



THỰC HỌC – THỰC NGHIỆP



Conceive Design Implement Operate

BÀI 2: CÁC KHÁI NIỆM CƠ BẢN

ĐẠI CƯƠNG LẬP TRÌNH



- Biết cách sử dụng biến
- Biết cách sử dụng hằng số
- Biết cách nhập & xuất dữ liệu
- Biết cách sử dụng các toán tử trong lập trình

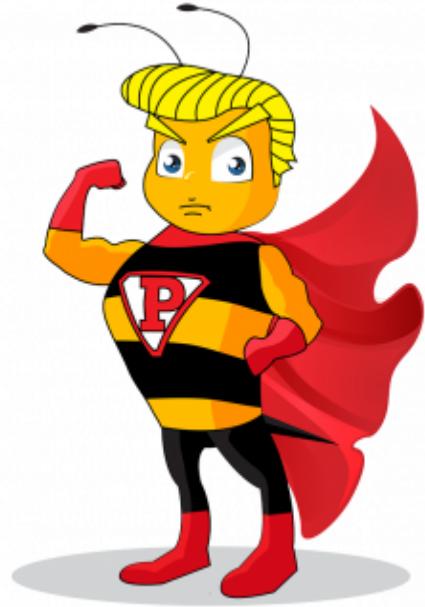
 Biến

 Hằng số

 Nhập & xuất dữ liệu

 Toán tử trong lập trình





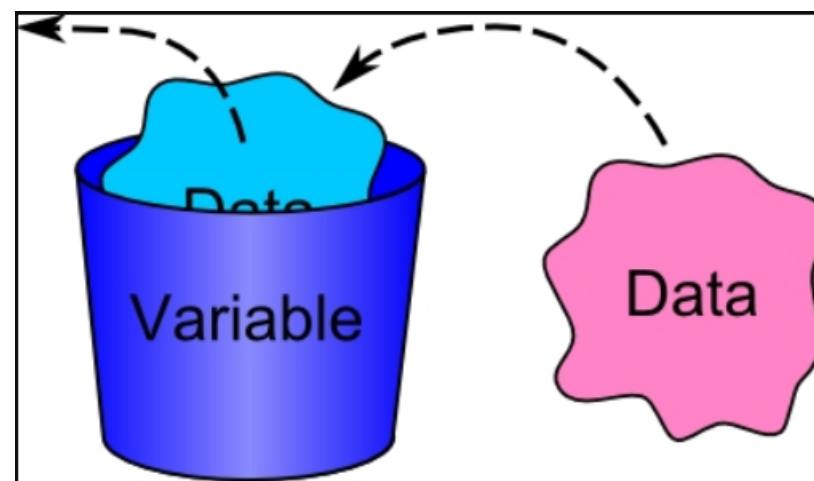
PHẦN 1: BIẾN VÀ HẰNG SỐ

- ❑ Đặt vấn đề: Khi sử dụng máy tính để làm việc (hoặc chơi game), người ta thường nói với nhau rằng: RAM (bộ nhớ tạm) càng lớn thì càng tốt.
- ❑ Vì sao???



- ❑ Trả lời: Khi khởi động một chương trình (game), thì máy tính cần đến một nơi để chứa chương trình (game) đó. Chỗ để chứa người ta gọi là bộ nhớ (RAM).
- ❑ Khi bộ nhớ (RAM) càng lớn thì khả năng chứa sẽ lớn hơn, và giúp cho máy tính hoạt động ổn định hơn khi vận hành những chương trình lớn.
- ❑ Tương tự, một chương trình cần sử dụng rất nhiều dữ liệu, và điều quan trọng là chúng ta cần có một nơi để chứa dữ liệu đó, gọi là **biến - variable**

- ❑ Định nghĩa: Biến là tên của một vùng nhớ, để chứa dữ liệu.
- ❑ Những dữ liệu có thể do người dùng nhập vào, các dữ liệu trung gian cần nơi chứa để tính toán, hay là các kết quả đầu ra sau khi đã tính toán
- ❑ Giả sử: Bạn muốn nhập vào 1 giá trị là 10 và 1 giá trị là 2 để yêu cầu máy tính xử lý phép toán 10×2 .



