40

➤ Chiều dài, chiều rộng của hình chữ nhật: **float, double**

- chiều dài: 23.123 (cm), chiều rộng: 12.243 (cm)

➤ Các chữ cái: **char**

- 'a', 'b', 'A', ..., 'Z'

➤ Các chữ số: **char**

- '0', '1', ... '9'



BIẾN

- Biến là nơi để chứa dữ liệu
- Mọi biến đều có tên và kiểu dữ liệu

- **Cú pháp khai báo:**

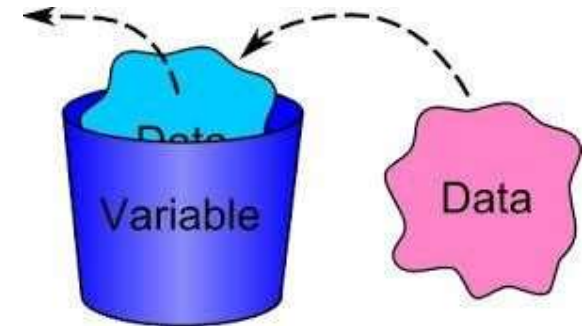
- Cách 1: **KieuDulieu** **tenBien**;

- Cách 2: **KieuDulieu** **tenBien1, tenBien2**;

- Cách 3: **KieuDulieu** **tenBien=GiaTri**;

KieuDulieu **tenBien(giaTri)**;

KieuDulieu **tenBien1=giaTri1,tenBien2 = giaTri2**;



*Mọi biến phải được khai báo trước khi sử dụng
Giá trị truyền vào cho biến phải phù hợp với kiểu dữ liệu khai báo*

BIẾN

- Phạm vi sử dụng của biến:

```
#include <iostream>
using namespace std;
int main ()
{
    int a;
    cin>>a; //Nhap gia tri cho a tu ban phim
    int b = a*2;
    if(a>0)
    {
        int c= b + a;
    }
    cout<< c;
    return 0;
```

Chưa khai báo ở phạm vi ngoài



HẰNG SỐ

- Hằng số chứa giá trị không đổi trong chương trình
- Giá trị của hằng số phải xác định ngay khi khai báo

- Khai báo:

➤ Cách 1: Dùng từ khóa tiền xử lý `#define`

- Cú pháp: `#define TEN_HANG_SO giatri`

➤ Cách 2: Dùng từ khóa `const`

- Cú pháp: `const kiểudulieu TEN_HANG_SO = giatri;`

Không có dấu "=" và dấu ";"



HÀNG SỐ

▪ Ví dụ:

```
#include <iostream>
using namespace std;
#define GIA_TOC 9.81
int main ()
{
    const double V0 = 1.5;
    double t;
    cout<<"Nhập thời gian t=";
    cin>>t;
    cout<<" Vận tốc tại thời điểm "<<t<<" là : "<<V0 + GIA_TOC*t;
    return 0;
}
```

Máy sẽ tự thay từ GIA_TOC bằng 9.81



BIỂU THỨC VÀ TOÁN TỬ

▪ **Biểu thức (expression):** là tổ hợp các **toán tử** và **toán hạng**

- **Toán tử:** là các phép toán
- **Toán hạng:** là các biến, hằng số, hàm, giá trị cụ thể...

▪ **Ví dụ:**

$7 + 2 * 5$
 $2 * r * PI$
 $a * (b + c / d)$
 $5 + (c = 3 + 8)$

▪ **Toán tử bao gồm:**

- Toán tử gán
- Toán tử số học
- Toán tử logic

- Toán tử so sánh
- Toán tử điều kiện



TOÁN TỬ GÁN

- Là phép gán giá trị, biểu thức, hằng số, hàm vào biến
- Dạng thức:

```
tenBien = Bieu_thuc;
```

- Ví dụ:

```
x = 100;
```

```
y = PI * r * r;
```

```
z = x / 2;
```

```
s = "ABC";
```

```
a = 'A';
```

```
c = true;
```

- Gán liên tiếp: *nhiều biến gán cùng một giá trị.*

```
x = y = z = 200;
```



TOÁN TỬ GÁN

- Phép gán mở rộng:

$x += y$	\leftrightarrow	$x = x + y$
$x -= y$	\leftrightarrow	$x = x - y$
$x *= y$	\leftrightarrow	$x = x * y$
$x /= y$	\leftrightarrow	$x = x / y$
$x \% = y$	\leftrightarrow	$x = x \% y$





