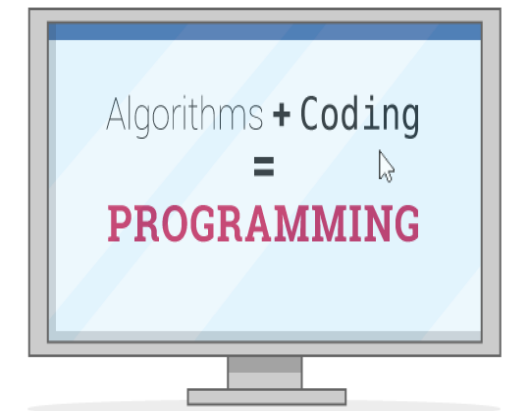
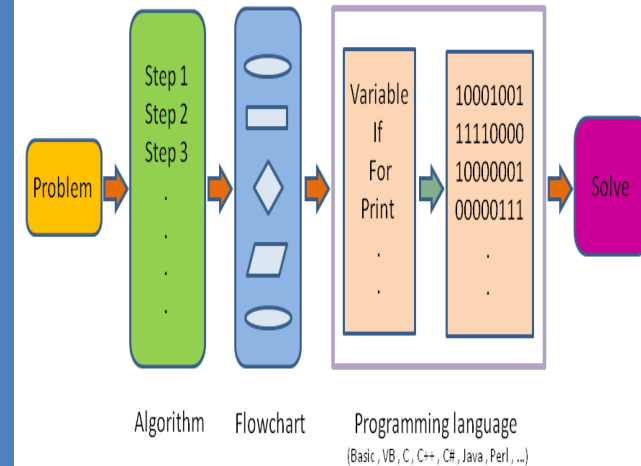


# NHẬP MÔN LẬP TRÌNH

## Chương 2 – CÁC THÀNH PHẦN CƠ BẢN CỦA NGÔN NGỮ LẬP TRÌNH

GV: Lê Thị Bích Hằng

Trường Đại học Nha Trang



# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở



Biến  
Hằng  
Biểu thức  
Phép toán  
Câu lệnh  
Một số hàm toán học  
Tài liệu tham khảo

## ■ Ngôn ngữ C có 4 kiểu dữ liệu cơ sở:

- **Kiểu số nguyên**: giá trị của nó là các số nguyên như 237, -706, ...
- **Kiểu số thực**: giá trị của nó là các số thực như 3.1415, 29.12, -17.06, ...
- **Kiểu boolean**: giá trị đúng hoặc sai.
- **Kiểu ký tự**: gồm 256 ký tự trong bảng mã ASCII.

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến  
Hằng  
Biểu thức  
Phép toán  
Câu lệnh  
Một số hàm toán học  
Tài liệu tham khảo



## ■ Kiểu số nguyên

### • Kiểu số nguyên có dấu (signed integer)

- Kiểu số nguyên có dấu biểu diễn số nguyên, bao gồm cả số âm và số không âm (chẳng hạn 3, -15,...).
- Dùng n bits để biểu diễn số nguyên có dấu trên máy tính -> miền giá trị biểu diễn được là  $-2^{n-1} \dots +2^{n-1}-1$

| Kiểu<br>(Type) | Độ lớn<br>(Byte) | Miền giá trị<br>(Range)           |
|----------------|------------------|-----------------------------------|
| char           | 1                | -128 ... +127                     |
| short          | 2                | -32.768 ... +32.767               |
| int            | 4                | -2.147.483.648 ... +2.147.483.647 |
| long           | 4                |                                   |

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến  
Hằng  
Biểu thức  
Phép toán  
Câu lệnh  
Một số hàm toán học  
Tài liệu tham khảo



## ■ Kiểu số nguyên

- Kiểu số nguyên không dấu (unsigned integer)
  - Dùng n bits để biểu diễn số nguyên không âm trên máy tính -> miền giá trị biểu diễn được là:  $0 \dots 2^{n-1}$

| Kiểu<br>(Type) | Độ lớn<br>(Byte) | Miền giá trị<br>(Range) |
|----------------|------------------|-------------------------|
| unsigned char  | 1                | 0 ... 255               |
| unsigned short | 2                | 0 ... 65.535            |
| unsigned int   | 4                | 0 ... 4.294.967.295     |
| unsigned long  | 4                |                         |

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến  
Hằng  
Biểu thức  
Phép toán  
Câu lệnh  
Một số hàm toán học  
Tài liệu tham khảo



## ■ Kiểu số thực (floating – point)

- Ví dụ:

$$17.06 = 1.706 * 10 = 1.706 * 10^1$$

| Kiểu<br>(Type) | Độ lớn<br>(Byte) | Miền giá trị<br>(Range)                |
|----------------|------------------|--|
| float (*)      | 4                | $3.4 * 10^{-38} \dots 3.4 * 10^{38}$   |
| double (**)    | 8                | $1.7 * 10^{-308} \dots 1.7 * 10^{308}$ |

- (\*) Độ chính xác đơn (Single-precision) chính xác đến 7 số lẻ.
- (\*\*) Độ chính xác kép (Double-precision) chính xác đến 19 số lẻ.

