



Phần trình bày của:

**ĐẬU HẢI PHONG**

*Giảng viên*

*Đại Nam, ngày 01 tháng 19 năm 2023*

# LƯU Ý

**KHÔNG NÓI  
CHUYỆN RIÊNG**



**KHÔNG SỬ DỤNG  
ĐIỆN THOẠI**



**KHÔNG NGỦ GẬT**



**GHI CHÉP ĐẦY ĐỦ**



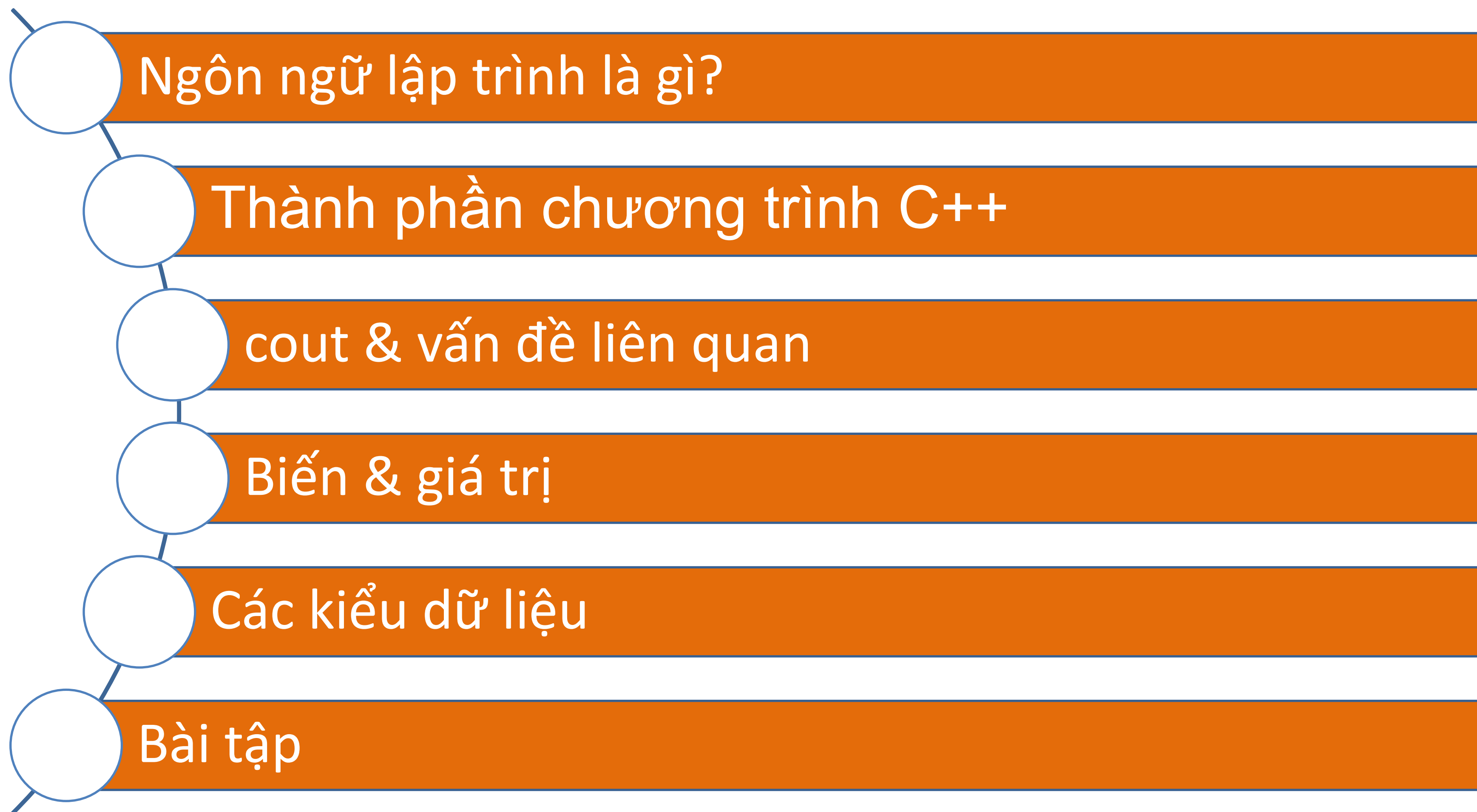


# LẬP TRÌNH CƠ BẢN

# CHƯƠNG 1

## GIỚI THIỆU VỀ NGÔN NGỮ LẬP TRÌNH





# Ngôn ngữ lập trình là gì?

- Ngôn ngữ lập trình (NNLT) là gì?
  - Là tập hợp các quy tắc cung cấp cho máy tính biết các thao tác cần thực hiện.
  - Để diễn đạt các **thuật toán** cho máy tính thực hiện