☐ Cách khai báo biến:

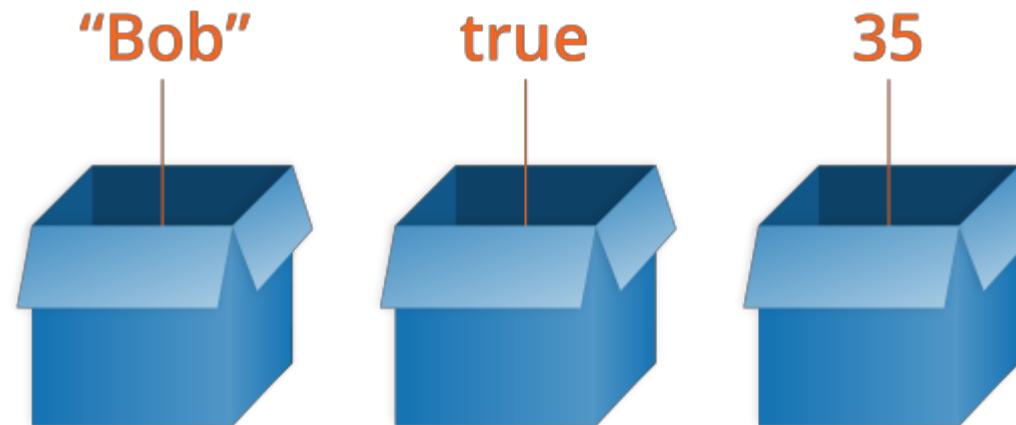
Cú pháp

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

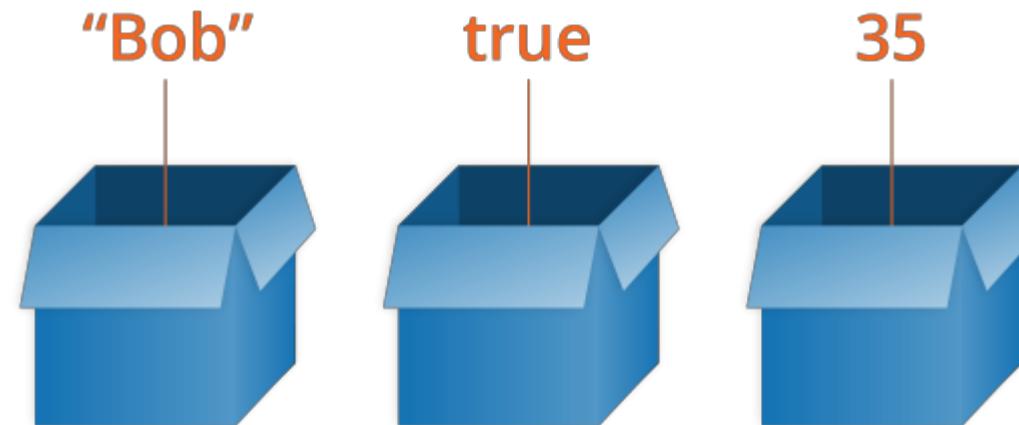
Ví dụ

int a; int b;
int a,b; //Khai báo 2 biến cùng kiểu dữ liệu

- ❑ Mỗi một biến là một kiểu dữ liệu riêng biệt
- ❑ Biến phải được khai báo rồi mới sử dụng
- ❑ Tên biến được phân biệt chữ hoa và chữ thường



- Phải bắt đầu bằng một chữ cái, hoặc dấu _
- Tên biến không được trùng với các keyword trong thư viện C
- Nên đặt tên biến theo cú pháp camelCase:
 - ❖ Ví dụ: double toanTu1; double toanTu2;



- ❑ Các từ dành riêng trong lập trình C cơ bản, những từ dành riêng có thể không được sử dụng như là hằng số hoặc các biến hoặc bất kỳ tên định danh khác.
- ❑ Một số từ khóa cơ bản trong C

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	_Packed
double			



CÁCH KHAI BÁO BIỂN

- ❑ Hằng số (constant) hướng đến những giá trị cố định mà chương trình không thể thay đổi trong quá trình thực thi.
- ❑ Hằng số có thể là một kiểu dữ liệu bất kỳ nào như kiểu dữ liệu: *số nguyên, số thực, ký tự hay chuỗi*.
- ❑ Ví dụ:
 - ❖ Số PI = 3.141592
 - ❖ Số C = 3*10⁸
 - ❖ e = 2.71828