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến  
Hằng  
Biểu thức  
Phép toán  
Câu lệnh  
Một số hàm toán học  
Tài liệu tham khảo



## ■ Kiểu boolean

- C ngầm định một cách không tường minh:
  - **false** (sai): giá trị 0.
  - **true** (đúng): giá trị khác 0, thường là 1.
- Ngôn ngữ C++ bổ sung kiểu **bool** để định nghĩa tường minh kiểu boolean.
- Ví dụ
  - 0 (false),
  - 1 (true), 2 (true), 2.5 (true)
  - $1 > 2$  (0, false)
  - $1 < 2$  (1, true)

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến  
Hằng  
Biểu thức  
Phép toán  
Câu lệnh  
Một số hàm toán học  
Tài liệu tham khảo



## ■ Kiểu ký tự

- Tên kiểu: **char**
- Miền giá trị: 256 ký tự trong bảng mã ASCII.
- Ký tự được lưu trong máy tính bằng mã ASCII của nó.
- Ví dụ:
  - Ký tự 'A' được lưu thành số 65
  - Ký tự 'a' được lưu thành số 97.

| ASCII-Zeichensatz |   |   |   |   |   |   |    |   |   |   |
|-------------------|---|---|---|---|---|---|----|---|---|---|
| +                 | 0 | 1 | 2 | 3 | 4 | 5 | 6  | 7 | 8 | 9 |
| 30                |   |   |   | ! | " | # | \$ | % | & | ' |
| 40                | ( | ) | * | + | , | - | .  | / | 0 | 1 |
| 50                | 2 | 3 | 4 | 5 | 6 | 7 | 8  | 9 | : | ; |
| 60                | < | = | > | ? | @ | A | B  | C | D | E |
| 70                | F | G | H | I | J | K | L  | M | N | O |
| 80                | P | Q | R | S | T | U | V  | W | X | Y |
| 90                | Z | [ | \ | ] | ^ | _ | `  | a | b | c |
| 100               | d | e | f | g | h | i | j  | k | l | m |
| 110               | n | o | p | q | r | s | t  | u | v | w |
| 120               | x | y | z | { |   | } | ~  |   |   |   |

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Biến (Variable)

- Một biến là một vùng nhớ được đặt tên, dùng để lưu trữ dữ liệu trong quá trình chương trình hoạt động.
- Giá trị của biến **có thể thay đổi**.
- Trong ngôn ngữ C, một biến có **kích thước và kiểu dữ liệu nhất định**; từ đó xác định được miền giá trị mà biến có thể lưu trữ, cũng như các thao tác có thể thực hiện được với biến này.
- Trong C, **phải khai báo biến trước khi sử dụng**.
- Khai báo biến:

```
<kiểu dữ liệu> <tên biến>;
```

```
<kiểu dữ liệu> <tên biến> = <giá trị>;
```

```
<kiểu dữ liệu> <tên biến 1>, <tên biến 2>;
```

- Ví dụ:

```
int a; //Khai báo biến a có kiểu int
```

```
float b=0; //khai báo biến b có kiểu float, b được khởi tạo giá trị 0
```

```
int x=5, y=7; //khai báo 2 biến x và y có kiểu int,
```



# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Hằng (Constant)

- Một hằng là một vùng nhớ được đặt tên, dùng để lưu trữ **giá trị không đổi** trong chương trình.
- Trong C, **phải khai báo hằng trước khi sử dụng**.
- Khai báo hằng:

```
constant <kiểu dữ liệu> <tên hằng> = <giá trị>;  
#define <tên hằng> <giá trị>
```

- Ví dụ:

```
constant PI = 3.1415;  
#define MAX 100
```

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Biểu thức (Expression)

- Biểu thức là sự kết hợp của các toán tử (operators) và các toán hạng (operands).
- Các toán tử tác động lên các toán hạng theo **thứ tự ưu tiên** xác định để tính giá trị của biểu thức.
- Toán tử: **+**, **-**, **\***, **/**, **%**....
- Toán hạng: **hằng**, **biến**, **lời gọi hàm**...
- Ví dụ:

$2 + 3$

$a / 5$

$(a + b) * 5$

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu

cơ sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Toán tử gán (Assignment operator)

- Toán tử gán thường được sử dụng trong lập trình, dùng để gán giá trị cho biến.