Giá trị cuối cùng của **a**, **b** là bao nhiêu?

```
// Ví dụ về phép gán
#include <iostream>
using namespace std;

int main ()
{
    int a, b;           // a:?, b:?
    a = 10;             // a:?, b:?
    b = 4;              // a:?, b:?
    a = b;              // a:?, b:?
    b += a;             // a:?, b:?
    a += 7;             // a:?, b:?

    cout << "a:"<< a;
    cout << " b:" << b;
}
```

TOÁN TỬ SỐ HỌC

▪ Toán tử hai ngôi:

Toán tử	Chức năng	Ví dụ
+	Cộng	$x = a + 2;$
-	Trừ	$a = 7 - 5;$
*	Nhân	$b = 7 * 5;$
/	Chia	$c = 5 / 2;$ $d = 5 / 2.0;$
%	Lấy số dư của phép chia với số nguyên	$e = 4 \% 3; // e = 1$ $e = 4 \% 3.0; // \text{Lỗi}$



Chú ý với phép chia với số nguyên



TOÁN TỬ SỐ HỌC

▪ Toán tử một ngôi:

Toán tử	Chức năng	Ví dụ
-	Lấy số đối	$x = -20;$ $y = -x;$
++	Tăng một giá trị	$x++; \quad // x = x + 1$ $++y; \quad // y = y + 1$
--	Giảm một giá trị	$x--; \quad // x = x - 1$ $--y; \quad // y = y - 1$



TOÁN TỬ SỐ HỌC

- **Toán tử một ngôi – tăng/giảm tiền tố:** tăng hoặc giảm giá trị **trước** khi sử dụng giá trị của toán hạng

- Ví dụ:

```
a = 10;  
b = 5;  
c = a * ++b;    // c= ?
```

- **Toán tử một ngôi – tăng/giảm hậu tố:** tăng hoặc giảm giá trị **sau** khi sử dụng giá trị của toán hạng

- Ví dụ:

```
a = 10;  
b = 5;  
c = a * b++;    // c= ?
```



TOÁN TỬ LOGIC

Toán tử	Chức năng	Ví dụ
&&	Trả kết quả là True khi cả 2 toán hạng đều là True	((5 == 5) && (3 > 6)) // trả về false (true && false)
 	Trả về kết quả là True khi chỉ một trong 2 toán hạng là True	((5 == 5) (3 > 6)) // trả về true (true false)
!	Chuyển đổi giá trị từ True thành False và ngược lại	!(5 == 5) // trả về false !(6 <= 4) // trả về true !true // trả về false !false // trả về true



TOÁN TỬ ĐIỀU KIỆN BA NGÔI

- Cú pháp:

bieuthucdieukien ? Bieuthuc1 : Bieuthuc2

- Nếu biểu thức điều kiện là **True** thì trả về biểu thức 1, ngược lại trả về biểu thức 2

- Ví dụ:

(7==5) ? 4 : 3 // trả về ?

(7==5+2) ? 4 : 3 // trả về ?

(5>3) ? a : b // ?

(a>b) ? a : b // ?



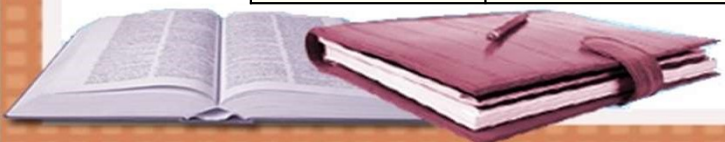
TOÁN TỬ QUAN HỆ

Toán tử	Mô tả	Ví dụ
==	So sánh bằng	$7 == 5$ // trả về false $(b=2) == 5$ // trả về false
!=	Khác	$(3 != 2)$ // trả về true
<	Nhỏ hơn	$(5 < 5)$ // trả về false
>	Lớn hơn	$(3 > 2)$ // trả về true
<=	Nhỏ hơn hoặc bằng	$(6 <= 6)$ // trả về true
>=	Lớn hơn hoặc bằng	$(6 >= 4 + 2)$ // trả về true