❑ Cách định nghĩa hằng số

- ❖ #define ten_hang gia_tri
- ❖ const kieu_du_lieu ten_hang = gia_tri;

❑ Ví dụ:

- ❖ #define CHIEUDAI 15
- ❖ const int CHIEURONG = 12;



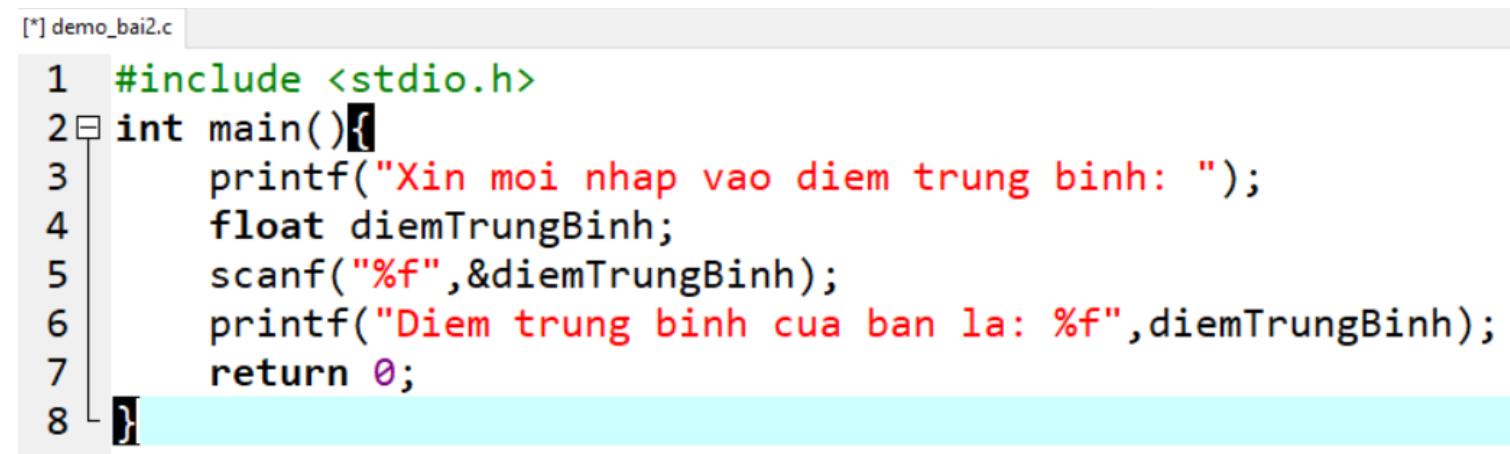
Cách khai báo hằng số

- ❑ Sử dụng hàm **printf** để xuất dữ liệu ra màn hình console (từ print có nghĩa là in).
- ❑ Khi chúng ta muốn nhập 1 giá trị từ bàn phím, và lưu giá trị đó vào trong 1 biến thì chúng ta phải sử dụng hàm **scanf**

```
demo_bai2.c
1 #include <stdio.h>
2 int main(){
3     printf("Xin moi nhap vao tuoi: ");
4     int tuoi=0;
5     scanf("%d",&tuoi);
6     printf("Tuoi cua ban la: %d",tuoi);|
7     return 0;
8 }
```

- ❑ Để nhập dữ liệu và chứa trong biến, hoặc xuất giá trị từ biến ra ngoài màn hình, chúng ta cần xác định được định dạng dữ liệu
- ❑ **Chuỗi định dạng dữ liệu bao gồm**

- ❖ %c : Kiểu char 1 ký tự
- ❖ %d : Kiểu int
- ❖ %u : Kiểu unsigned int
- ❖ %f : Kiểu float
- ❖ %lf : Kiểu double
- ❖ %s : Chuỗi ký tự



```
[*] demo_bai2.c
1 #include <stdio.h>
2 int main(){
3     printf("Xin moi nhap vao diem trung binh: ");
4     float diemTrungBinh;
5     scanf("%f",&diemTrungBinh);
6     printf("Diem trung binh cua ban la: %f",diemTrungBinh);
7     return 0;
8 }
```



NHẬP XUẤT DỮ LIỆU

- Vì sao phải sử dụng biến
- Cách sử dụng biến
- Cách sử dụng hằng số
- Nhập xuất dữ liệu