  - Thành phần và đặc tính của NNLT:
    - Cú pháp, Kiểu dữ liệu, Biến, Toán tử, Cấu trúc điều khiển, Thư viện, Mô hình (cấu trúc, hướng đối tượng)
  - ...

# Ngôn ngữ lập trình là gì?

- Các mức độ của ngôn ngữ lập trình?

- Ngôn ngữ bậc cao:

```
class Triangle {  
    ...  
    float surface()  
        return b*h/2;  
}
```

- Ngôn ngữ bậc thấp:

```
LOAD r1,b  
LOAD r2,h  
MUL r1,r2  
DIV r1,#2  
RET
```

- Mã máy tính thực hiện:

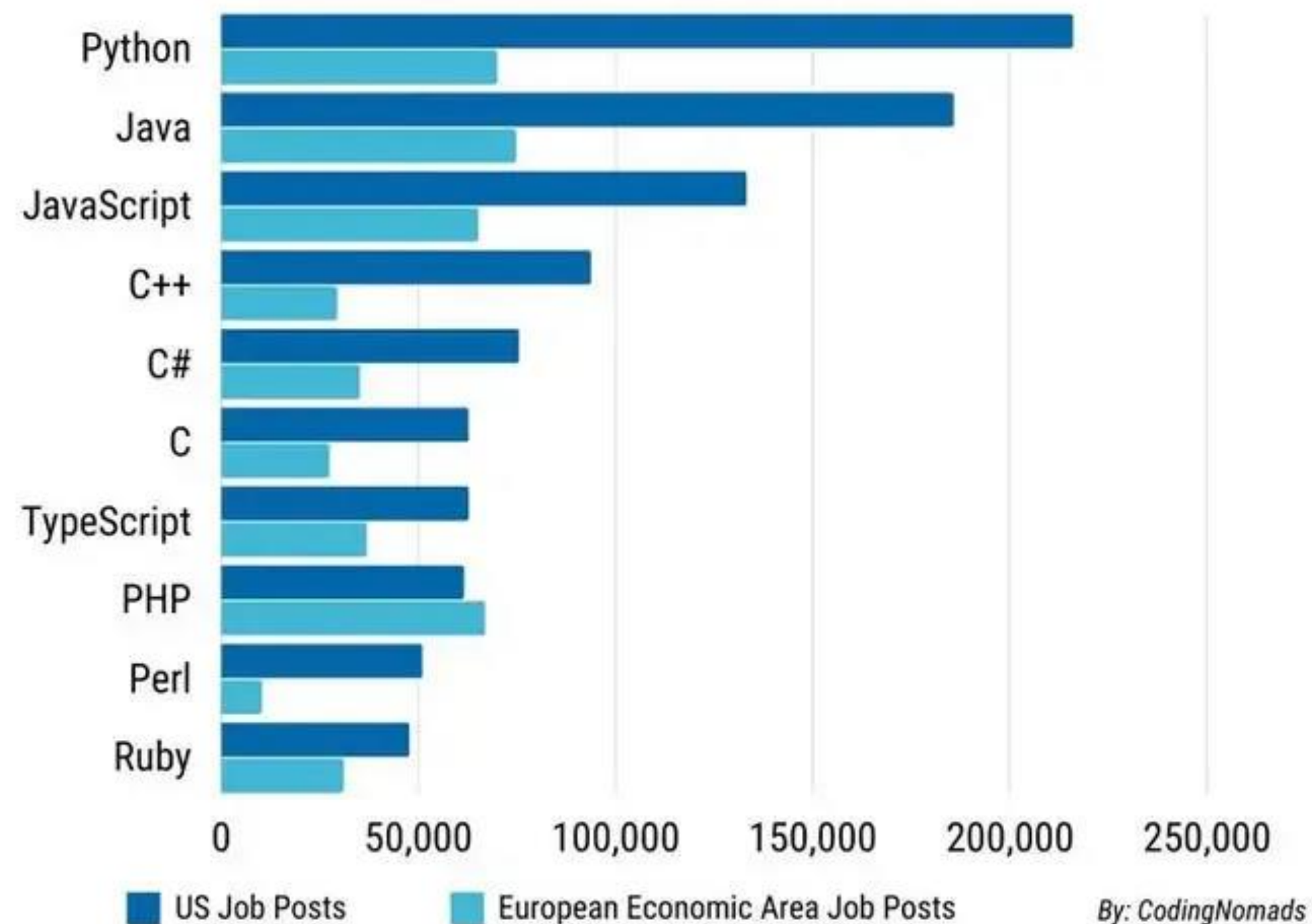
```
0001001001000101  
0010010011101100  
10101101001...
```