- Cú pháp:

<tên biến> = <giá trị>;

<tên biến> = <tên biến>;

<tên biến> = <biểu thức>;

- Ví dụ:

```
int main()
{
    int a = 5, b;
    b = a;
    a = a + 3;
    b = a * 2 - 10;
}
```

- **Lưu ý:** Kiểu của biểu thức/biến/giá trị ở vế phải của toán tử gán phải tương thích với kiểu của biến. Nếu phép gán không tương thích thì chương trình sẽ phát sinh một lỗi.

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Toán tử số học (Arithmetic operator)

- Toán tử số học được dùng để viết các công thức toán học.
- Các phép toán số học (Giả sử biến  $a$  có giá trị 5 và biến  $b$  có giá trị 10):

| Toán tử | Ý nghĩa                                | Ví dụ            | Ghi chú   |
|---------|--|------------------|---|
| +       | Cộng hai toán hạng                     | $a = 3 + 5;$     | Áp dụng trên tất cả các toán hạng có kiểu dữ liệu <code>char</code> , <code>int</code> , <code>float</code> , <code>double</code> (kể cả <code>long</code> , <code>short</code> , <code>unsigned</code> ) |
| -       | Trừ toán hạng thứ 2 từ toán hạng thứ 1 | $b = 3 - 5;$     |   |
| *       | Nhân hai toán hạng                     | $c = 3 * 5;$     |   |
| /       | Chia lấy phần nguyên                   | $d = 3 / 5;$     |   |
| %       | Chia lấy phần dư                       | $e = 3 \% 5;$    | Áp dụng trên các toán hạng có kiểu dữ liệu <code>char</code> , <code>int</code> , <code>long</code>   |
| -       | Đảo dấu                                | $a = 3; a = -a;$ |   |

- Lưu ý:
  - Nếu hai toán hạng là số nguyên sẽ cho kết quả là số nguyên. (Ví dụ:  $9 / 6 = 1$ )
  - Nếu một trong hai, hoặc cả hai toán hạng là số thực thì sẽ cho kết quả là số thực (Ví dụ:  $9.0 / 6 = 9 / 6.0 = 9.0 / 6.0 = 1.5$ )
  - $f = 1 * 1.0 / 2;$     $g = \text{float}(1) / 2;$     $h = \text{float}(1 / 2);$

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Toán tử số học (Arithmetic operator)

- Toán tử gộp: **+=, -=, \*=, /=, %=**

| Biểu thức đầy đủ | Biểu thức rút gọn | Ví dụ                               |
|------------------|-------------------|-------------------------------------|
| $a = a + b;$     | $a += b;$         | $a = 3;$<br>$a += 2;$               |
| $a = a - b;$     | $a -= b;$         | $a = 3;$<br>$b = 5;$<br>$a += b;$   |
| $a = a * b;$     | $a *= b;$         | $a = 3;$<br>$b = 5;$<br>$a *= b;$   |
| $a = a / b;$     | $a /= b;$         | $a = 3;$<br>$b = 5;$<br>$a /= b;$   |
| $a = a \% b;$    | $a \% = b;$       | $a = 3;$<br>$b = 5;$<br>$a \% = b;$ |

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Toán tử tăng giảm một đơn vị (Increment/Decrement operator)

- Ngôn ngữ C đưa ra các phép toán một ngôi để **tăng hoặc giảm các biến** (kiểu số nguyên hoặc số thực) **một giá trị**.
- Các phép toán tăng giảm có trong C (giả sử *i* là biến):

| Toán tử    | Biểu diễn  | Ý nghĩa   |
|------------|------------|---|
| Tăng trước | <b>++i</b> | Giá trị của biến <i>i</i> được cộng thêm 1 trước, sau đó giá trị này mới được thực hiện trong biểu thức.            |
| Tăng sau   | <b>i++</b> | Giá trị của biến <i>i</i> được đưa vào thực hiện biểu thức trước, sau đó giá trị của <i>i</i> mới được cộng thêm 1. |
| Giảm trước | <b>--i</b> | Giá trị của biến <i>i</i> được trừ đi 1 trước, sau đó giá trị này mới được thực hiện trong biểu thức.               |
| Giảm sau   | <b>i--</b> | Giá trị của biến <i>i</i> được đưa vào thực hiện biểu thức trước, sau đó giá trị của <i>i</i> mới được trừ đi 1.    |