CHÈN QUIZ



PHẦN 2: TOÁN TỬ & BIỂU THỨC TOÁN HỌC

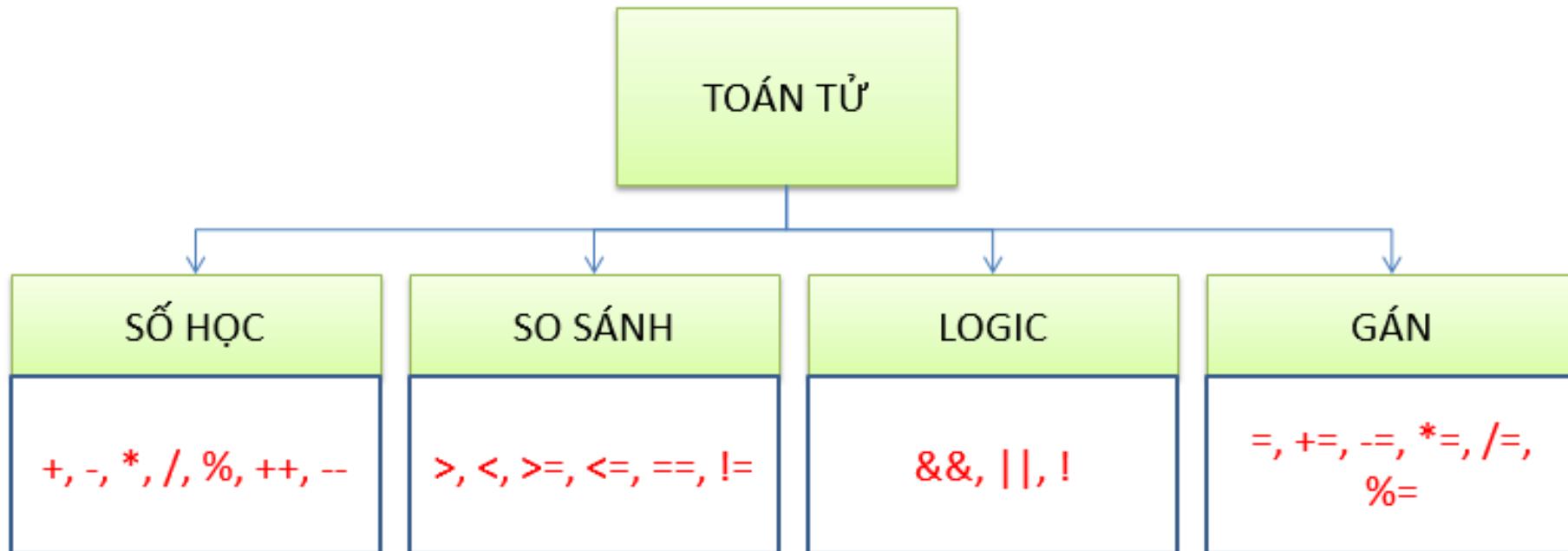
❑ Khái niệm

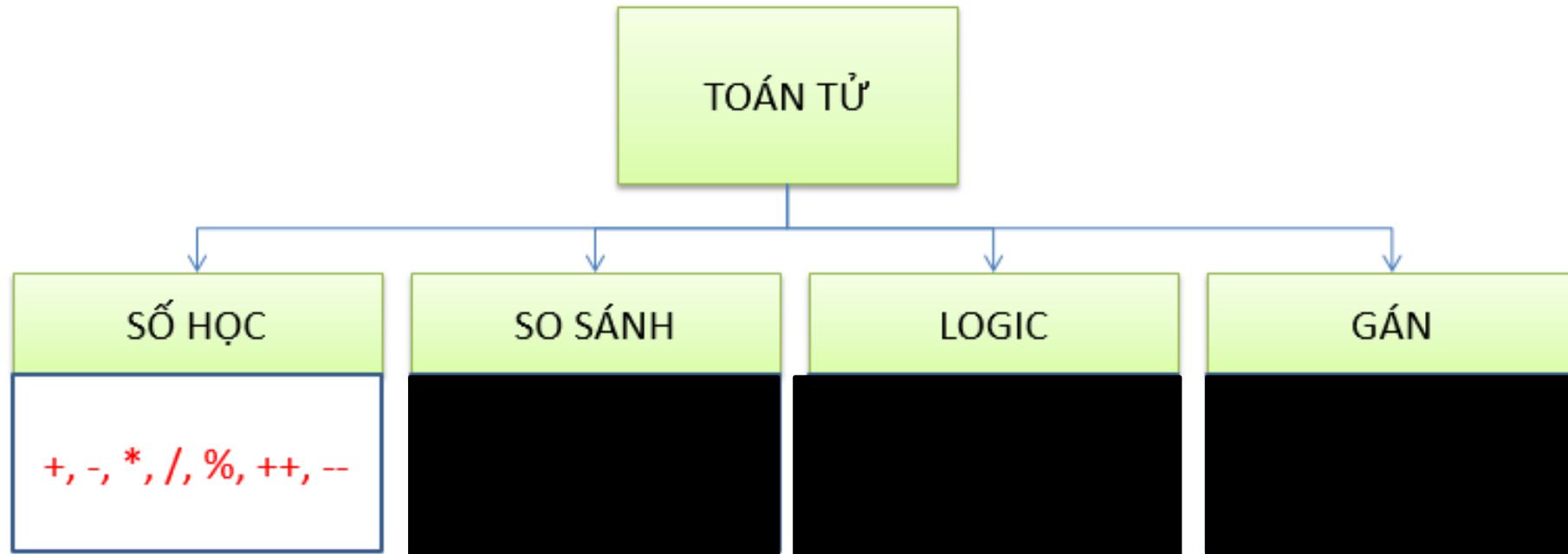
- ❖ Tạo thành từ các **toán tử** (Operator) và các **toán hạng** (Operand).
- ❖ Toán tử tác động lên các giá trị của toán hạng và cho giá trị có kiểu nhất định.
- ❖ Toán tử: **+, -, *, /, %....**
- ❖ Toán hạng: **hằng (PI), biến (toanTu1, toanTu2), lời gọi hàm...**

❑ Ví dụ

- ❖ **2 + 3, a / 5, (a + b) * 5, ...**

- Trong C, có thể phân ra 4 nhóm toán tử như sau





- ☐ Toán tử số học là các phép toán thao tác trên các số nguyên và số thực

+	Tính tổng của 2 số
-	Tính hiệu của 2 số
*	Tính tích của 2 số
/	Tích thương của 2 số
%	Thực hiện chia có dư của 2 số
++	Tăng giá trị của biến lên 1 đơn vị
--	Giảm giá trị của biến xuống 1 đơn vị

Ví dụ: tìm kết quả cho các phép tính sau

Biểu thức	Đáp án
$15 + 5 + 9 / 3 - 10 + 2 * 2$	17
$25 / 5 + 6 - 4 \% 2 + 3 * 6$	29
$14 + 16 + 30 \% 5 - 8$	22
$33 - 3 + 4 - (12 + 3) / 3$	29
$12 + 14 - 8 \% 5 + 12 * 2$	47
$13 + 14 \% 3 * 2 - 15$	2