# Ngôn ngữ lập trình là gì?

- Có những loại ngôn ngữ lập trình nào?

## Most in-demand programming languages of 2022

*Based on LinkedIn job postings in the USA & Europe*





# Các thành phần của chương trình C++

```
// sample C++ program ← Lời chú thích
#include <iostream> ← Chỉ thị tiền xử lý
using namespace std; ← Sử dụng thư viện std
int main() ← Bắt đầu hàm main
{
    ← Bắt đầu khối lệnh của hàm main
    cout << "Hello, there!"; ← Câu lệnh đưa ra màn hình
    return 0; ← Gửi 0 tới hệ điều hành
} ← Kết thúc khối lệnh của hàm main
```

Chuỗi ký tự

# Một số kí tự đặc biệt

Ký tự	Ý nghĩa
//	Bắt đầu dòng ghi chú
#	Bắt đầu chỉ thị tiền xử lý
< >	Bao tên tệp trong #include
( )	Sử dụng khi đặt tên 1 hàm
{ }	Bao nhóm các câu lệnh
" "	Bao một chuỗi ký tự
;	Kết thúc câu lệnh

# *cout* & vấn đề liên quan

- ***cout***: hiện thị thông tin trên màn hình máy tính
- **<<**: toán tử gửi thông tin ra màn hình
  - VD1: `cout << "Programming is fun!";`
  - VD2:
    - `cout << "Hello " << "there!";`Tương đương:
    - `cout << "Hello ";`
    - `cout << "there!";`
- **`endl`**: kết thúc 1 dòng
  - VD3: `cout << "Programming is" << endl << "fun!";`
- **`\n`**: bắt đầu 1 dòng mới
  - VD4: `cout << "Programming \n" << "fun!";`

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Programming is fun!";
    return 0;
}
```

```
Console  Shell
❯ clang++-7 -pthread -std=c++17 -o main main.cpp
❯ ./main
Programming is fun!❯
```

```
Programming is
fun!❯
```

# Chỉ thị `#include`

- Chèn nội dung 1 tệp khác vào trong chương trình
- Xuất hiện ở đầu mọi chương trình
- VD:
  - `#include <iostream>`
- Không có dấu chấm phẩy “;” ở cuối

`#include`



# Biến và giá trị

- Biến (variable): thể hiện vị trí lưu trữ trên bộ nhớ máy tính
  - Biến gồm: tên biến và kiểu giá trị có thể lưu trữ
  - Phải được định nghĩa trước khi sử dụng
- Giá trị (literal): là các giá trị không đổi được gán cho biến