- Ví dụ:

```
int i = 5, x;
```

```
x = i++;
```

```
// x có giá trị là 5
```

```
// i có giá trị là 6
```

```
int i = 5, x;
```

```
x = ++i;
```

```
// x có giá trị là 6
```

```
// i có giá trị là 6
```

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Toán tử quan hệ (Relational operator)

- Toán tử quan hệ được dùng để **so sánh** hai biểu thức với nhau.
- Biểu thức dùng toán tử quan hệ trả về **0** hoặc **false** nếu sai và **1** hoặc **true** nếu đúng.
- Các phép toán quan hệ: **==**, **>**, **<**, **>=**, **<=**, **!=**
- Ví dụ:

```
s1 = (3 == 5);  
s2 = (3 != 5);  
s3 = (3 > 5);  
s4 = (3 >= 5);  
s5 = (3 < 5);  
s6 = (3 <= 5);
```

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Toán tử logic (Logical operator)

- Biểu thức chứa toán tử logic trả về **0 (false)** hoặc **1 (true)** tùy thuộc vào kết quả biểu thức là đúng hay sai.
- Các toán tử logic thường được sử dụng trong việc ra quyết định làm một điều gì đó trong lập trình.
- Giả sử ta có **A (operator) B**, trong đó **operator** chính là một trong ba toán tử sau:

| Toán tử | Ý nghĩa  | Ví dụ                    |
|---------|--|--------------------------|
| &&      | <b>Toán tử AND</b> , trả về true khi A và B đều true, chỉ cần một trong hai vế sai là trả về false | s1 = (1 > 2) && (3 > 4); |
|         | <b>Toán tử OR</b> , trả về true khi một trong hai vế đúng, trả về false khi cả hai vế sai.         | s2 = (1 > 2)    (3 > 4); |
| !       | <b>Toán tử NOT</b> , trả về true nếu biểu thức false, trả về false nếu biểu thức true              | s3 = !(1 > 2);           |

### • Lưu ý:

- Giả sử một chương trình phải kiểm tra điều kiện  $-5 \leq x \leq 5$ . Khi đó điều kiện được viết thành  $(x \geq -5) \&\& (x \leq 5)$ .
- Giả sử một chương trình phải kiểm tra đồng thời hai điều kiện  $a < 10$  và  $b == 5$ . Khi đó điều kiện được viết thành  $(a < 10) \&\& (b == 5)$ .
- Giả sử một chương trình phải kiểm tra một trong hai điều kiện  $a < 10$  và  $b == 5$ . Khi đó điều kiện được viết thành  $(a < 10) || (b == 5)$



# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Toán tử trên bit (Bitwise operator)

- Toán tử trên bit xem xét các toán hạng dưới dạng chuỗi bit.
- Các toán tử trên bit:

| Bit a | Bit b | $\sim a$ | $a \& b$ | $a   b$ | $a \wedge b$ |
|-------|-------|----------|----------|---------|--------------|
| 0     | 0     | 1        | 0        | 0       | 0            |
| 0     | 1     | 1        | 0        | 1       | 1            |
| 1     | 0     | 0        | 0        | 1       | 1            |
| 1     | 1     | 0        | 1        | 1       | 0            |

- Ví dụ:

○  $13 \& 10$

○  $13 | 10$

○  $13 \wedge 10$

$a = 13 \rightarrow$  đổi ra hệ nhị phân là 1101

$b = 10 \rightarrow$  đổi ra hệ nhị phân là 1010

1101

$\& \underline{1010}$

= 1000

= 8

1101

$| \underline{1010}$

= 1111

= 15

1101

$\wedge \underline{1010}$

= 0111

= 7

(dạng thập phân)

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Toán tử điều kiện (Conditional operator)

- Toán tử điều kiện yêu cầu 3 toán hạng, có dạng tổng quát như sau:

<biểu thức 1> ? <biểu thức 2> : <biểu thức 3>

- Nếu <biểu thức 1> **đúng** thì giá trị là <biểu thức 2>.
- Nếu <biểu thức 1> **sai** thì giá trị là <biểu thức 3>.
- Ví dụ:

```
int main()
{
    int m = 1, n = 2;
    int min = (m < n? m : n); // min nhận giá trị 1
}
```