☐ Toán tử 1 ngôi

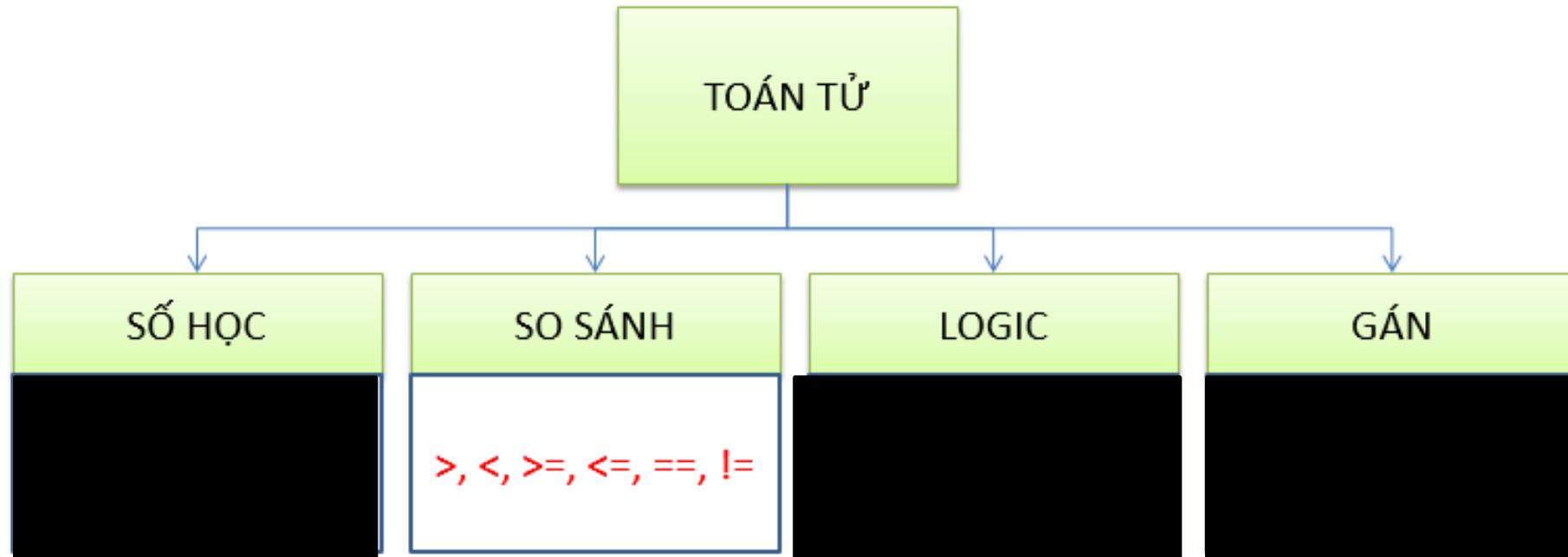
- ❖ Chỉ có một toán hạng trong biểu thức.
- ❖ ++ (tăng 1 đơn vị), -- (giảm 1 đơn vị)
- ❖ Đặt trước toán hạng
 - Ví dụ **++x** hay **--x**: thực hiện tăng/giảm **trước**.
- ❖ Đặt sau toán hạng
 - Ví dụ **x++** hay **x--**: thực hiện tăng/giảm **sau**.

☐ Ví dụ:

```
int a=5;  
a++; //a bằng 6
```



TOÁN TỬ SỐ HỌC



- ❑ Toán tử so sánh là các phép toán so sánh hai toán hạng

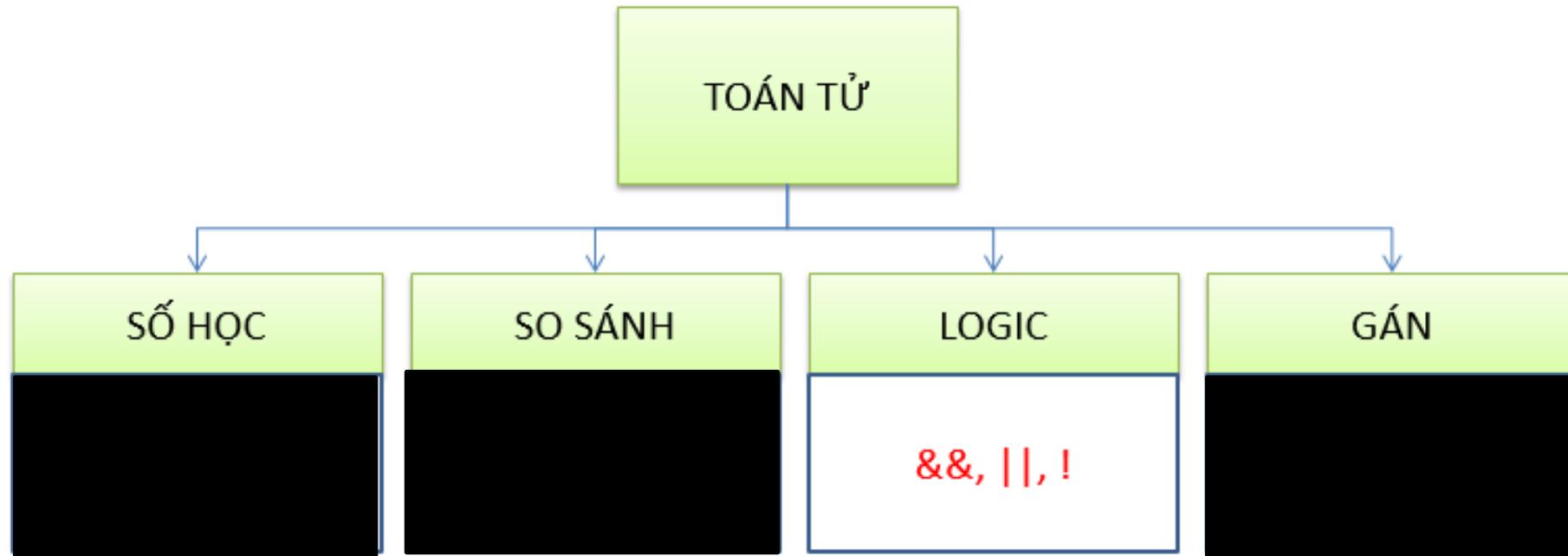
==	So sánh bằng
>	So sánh lớn hơn
>=	So sánh lớn hơn hoặc bằng
<	So sánh nhỏ hơn
<=	So sánh nhỏ hơn hoặc bằng
!=	So sánh khác