```
1 // This program has a variable.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int number;
8
9     number = 5;
10    cout << "The value of number is " << "number" << endl;
11    cout << "The value of number is " << number << endl;
12
13    number = 7;
14    cout << "Now the value of number is " << number << endl;
15
16    return 0;
17 }
```

**Biến number**

**5 - giá trị số nguyên**

**giá trị chuỗi**

# Tên xác định - Identifier

- Là tên người dùng định nghĩa trong chương trình
  - VD: Tên biến, tên hàm, hằng số,...

**Table 2-4 The C++ Key Words**

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

# Tên biến

- Tên biến nên thể hiện nội dung của biến
  - VD:
    - **iTongDonhang** //tên biến để lưu trữ tổng số đơn hàng kiểu nguyên
    - **fDTB** // tên biến để lưu trữ điểm trung bình kiểu thực
- **Chú ý:**
  - Tên biến bắt đầu là 1 kí tự theo bảng chữ cái hoặc kí tự “\_”
  - Sau kí tự đầu thì có thể các chữ cái hoặc số
  - Tên biến **phân biệt** chữ HOA và chữ thường

Đặt tên	Đúng/Sai	Lý do
totalSales	Đúng	
total_Sales	Đúng	
total.Sales	Sai	Bao gồm dấu chấm “.”
4thQtrSales	Sai	Bắt đầu là 1 chữ số
totalSale\$	Sai	Bao gồm ký tự “\$”

# Các kiểu dữ liệu (1)

- Kiểu nguyên – integer:
  - Biến kiểu nguyên có thể lưu trữ các số như: 12, 9 hay -99

**Table 2-6 Integer Data Types, Sizes, and Ranges**

Data Type	Size	Range
short	2 bytes	-32,768 to +32,767
unsigned short	2 bytes	0 to +65,535
int	4 bytes	-2,147,483,648 to +2,147,483,647
unsigned int	4 bytes	0 to 4,294,967,295
long	4 bytes	-2,147,483,648 to +2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295

- VD:
  - `int length, width; // Khai báo 2 biến length, width kiểu số nguyên`
  - `int width = 20; // Khai báo biến width kiểu nguyên và khởi tạo giá trị là 20`
  - `unsigned int area; // Khai báo biến area kiểu số nguyên không dấu`



# Các kiểu dữ liệu (2)

- **Kiểu kí tự - char:**

- Biến kí tự có thể lưu trữ các kí tự hoặc giá trị nguyên nhỏ
- Thường sử dụng 1 byte trong bộ nhớ
- Giá trị số của ký tự được lưu trong bộ nhớ

**CODE:**