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ▪ Độ ưu tiên của các toán tử (Operator precedence)

| Toán tử                       | Độ ưu tiên |
|-------------------------------|------------|
| () [] -> .                    | →          |
| ! ++ -- - + * (cast) & sizeof | ←          |
| * / %                         | →          |
| + -                           | →          |
| << >>                         | →          |
| < <= > >=                     | →          |
| == !=                         | →          |
| &                             | →          |
|                               | →          |
| ^                             | →          |
| &&                            | →          |
|                               | →          |
| ? :                           | ←          |
| = += -= *= /= %= &= ...       | ←          |
| ,                             | ←          |

### • Quy tắc thực hiện

- Thực hiện biểu thức trong ( ) sâu nhất trước.
  - Thực hiện theo thứ tự ưu tiên các toán tử.
- => Tự chủ động thêm ( )

### • Ví dụ: $n = 2 + 3 * 5;$

=>  $n = 2 + (3 * 5);$

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Giới thiệu

- Câu lệnh (Statement) là một chỉ thị trực tiếp, hoàn chỉnh nhằm ra lệnh cho máy tính thực hiện một số tác vụ nhất định nào đó.
- Phân loại
  - Câu lệnh đơn: chỉ gồm một câu lệnh.
  - Câu lệnh phức (khối lệnh): gồm nhiều câu lệnh đơn được bao bởi { }
- Trình biên dịch bỏ qua các khoảng trắng (hay tab hoặc xuống dòng) chen giữa lệnh.

○ Ví dụ:

```
a=2912 ;
```

```
a = 2912 ;
```

```
a  
=  
2912 ;
```

- Một chương trình máy tính bao gồm nhiều câu lệnh, một trong các lệnh cơ bản nhất của chương trình chính là **các lệnh nhập xuất**.

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Câu lệnh xuất

- Thư viện

`#include <stdio.h>` (standard input/output)

- Cú pháp

`printf(<chuỗi định dạng>[, <Trường ra 1>, <Trường ra 2>, ...]);`

- **<chuỗi định dạng>**: Là cách trình bày thông tin xuất và được đặt trong cặp nháy kép “ ”. Các dạng <chuỗi định dạng>:
  - Văn bản thường (literal text)
  - Ký tự điều khiển (escape sequence)
  - Đặc tả (conversion specifier)
- **<Trường ra 1>, <Trường ra 2>, ...**: Là các mục dữ kiện cần in ra màn hình. Các trường này có thể là **biến**, **hằng** hoặc **biểu thức** và chúng phải được xác định giá trị trước khi in ra.

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Câu lệnh xuất

- Chuỗi định dạng là dạng văn bản thường

- Nội dung được xuất giống như nội dung được gõ trong chuỗi định dạng.
- Ví dụ
  - Xuất chuỗi **Hello World**
    - `printf("Hello "); printf("World");`
    - `printf("Hello World");`
  - Xuất chuỗi **a + b**
    - `printf("a + b");`

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học



## ■ Câu lệnh xuất

- Chuỗi định dạng là dạng ký tự điều khiển
  - Gồm dấu \ và một ký tự đặc biệt.

| Ký tự | Ý nghĩa                       |
|-------|-------------------------------|
| \n    | Sang dòng mới                 |
| \t    | Dấu tab                       |
| \b    | Lùi lại một vị trí            |
| \f    | Sang trang mới                |
| \\    | In ra ký tự \                 |
| \"    | In ra ký tự "                 |
| \'    | In ra ký tự '                 |
| \?    | In ra ký tự ?                 |
| \a    | Phát ra tiếng chuông hệ thống |

- Ví dụ

```
printf("\n");  
printf("\t");  
printf("\n\t");
```

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Câu lệnh xuất

### • Chuỗi định dạng là dạng đặc tả:

- Gồm dấu % và một ký tự, dùng để xác định kiểu của biến hoặc giá trị muốn xuất.
- Các trường ra chính là các biến/giá trị muốn xuất, được liệt kê theo thứ tự cách nhau dấu phẩy

| Kiểu dữ liệu      | Định dạng |
|-------------------|-----------|
| int               | %d        |
| char              | %c        |
| float             | %f        |
| double            | %lf       |
| unsigned int      | %u        |
| long int          | %li       |
| unsigned long int | %lu       |
| unsigned char     | %c        |
| long double       | %Lf       |