Ví dụ: tìm kết quả cho các biểu thức

Biểu thức	Kết quả
$25 + 3 * 2 != 18$	true
$5 * 3 + 2 == 11$	false
$12 - 8 <= 3 * 2$	true
$5 > 14 / 2$	false
$6 + 1 >= 7$	true
$36 / 2 - 14 != 35$	true
$45 - 5 + 12 < 19$	false



TOÁN TỬ SO SÁNH



- ❑ Là sự kết hợp của 2 hay nhiều biểu thức
- ❑ Kết quả trả về của toán tử Logic là giá trị đúng (true) hoặc sai (false)

Toán tử logic	Kết quả trả về
&&	Trả về giá trị true khi tất cả biểu thức tham gia biểu thức có giá trị true
	Trả về giá trị true khi có 1 biểu thức tham gia biểu thức có giá trị là true
!	Lấy giá trị phủ định của biểu thức

Xét ví dụ:

- ❖ Sinh viên đạt học bổng khi **điểm trung bình ≥ 8** , và **hạnh kiểm = tốt**
- ❖ Thí sinh sẽ đậu đại học khi **tổng điểm 3 môn (toán lý hóa) ≥ 21** hoặc **tổng điểm 3 môn (toán hóa sinh) ≥ 21**

Biểu thức 1	Toán tử logic	Biểu thức 2	Kết quả
Điểm trung bình ≥ 8	&&	Hạnh kiểm = tốt	True (đạt học bổng)
Điểm trung bình < 8	&&	Hạnh kiểm = tốt	False (không đạt học bổng)
Điểm trung bình ≥ 8	&&	Hạnh kiểm = khá	False (không đạt học bổng)
Điểm trung bình < 8	&&	Hạnh kiểm = khá	False (không đạt học bổng)

Xét ví dụ:

- ❖ Sinh viên đạt học bổng khi **điểm trung bình ≥ 8 , và hạnh kiểm = tốt**
- ❖ Thí sinh sẽ đậu đại học khi **tổng điểm 3 môn (toán lý hóa) ≥ 21 hoặc tổng điểm 3 môn (toán hóa sinh) ≥ 21**

Biểu thức 1	Toán tử logic	Biểu thức 2	Kết quả
Toán+ lý + hóa ≥ 21		Toán + hóa + sinh ≥ 21	True (đậu)
Toán+ lý + hóa < 21		Toán + hóa + sinh ≥ 21	True (đậu)
Toán+ lý + hóa ≥ 21		Toán + hóa + sinh < 21	True (đậu)
Toán+ lý + hóa < 21		Toán + hóa + sinh < 21	False (không đậu)