```
char letter;  
letter = 'C';
```

**MEMORY:**

```
letter  
67
```

- Giá trị kí tự phải được bao bọc bởi nháy đơn như 'A', '1', 'c'

# Các kiểu dữ liệu (3)

- **Chuỗi kí tự:**

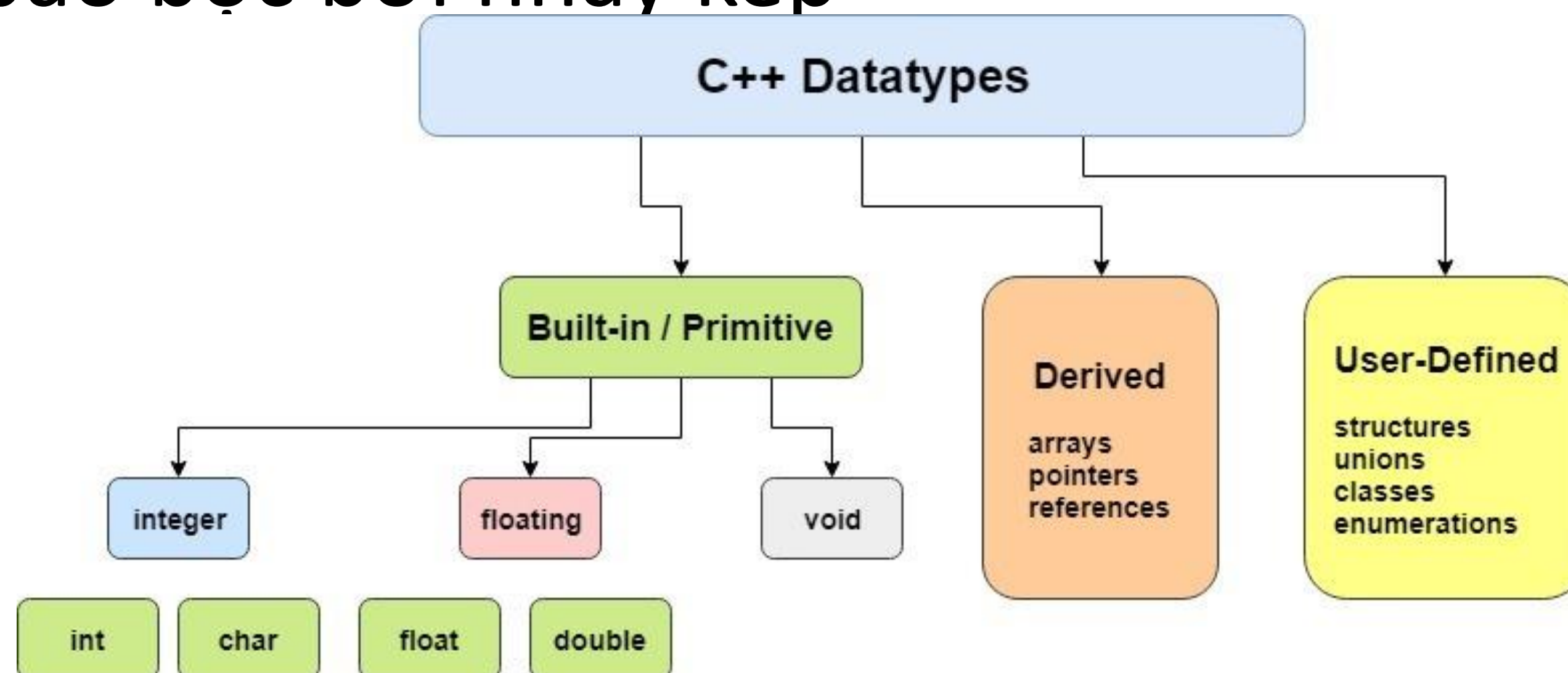
- Gồm 1 dãy kí tự trong vị trí liên tiếp

- “Hello”, “programming”,...

- Lưu trữ kí tự kết thúc \0 ở cuối

H	e	l	l	o	\0
---	---	---	---	---	----

- Giá trị được bao bọc bởi nháy kép “”



# Các kiểu dữ liệu (4)

- Kiểu dữ liệu bool:
  - Biểu diễn giá trị **true** hoặc **false**
  - Có thể lưu trữ giá trị nguyên nhỏ: 0 ~ false và 1 ~ true
    - `bool allDone = true;`
    - `bool finished = false;`

`allDone`   `finished`

1

0

## Program 2-17

```
1 // This program demonstrates boolean variables.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     bool boolValue;
8
9     boolValue = true;
10    cout << boolValue << endl;
11    boolValue = false;
12    cout << boolValue << endl;
13    return 0;
14 }
```

## Program Output

1  
0

# Lớp string

- Kiểu dữ liệu đặc biệt làm việc với chuỗi
- Khi sử dụng phải có chỉ thị `#include <string>`
  - VD:
    - Khai báo biến kiểu string
      - `string firstName, lastName;`
    - Gán giá trị "George" cho biến `firstName`
      - `firstName = "George";`
    - Gán giá trị "Washington" cho biến `lastName`
      - `lastName = "Washington";`
    - In giá trị 2 biến `firstName, lastName` ra màn hình
      - `cout << firstName << " " << lastName;`