TOÁN TỬ BIT

Toán tử	Mô tả	Ví dụ	
&	AND	$0 \& 0 = 0$ $1 \& 0 = 0$	$0 \& 1 = 0$ $1 \& 1 = 1$
 	OR	$0 0 = 0$ $1 0 = 1$	$0 1 = 1$ $1 1 = 1$
^	XOR	$0 \wedge 0 = 0$ $1 \wedge 0 = 1$	$0 \wedge 1 = 1$ $1 \wedge 1 = 0$
~	NOT	$\sim 0 = 1$ $\sim 1 = 0$	
<<	Dịch bit sang trái	$0001 \ll 1 = 0010$	
>>	Dịch bit sang phải	$1000 \gg 1 = 0100$	



ĐỘ ƯU TIÊN CỦA TOÁN TỬ

STT	Toán tử	Mô tả
1	++ -- ()	Tăng, giảm hậu tố, dấu ngoặc
2	++ -- ~ !	Tăng, giảm tiền tố, NOT bit, phủ định
3	* /	Nhân, chia
4	+ -	Cộng, trừ
5	<< >>	Dịch bit sang trái, Dịch bit sang phải
6	< > <= >=	So sánh hơn kém
7	== !=	So sánh bằng, khác
8	&	AND bit

STT	Toán tử	Mô tả
9	^	XOR
10	 	OR
11	&&	Toán tử logic AND
12	 	Toán tử logic OR
13	= *= /= %= += -=	Toán tử gán



CÁC HÀM TOÁN HỌC

(Xem mục 2.5.1 trong giáo trình)

- Thêm khai báo **#include <cmath>** trên phần tiền xử lý
- Hàm lượng giác:

Hàm	Mô tả
cos (x)	Tính cosin của một góc x đo bằng radian
sin (x)	Tính sin của một góc x đo bằng radian
tan (x)	Tính tag của một góc x đo bằng radian
acos (x)	Tính acos trả ra giá trị radian
asin(x)	Tính asin trả ra giá trị radian
atan(x)	Tính actag trả ra giá trị radian

cmath



CÁC HÀM TOÁN HỌC

▪ Ví dụ:

```
/* cos example */
#include <iostream>
#include <cmath>          /* cos */
using namespace std;

#define PI 3.14159265

int main ()
{
    double param, result;
    param = 60.0;
    result = cos ( param * PI / 180.0 );
    cout<<"cosin cua goc "<<param<< " do la "<< result;
    return 0;
}
```



CÁC HÀM TOÁN HỌC

▪ Hàm mũ và logarit:

Hàm	Mô tả
exp(x)	Tính e^x , x có kiểu double , float
log (x)	Tính logarit cơ số e của x, x có kiểu double , float
log10 (x)	Tính logarit cơ số 10 của x, x có kiểu double , float

▪ Hàm lũy thừa, căn, lấy giá trị tuyệt đối:

Hàm	Mô tả
pow(x, y)	Tính x^y , x,y có kiểu double , float
sqrt(x)	Tính căn bậc 2 của x, x có kiểu double , float
cbrt (x)	Tính căn bậc 3 của x, x có kiểu double , float
abs(x)	Lấy giá trị tuyệt đối,



CÁC HÀM TOÁN HỌC

▪ Ví dụ:

$$e^3 \quad \longrightarrow \quad \text{exp}(3)$$

$$\log 10 \quad \longrightarrow \quad \text{log}(10)$$

$$2^5 \quad \longrightarrow \quad \text{pow}(2, 5)$$

$$2^{\frac{1}{3}} \quad \longrightarrow \quad \text{pow}(2, 1/3.0)$$

$$\sqrt{9} \quad \longrightarrow \quad \begin{array}{l} \text{pow}(9, 1/2.0) \\ \text{sqrt}(9) \end{array}$$

CÁC HÀM TOÁN HỌC

▪ Hàm làm tròn:

Hàm	Mô tả
ceil(x)	Trả về số nguyên nhỏ nhất không nhỏ hơn x (làm tròn lên), x có kiểu double , float
floor(x)	Trả về số nguyên lớn nhất không lớn hơn x (làm tròn xuống), x có kiểu double , float

▪ Ví dụ:

```
ceil(2.2) // = ?  
ceil(3.7) // = ?  
ceil(-2.2) // = ?  
ceil(-3.7) // = ?
```