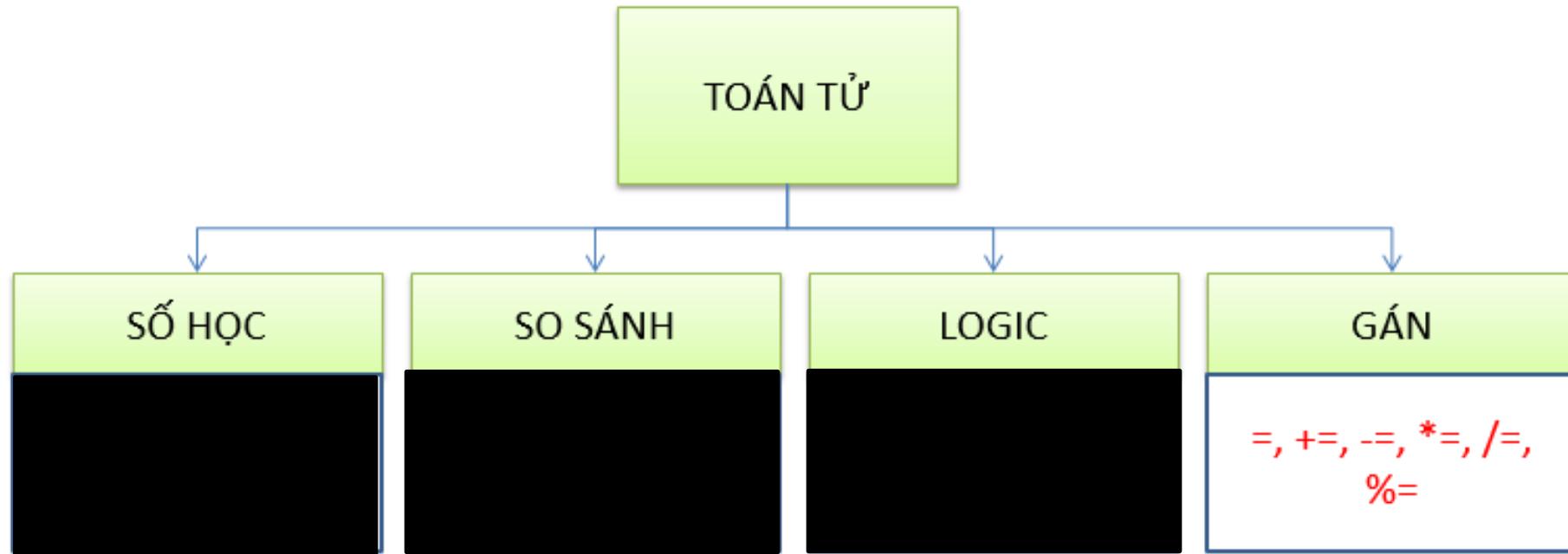
Bảng chân trị

Toán tử &&		
Biểu thức A	Biểu thức B	A && B
True	True	True
True	False	False
False	True	False
False	False	False

Toán tử		
Biểu thức A	Biểu thức B	A B
True	True	True
True	False	True
False	True	True
False	False	False



TOÁN TỬ LOGIC



❑ Khái niệm

- ❖ Thường được sử dụng trong lập trình.
- ❖ Gán giá trị cho biến.

❑ Cú pháp

- ❖ <biến> = <giá trị>;
- ❖ <biến> = <biến>;
- ❖ <biến> = <biểu thức>;

❑ Ví dụ

```
int a = 5;  
int b = a;  
int c = a+b;
```

Các phép gán

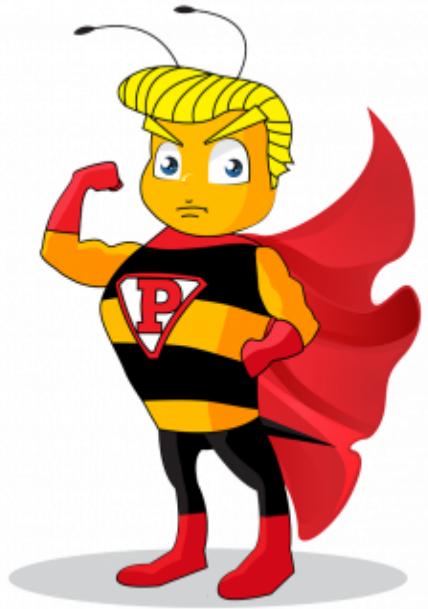
Phép gán	Ví dụ:	Kết quả:
=	$a = 20$ (gán 20 cho a)	$a == 20$
+=	$a += 10$ ($a = a + 10$)	$a == 30$
-=	$a -= 10$ ($a = a - 10$)	$a == 20$
*=	$a *= 10$ ($a = a * 10$)	$a == 200$
/=	$a /= 10$ ($a = a / 10$)	$a == 20$
%=	$a %= 10$ ($a = a \% 10$)	$a == 0$



TOÁN TỬ GÁN

- Biểu thức
- Toán tử số học
- Toán tử so sánh
- Toán tử logic
- Toán tử gán





ADD QUIZ



thank
you!