```
1 // This program demonstrates the string class.
2 #include <iostream>
3 #include <string>           // Required for the string class
4 using namespace std;
5
6 int main()
7 {
8     string movieTitle;
9
10    movieTitle = "Wheels of Fury";
11    cout << "My favorite movie is " << movieTitle << endl;
12    return 0;
13 }
14
```



# Kiểu dữ liệu số thực

- Gồm các kiểu: **Table 2-8 Floating Point Data Types on PCs**

Data Type	Key Word	Description
Single precision	float	4 bytes. Numbers between $\pm 3.4\text{E-}38$ and $\pm 3.4\text{E}38$
Double precision	double	8 bytes. Numbers between $\pm 1.7\text{E-}308$ and $\pm 1.7\text{E}308$
Long double precision	long double*	8 bytes. Numbers between $\pm 1.7\text{E-}308$ and $\pm 1.7\text{E}308$

- float
- double
- long double
- Có thể biểu diễn dưới dạng:
  - Dấu chấm cố định:
    - 31.4159
    - 0.0000625
  - Ký pháp khoa học:
    - 3.14159E1
    - 6.25e-5

# Gán và khởi tạo giá trị cho biến

- Câu lệnh gán sử dụng dấu “=”
  - `Item = 12; // Gán giá trị 12 cho biến item`
  - `12 = item; // Sai phép gán`

- Khởi tạo giá trị cho biến:
  - `int length = 12;`
  - `int length = 12, width = 5,`

- Phạm vi của biến:
  - Là phần chương trình mà biến có thể truy cập
  - Biến không thể sử dụng trước khi định nghĩa

## Program 2-19

```
1 // This program shows variable initialization.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int month = 2, days = 28;
8
9     cout << "Month " << month << " has " << days << " days.\n";
10    return 0;
11 }
```

## Program Output

Month 2 has 28 days.

# Toán tử số học

- Gồm toán tử: đơn phân (unary), nhị phân (binary) và tam phân (ternary).
  - unary (1 toán hạng)
    - VD: -5
  - binary (2 toán hạng)
    - VD: 13 – 7
    - Cộng (+), trừ (-), nhân (\*), chia (/), chia lấy dư (%)
    - Chia 1 số nguyên cho 1 số nguyên, kết quả là 1 số nguyên
      - `cout << 13 / 5; // hiển thị 2`
    - Chia 1 số nguyên cho 1 số thực, kết quả là 1 số thực
      - `cout << 13 / 5.0; // hiển thị 2.6`
      - `cout << 91.0 / 7; // hiển thị 13.0`
    - Chia lấy phần dư:
      - `cout << 13 % 5; //hiển thị 3`
  - ternary (3 toán hạng)
    - `exp1 ? exp2 : exp3`

# Ghi chú

- Sử dụng như 1 phần của chương trình
- Nhằm giải thích cho người đọc mã nguồn của chương trình:
  - Mục đích chương trình
  - Miêu tả sử dụng biến
  - Giải thích 1 đoạn mã phức tạp
- Bỏ qua khi biên dịch
- Bắt đầu bằng:
  - `//` cho lời giải thích trên 1 dòng
  - `/*` cho lời giải thích trên nhiều dòng, kết thúc bằng `*/`
- VD:
  - `int width = 15; // width in inches`
  - `int area; /* calculated area */`



# Tên hằng số

- Tên hằng (biến hằng) là biến không thể thay đổi trong suốt quá trình thực hiện chương trình.
- Sử dụng để biểu diễn giá trị hằng với tên:
  - `const double TAX_RATE = 0.0675;`
  - `const int NUM_STATES = 50;`
- Thường đặt tên viết hoa

## Program 2-28

```
1 // This program calculates the circumference of a circle.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     // Constants
8     const double PI = 3.14159;
9     const double DIAMETER = 10.0;
10
11     // Variable to hold the circumference
12     double circumference;
13
14     // Calculate the circumference.
15     circumference = PI * DIAMETER;
16
17     // Display the circumference.
18     cout << "The circumference is: " << circumference << endl;
19     return 0;
20 }
```

## Program Output

The circumference is: 31.4159

# Chuẩn và trước chuẩn C++

- Kiểu C++ phiên bản cũ:
  - Sử dụng .h tại cuối tên header
    - `#include <iostream.h>`
  - Sử dụng `#define` thay vì `const`
  - Không sử dụng `using namespace`

# HỎI ĐÁP



# Bài tập

- **Bài tập chương 1:**
  - Chapter1\_Programming Challenges.pdf



**Trân trọng cảm ơn!**

**ĐẬU HẢI PHONG**

*Giảng viên*

*dauhaiphong@dainam.edu.vn*

*0912441435*