NHẬP XUẤT DỮ LIỆU TRONG C++



7/3/2023

Nguyễn Thị Phương Dung



NHẬP XUẤT DỮ LIỆU

- C++ sử dụng lớp trừu tượng stream (dòng) để vào ra dữ liệu
- Dòng là chuỗi các bytes dữ liệu
- **Dòng nhập:** là chuỗi bytes từ thiết bị bàn phím, ổ cứng, mạng
- **Dòng xuất:** là chuỗi các bytes đi ra màn hình, máy in, ổ cứng, mạng...
- **Các thư viện hỗ trợ nhập/xuất:**
 - **iostream:** Định nghĩa các đối tượng **cin, cout**
 - **fstream:** Cung cấp các hàm làm việc với tệp tin



NHẬP XUẤT DỮ LIỆU

- Để nhập dữ liệu từ bàn phím và xuất dữ liệu ra màn hình cần khai báo thư viện iostream bằng câu lệnh sau:

#include <iostream>

- Nên khai báo không gian tên std để sử dụng các đối tượng cin, cout được thuận tiện:

using namespace std;



XUẤT DỮ LIỆU

- Sử dụng đối tượng **cout** và toán tử **<<** để xuất dữ liệu ra màn hình
- Sau toán tử **<<** là biến, biểu thức, xâu ký tự, hàm
- **Cú pháp:**
 - Cách 1: **cout<<** biethuc;
 - Cách 2: **cout<<** biethuc1 **<<** biethuc2;
- Toán tử **<<** có thể dùng nhiều lần trên một câu lệnh.



Khi xuất xâu ký tự ra màn hình phải đặt xâu đó trong dấu ""



XUẤT DỮ LIỆU

- Sử dụng **endl** để thêm dòng mới
- Với chuỗi ký tự:
 - sử dụng “**\n**” tại vị trí muốn xuống dòng
 - sử dụng “**\t**” để cách ra một khoảng trống



XUẤT DỮ LIỆU

▪ Ví dụ:

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string str = "DAI HOC THUY LOI";
    cout << "Chao mung ban den voi " << str << endl;
    int n = 2023;
    cout << "Day la nam hoc " << n - 1 << "-" << n << "\nHoc ky II \n";
}
```

```
Chao mung ban den voi DAI HOC THUY LOI
Day la nam hoc 2022-2023
Hoc ky II
```





ĐOẠN LỆNH SAU CHO NHỮNG GÌ TRÊN MÀN HÌNH?

```
const double PI = 3.1415926;  
double x= 30;  
cout<<"sin cua goc "<<x<<" do la "<<sin(PI*x/180);
```

```
cout<<"DAI HOC THUY LOI\nKHOA CONG NGHE THONG TIN" <<endl;  
cout<<"*****\n"<<endl;  
cout<<"MON TIN HOC DAI CUONG \n"<<endl<<"*****";
```





ĐOẠN LỆNH SAU CHO NHỮNG GÌ TRÊN MÀN HÌNH?

```
int x = 3, y=5;  
int z = 3*x + y;  
cout<<"z="<<z<<endl;  
cout<<" (x+y)/7="<<(x+y)/7<<endl;  
cout<<"(x+y)/z="<<(x+y)/z<<endl;  
cout<<"(x+y)/z="<<(x+y)/z*1.0<<endl;
```

```
int a=10;  
double f=(9/5) * c + 32.0; cout<<" f="<<f<<endl;  
f=(9.0/5) * c + 32.0; cout<<" f="<<f<<endl;
```



NHẬP DỮ LIỆU TỪ BÀN PHÍM

- Sử dụng đối tượng **cin** và toán tử **>>** để nhập dữ liệu
- Sau toán tử **>>** là biến
- Cú pháp:
 - Cách 1: **cin>>** tenBien;
 - Cách 2: **cin>>** tenBien1 **>>** tenBien2;
- **cin** nhập cho đến khi ấn phím xuống dòng, phím tab hoặc phím space



NHẬP DỮ LIỆU TỪ BÀN PHÍM

- Ví dụ 1: Nhập độ dài cạnh hình vuông và tính diện tích.

➤ Đầu vào: độ dài cạnh hình vuông → kiểu dữ liệu?

➤ Đầu ra: diện tích hình vuông → kiểu dữ liệu?