○ Ví dụ:

```
int a = 10, b = 20;
```

```
printf("%d", a);
```

→ Xuất ra 10

```
printf("%d", b);
```

→ Xuất ra 20

```
printf("%d%d", a, b);
```

→ Xuất ra 1020

```
float x = 15.06;
```

```
printf("%f", x);
```

→ Xuất ra 15.060000

```
printf("%f", 1.0/3);
```

→ Xuất ra 0.333333



# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo

## ■ Câu lệnh xuất

- Chuỗi định dạng là dạng đặc tả

- Một số định dạng khác đối với số nguyên, số thực và ký tự:

| Định dạng          | Ý nghĩa   |
|--------------------|---|
| <code>%nd</code>   | Xuất số nguyên có bề rộng <code>n</code> , và số được canh phải                   |
| <code>%n.kf</code> | Xuất số thực có bề rộng <code>n</code> trong đó có <code>k</code> số sau dấu chấm |
| <code>%o</code>    | Xuất số nguyên hệ bát phân không dấu  |
| <code>%x</code>    | Xuất số nguyên hệ thập lục phân không dấu   |
| <code>%s</code>    | Xuất một chuỗi ký tự  |
| <code>%e</code>    | Xuất số nguyên dạng khoa học (nhân 10 mũ x)                                       |

- Ví dụ:

```
int a = 1706;
float x = 176.85;
printf("%10d", a);printf("\n");
printf("%10.2f", x);printf("\n");
printf("%.2f", x);printf("\n");
```

|   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
|   |   |   |   |   |   | 1 | 7 | 0 | 6 |  |  |  |  |  |  |
|   |   |   |   | 1 | 7 | 6 | . | 8 | 5 |  |  |  |  |  |  |
| 1 | 7 | 6 | . | 8 | 5 |   |   |   |   |  |  |  |  |  |  |

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo

## ■ Câu lệnh nhập

- Thư viện

#include <stdio.h> (standard input/output)

- Cú pháp

**scanf**(<chuỗi định dạng>, <Địa chỉ của các biến>);

- <chuỗi định dạng>: giống định dạng xuất nhưng chỉ có các đặc tả.
- <Địa chỉ của các biến>: là địa chỉ của các biến mà chúng ta cần nhập giá trị cho nó, được viết ở dạng: **&Tên\_biến**

| Kiểu dữ<br>liệu | Định<br>dạng | Ví dụ                     | Giải thích                             |
|-----------------|--------------|---------------------------|--|
| int             | %d           | <b>scanf</b> ("%d", &a);  | Nhập dữ liệu cho biến a có kiểu int    |
| long            | %ld          | <b>scanf</b> ("%ld", &b); | Nhập dữ liệu cho biến b có kiểu long   |
| float           | %f           | <b>scanf</b> ("%f", &c);  | Nhập dữ liệu cho biến c có kiểu float  |
| double          | %lf          | <b>scanf</b> ("%lf", &d); | Nhập dữ liệu cho biến d có kiểu double |
| char            | %c           | <b>scanf</b> ("%c", &e);  | Nhập dữ liệu cho biến e có kiểu char   |

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Câu lệnh nhập

- Một số câu lệnh nhập sai (Giả sử a và b là biến kiểu **int**)
  - `scanf("%d", a);` // Thiếu dấu **&**
  - `scanf("%d", &a, &b);` // Thiếu **%d** cho biến b
  - `scanf("%f", &a);` // a là biến kiểu số nguyên
  - `scanf("%9d", &a);` // không được định dạng
  - `scanf("a = %d, b = %d", &a, &b);` // Chuỗi định dạng sai

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

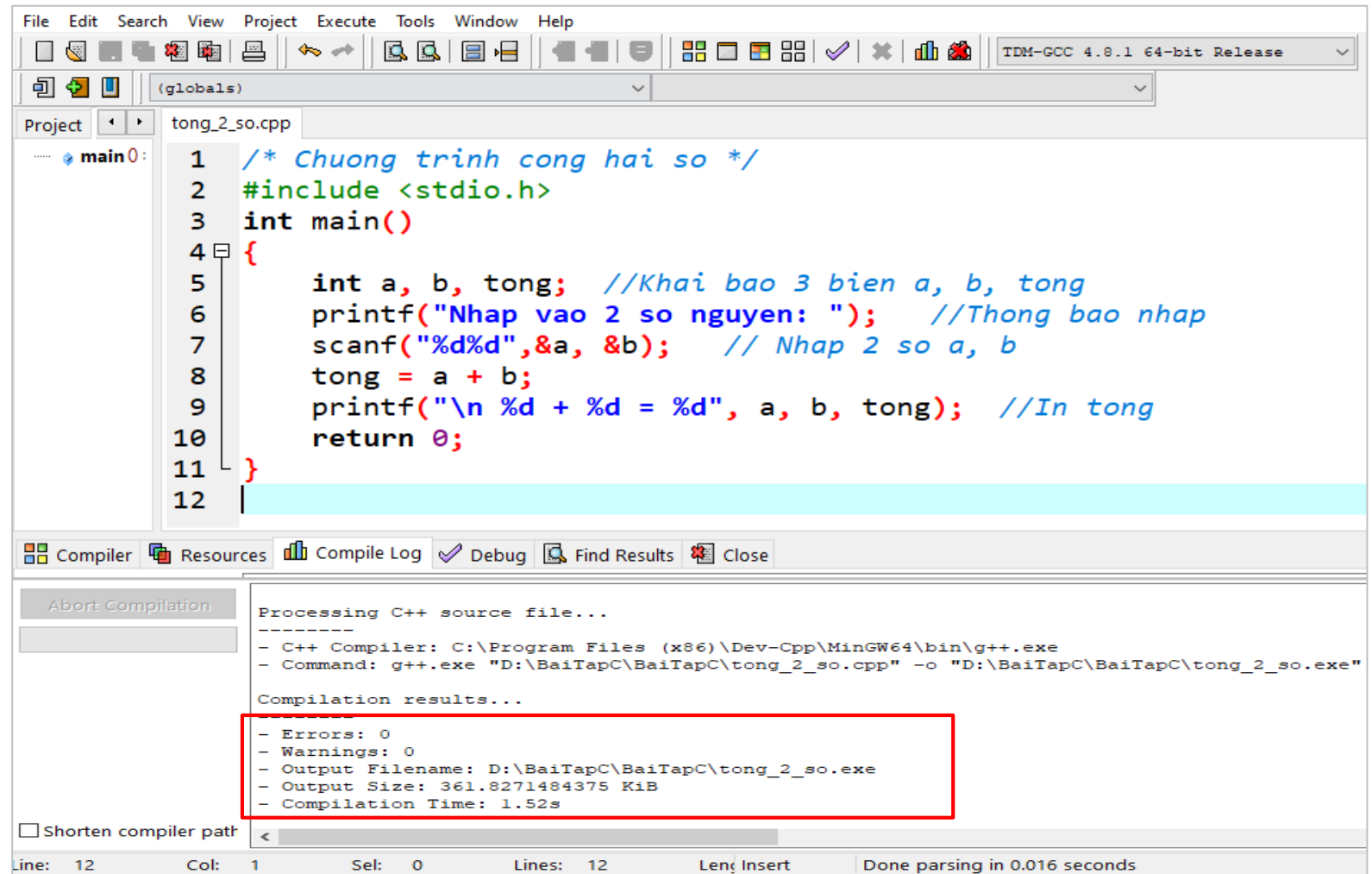
Câu lệnh

Một số hàm toán học

Tài liệu tham khảo

## ■ Chương trình minh họa

- Chương trình nhập vào 2 số nguyên a và b từ bàn phím, sau đó tính tổng của 2 số.



```

File Edit Search View Project Execute Tools Window Help
(globals)
Project: tong_2_so.cpp
main0:
1  /* Chương trình cộng hai số */
2  #include <stdio.h>
3  int main()
4  {
5      int a, b, tong; // Khai báo 3 biến a, b, tong
6      printf("Nhập vào 2 số nguyên: "); // Thông báo nhập
7      scanf("%d%d", &a, &b); // Nhập 2 số a, b
8      tong = a + b;
9      printf("\n %d + %d = %d", a, b, tong); // In tong
10     return 0;
11 }
12

Compiler Resources Compile Log Debug Find Results Close
Processing C++ source file...
- C++ Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\g++.exe
- Command: g++.exe "D:\BaiTapC\BaiTapC\tong_2_so.cpp" -o "D:\BaiTapC\BaiTapC\tong_2_so.exe"
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: D:\BaiTapC\BaiTapC\tong_2_so.exe
- Output Size: 361.8271484375 KiB
- Compilation Time: 1.52s
Shorten compiler path
Line: 12 Col: 1 Sel: 0 Lines: 12 Len: Insert Done parsing in 0.016 seconds
  