```
#include <iostream>
using namespace std;

int main()
{
    double a;
    cin>>a; //nhap du lieu cho canh hinh vuong luu vao bien a

    double s; //luu dien tich hinh vuong
    s = a*a;
    cout<<"dien tich hinh vuong la:"<<s;
    return 0;
}
```



NHẬP DỮ LIỆU TỪ BÀN PHÍM

▪ Ví dụ 2: Nhập ba điện trở và tính điện trở tương đương.

➤ Đầu vào: ba điện trở → kiểu dữ liệu?

➤ Đầu ra: giá trị điện trở tương đương → kiểu dữ liệu?

```
#include <iostream>
using namespace std;
int main()
{
    double r1, r2, r3;
    cout<<"Nhập giá trị cho 3 điện trở \n";
    cout<<"R1 = "; cin>>r1;
    cout<<"R2 = "; cin>>r2;
    cout<<"R3 = "; cin>>r3;

    double r = 1/(1/r1+1/r2+1/r3);
    cout<<"Điện trở tương đương R="<<r;
    return 0;
}
```


NHẬP DỮ LIỆU TỪ BÀN PHÍM



Trường hợp nhập nhiều biến trên cùng một lệnh, giá trị nhập cách nhau bởi dấu cách

▪ Ví dụ:

```
int x,y;  
cout<<"Nhập vào 2 số: ";  
cin>>x>>y;  
cout<<"Tổng 2 số là: "<<x+y;
```

Khai báo nhiều biến cùng kiểu
cách nhau dấu,

```
Nhập vào 2 số: 2 3  
Tổng 2 số là: 5
```

Nhập dữ liệu nhiều biến
cách nhau khoảng trống



NHẬP DỮ LIỆU TỪ BÀN PHÍM



- Chú ý đến kiểu dữ liệu khi nhập đầu vào
- Khi nhập sai kiểu dữ liệu chương trình không báo lỗi nhưng sẽ cho giá trị không đúng.

▪ Ví dụ:

```
int x;  
cout<<"Nhập x";  
cin>>x;  
cout<<"x="<<x;
```

```
Nhập x= 3.14  
x=3
```



NHẬP DỮ LIỆU TỪ BÀN PHÍM



- Trường hợp số giá trị nhập vào nhiều hơn số biến thì máy sẽ lưu vào bộ đệm để cho lần nhập biến tiếp theo

▪ Ví dụ:

```
int x,y;  
cout<<"Nhập x=";  
cin>>x;  
cout<<"Nhập y=";  
cin>>y;  
cout<<"Bạn vừa nhập x="<<x<<" , y= " <<y;
```

Nhập x= 2 8

Nhập y=Bạn vừa nhập x=2, y=8

Tự động gán giá trị 8 cho biến y ngay khi gõ enter



NHẬP XÂU KÝ TỰ

- Toán tử **>>** nhập xâu cho đến khi gặp khoảng trống → chỉ nhập được một từ

- **Cách khác:**

- Sử dụng hàm **getline** trong thư viện **string**
- Phải khai báo thư viện

#include <string>

- Khai báo biến thuộc kiểu dữ liệu **string**
- Gọi hàm **getline** trong câu lệnh

getline(cin, tenbien);



NHẬP XÂU KÝ TỰ

▪ Ví dụ:

Thêm khai báo
thư viện **string**

Khai báo biến
kiểu **string**

Gọi hàm **getline**

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string diachi;
    cout << "Chao ban! ban tu dau den?"<<endl;
    getline(cin,diachi);
    cout <<"Minh cung den tu "<< diachi << endl;
}
```



Tránh trôi lệnh nhập string

▪ Dùng lệnh fflush(stdin)

Thêm khai báo
thư viện **string**

```
#include <iostream>
#include <string>
using namespace std;
```

Khai báo biến
kiểu **string**

```
int main()
{
    string diachi;
    cout << "Chao ban! ban tu dau den?"<<endl;
    fflush(stdin);
    getline(cin,diachi);
    cout <<"Minh cung den tu "<< diachi << endl;
}
```

Gọi hàm **getline**





TÌM LỖI SAI TRONG CÁC CÂU LỆNH SAU

1. `cout<<"sin cua goc "x<<" do la "<<sin(PI*x/180);`
2. `cou<<"Dien tich hinh vuong canh "<<x<<" =<<x*x<<end;`
3. `double x,y;`
`cin>>x,y;`
4. `int x,y;`
`cin>>x+y;`
`cin>>"x";`
5. `cout<<"Tuoi cua ban la <<"x;`





7/3/2023

Nguyễn Thị Phương Dung