```

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo

## ■ Các hàm trong thư viện toán học

- Thư viện

```
#include <math.h>
```

```
#include <stdlib.h>
```

| Hàm     | Mô tả  | Ví dụ  |
|---------|--|--|
| sqrt()  | Tìm căn bậc hai của một số<br>Cú pháp: <i>sqrt(number)</i>   | <code>float x = sqrt(225);</code><br><code>// x = 15.000000</code>   |
| pow()   | Tính lũy thừa của một số<br>Cú pháp: <i>pow(m, n)</i> trong đó m là cơ số, n là số mũ  | <code>float x = pow(8, 3);</code><br><code>// x = 512.000000</code>  |
| abs()   | Lấy giá trị tuyệt đối của một số nguyên<br>Cú pháp: <i>abs(number)</i>   | <code>int a = abs(-341);</code><br><code>// a = 341</code>   |
| fabs()  | Lấy giá trị tuyệt đối của một số thực<br>Cú pháp: <i>fabs(number)</i>  | <code>float y = fabs(35.47);</code><br><code>// y = 35.47</code>   |
| round() | Trả về giá trị nguyên gần nhất của các kiểu <i>float/ double/ long double</i> . Nếu giá trị thập phân $< 0.5$ , nó trả về giá trị nguyên nhỏ hơn số đó. Nếu giá trị thập phân $\geq 0.5$ , nó trả về giá trị nguyên lớn hơn số đó một đơn vị | <code>int a = round(43.56);</code><br><code>int b = round(56.18);</code><br><code>int c = round(-43.56);</code><br><code>// a = 44, b = 56, c = -44</code> |
| floor() | Trả về giá trị nguyên gần nhất nhỏ hơn hoặc bằng số đã cho<br>Cú pháp: <i>floor(number)</i>  | <code>int x = floor(43.99);</code><br><code>// x = 43</code>   |
| ceil()  | Trả về giá trị nguyên gần nhất lớn hơn hoặc bằng số đã cho<br>Cú pháp: <i>ceil(number)</i>   | <code>int x = ceil(43.99);</code><br><code>// x = 44</code>  |
| trunc() | Loại bỏ giá trị thập phân từ giá trị dấu phẩy động và trả về giá trị nguyên<br>Cú pháp: <i>trunc(number)</i>   | <code>int x = trunc(43.99);</code><br><code>// x = 43</code>   |

# TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo



## ■ Các hàm trong thư viện toán học

- Thư viện

```
#include <math.h>
```

```
#include <stdlib.h>
```

|         |  |  |
|---------|--|--|
| exp()   | Tính lũy thừa $e^x$ của một số x<br>Cú pháp: <i>exp(number)</i>                  | float x = exp(0) ;<br>// x = 1.000000  |
| log()   | Tính logarit cơ số e của một số<br>Cú pháp: <i>log(number)</i>                   | double x = log(2.7) ;<br>// x = 0.993252   |
| log10() | Tính logarit cơ số 10 của một số<br>Cú pháp : <i>log10(number)</i>               | double x = log10(10000) ;<br>// x = 4.000000   |
| atoi()  | Chuyển một chuỗi thành một số nguyên<br>Cú pháp: <i>atoi(string)</i>             | int i = atoi("1234") ;<br>// i = 1234  |
| atof()  | Chuyển một chuỗi thành một số kiểu <i>double</i><br>Cú pháp: <i>atof(string)</i> | float f = atof("123.45") ;<br>// f = 123.45  |
| cos()   | Trả về cosin của một góc tính bằng radian<br>Cú pháp: <i>cos(number)</i>         | float x = 60.0;<br>const float PI = 3.14159;<br>float y = cos(x*(PI/180.0)) ;<br>// y = 0.500000 |
| acos()  | Trả về arc cosin của một góc tính bằng radian<br>Cú pháp: <i>cos(number)</i>     | float x = 0.9;<br>float y = acos(x*(180.0/PI)) ;<br>// y = 25.855040                             |

## TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH

Các kiểu dữ liệu cơ  
sở

Biến

Hằng

Biểu thức

Phép toán

Câu lệnh

Một số hàm toán học

Tài liệu tham khảo

- Phạm Văn Ất, Đỗ Văn Tuấn, **Giáo trình Kỹ thuật lập trình C : Căn bản và nâng cao**, NXB Thông tin và truyền thông, 2016
- David Griffiths & Dawn Griffiths, **Head First C**, O'Reilly, 2012
- Lê Thị Bích Hằng, **Tài liệu tham khảo Nhập môn lập trình**, ĐH Nha Trang
- <https://www.tutorialspoint.com/cprogramming/>